# Termodinamika interagirajućih plinova

**Trtinjak, Ivica**

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Ivica Trtinjak

# TERMODINAMIKA INTERAGIRAJUĆIH PLINOVA

Diplomski rad

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

SMJER: PROFESOR FIZIKE I INFORMATIKE

**Ivica Trtinjak**

Diplomski rad

# TERMODINAMIKA INTERAGIRAJUĆIH PLINOVA

Voditelj diplomskog rada: izv. prof. dr. sc. Robert Pezer

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2017.

## Sažetak

U ovom diplomskom radu izrađena je programska platforma za izvođenje simulacija dvodimenzionalne molekularne dinamike čije čestice međudjeluju u Lennard-Jonesovom potencijalu. Program za simuliranje molekularne dinamike na računalu je napisan u programskom jeziku Python korištenjem objektno orijentiranih principa. Korištena je programska platforma Anaconda koja uključuje integrirano okruženje za razvoj aplikacija (IDE) Spyder (Scientific PYthon Development EnviRonment) koji uključuje sve potrebne biblioteke za programsko ostvarenje naše aplikacije za simulaciju termodinamike međudjelujućih čestica. Zatim se bavimo proučavanjem uvjeta koje sustav treba zadovoljiti da bismo smatrali da se nalazi u termodinamičkoj ravnoteži. Da bismo bili sigurni u ispravnost rada simulacije molekularne dinamike, rezultate izvođenja simulacije uspoređujemo s rezultatima dobivenih LAMMPS-om (Large-scale Atomic/Molecular Massively Parallel Simulator), profesionalnom platformom za izvođenje simulacija molekularne dinamike. Zatim izvodimo računalni eksperiment na sustavu sastavljenom od deset tisuća atoma argona, čije rezultate uspoređujemo s teoretski dobivenim rezultatima pomoću virijalne jednadžbe.

# Sadržaj

# 1 Uvod

Od čega je svijet sastavljen? Kako objasniti pojave oko sebe? Takva pitanja zaokupljuju čovjeka milenijima. Prva zapisana djela s takvim pitanjima i ponuđenim potencijalnim odgovorima sežu dva i pol tisućljeća daleko u prošlost sve do Demokrita[1] i njegovog učitelja Leukipa[2]. Demokrit, zajedno s učiteljom Leukipom je začetnik ideje da su sve stvari sastavljene od nedjeljivih elemenata zvanih "atomi". Da bismo dobili zadovoljavajuće odgovore na takva pitanja trebalo je pričekati više od dva tisućljeća. Odgovori su ponuđeni u idejama vrhunskih mislioca koji su postavili temelje moderne znanosti. Jedan od njih je **Isaac Newton**[3] sa svojim monumentalnim djelima (tri knjige) *Matematički principi filozofije prirode (lat. Philosophiae Naturalis Principia Mathematica, 1687.)* U njima su zapisani temelji infinitezimalnog računa, temelji klasične mehanike - zakoni gibanja te zakon gravitacije. Dok je drugi **Albert Einstein**[4] s radom "*O zahtjevima molekularno-kinetičke teorije topline na gibanje čestica raspršenih u stacionarnoj tekućini*" kojim je otvoren put ka nepobitnoj potvrdi čestične prirode tvari. Da dodatno naglasimo koliko je ta ideja zapravo bitna, da su osnovni gradivni elementi materije čestice, poslužit ćemo se citatom još jednog velikog fizičara[5],

> "*Ako, u slučaju kakve katastrofe, svo znanstveno znanje biva uništeno, i moguće je samo jednu rečenicu prenijeti sljedećoj generaciji, koja tvrdnja bi sadržavala najviše informacija u što manje riječi? Mislim da bi to bila atomska hipoteza (ili atomska činjenica, ili kako god ju želite nazvati) koja kaže da su sve stvari sastavljene od atoma - malenih čestica koje se u neprestanom gibanju sele uokolo, privlačeći jedna drugu na malenoj udaljenosti, ali odbijajući se kada su stisnute jedna na drugu*"
>
> ***Richard Feynman***

Molekularna dinamika (MD) predstavlja metodu proučavanja svojstava sustava sastavljenog od mnoštva molekula (čestica) koje međusobno međudjeluju. Iz razloga što se takav sustav općenito sastoji od velikog broja čestica, sustav jednadžbi gibanja nije moguće analitički rješiti te to trebamo učiniti numeričkim metodama. Iz takvih razloga je očito da razvoj molekularne dinamike uvelike ovisi o razvoju računalne tehnologije jer bez računala ne bi daleko dospjeli. U ovom diplomskom radu izradit ćemo programsku platformu za izvođenje simulacija dvodimenzionalne molekularne dinamike čije čestice međudjeluju u Lennard-Jonesovom potencijalu.

[1] predsokratovski grčki filozof (rođen u Abderi-Traciji 460. - 370.pr.n.e)

[2] filozof, danas se smatra osnivačem antičkog atomizma (5 stoljeće pr.n.e).

[3] engleski fizičar, matematičar i astronom. Jedan od najznačajnijih znanstvenika u povijesti (4.1.1643.- 31.3.1727.),<https://hr.wikipedia.org/wiki/Isaac_Newton>

[4] teorijski fizičar (14.3.1879.-18.4.1955.), čiji doprinosi razvoju fizike bi ispunili cijeli prostor predviđen za uvodni dio diplomskog rada te stoga čitatelja upućujem na web stranicu <https://hr.wikipedia.org/wiki/Albert_Einstein> gdje se može informirati o tom velikanu moderne znanosti

[5] jedan od najutjecajnijih američkih fizičara 20. stoljeća (11.5.1918.-15.2.1988.), <https://en.wikipedia.org/wiki/Richard_Feynman>

Program za simuliranje molekularne dinamike na računalu je napisan u programskom jeziku Python korištenjem objektno orijentiranih principa. Korištena je programska platforma Anaconda koja uključuje integrirano okruženje za razvoj aplikacija (IDE) Spyder (Scientific PYthon Development EnviRonment) koji uključuje sve potrebne biblioteke za programsko ostvarenje naše aplikacije za simulaciju termodinamike međudjelujućih čestica. Za numeričko integriranje koristit ćemo brzinski Verlet algoritam. Da bismo se riješili kvadratne ovisnosti vremena izvođenja simulacija MD, implementirat ćemo Verletovu listu susjeda u kombinaciji s listom ćelija susjeda s čime će ovisnost vremena izvođenja o broju čestica biti linearizirana. Termodinamičke veličine su definirane kao vremenski prosjeci veličina sustava koji se nalazi u termodinamičkoj ravnoteži. Iz tog razloga ćemo značajnu pažnju posvetiti proučavanju uvjeta koje sustav treba zadovoljiti da bi ga smatrali termodinamički uravnoteženim. Međutim, iako bi simulacija mogla raditi, postavlja se pitanje kako znamo radi li ispravno. U tu svrhu ćemo rezultate rada simulacije MD uspoređivati s rezultatima dobivenih LAMMPS-om (Large-scale Atomic/Molecular Massively Parallel Simulator), profesionalnom platformom za izvođenje simulacija molekularne dinamike. U drugom dijelu diplomskog rada izvest ćemo računalni eksperiment na sustavu sastavljenom od deset tisuća atoma argona čije ćemo rezultate usporediti s teoretski dobivenim rezultatima pomoću virijalne jednadžbe koju proučavamo u drugom dijelu ovog rada.

# 2 Uvod u molekularnu dinamiku

Molekularna dinamika je metoda računalne simulacije koja nam omogućuje predviđanje vremenske evolucije sustava međudjelujućih čestica. U grubo, razvoj procesa simulacije molekularne dinamike na računalu prikazan je na dijagramu 2.1. Za sustav koji želimo proučavati prvo zadamo početne uvjete (početni položaji i početne brzine čestica) i međučestični potencijal. Zatim, evoluciju sustava možemo proučavati kada riješimo sustav jednadžbi gibanja za sve čestice u sustavu. Rješenja tih jednadžbi su položaji ($\vec{r}_i(t)$) i brzine ($\vec{v}_i(t)$) svih čestica kao funkcije vremena. To učinimo pomoću drugog Newtonovog zakona gibanja:

$$m_i \frac{d^2\vec{r}_i}{dt^2} = \vec{F}_i \tag{2.1}$$

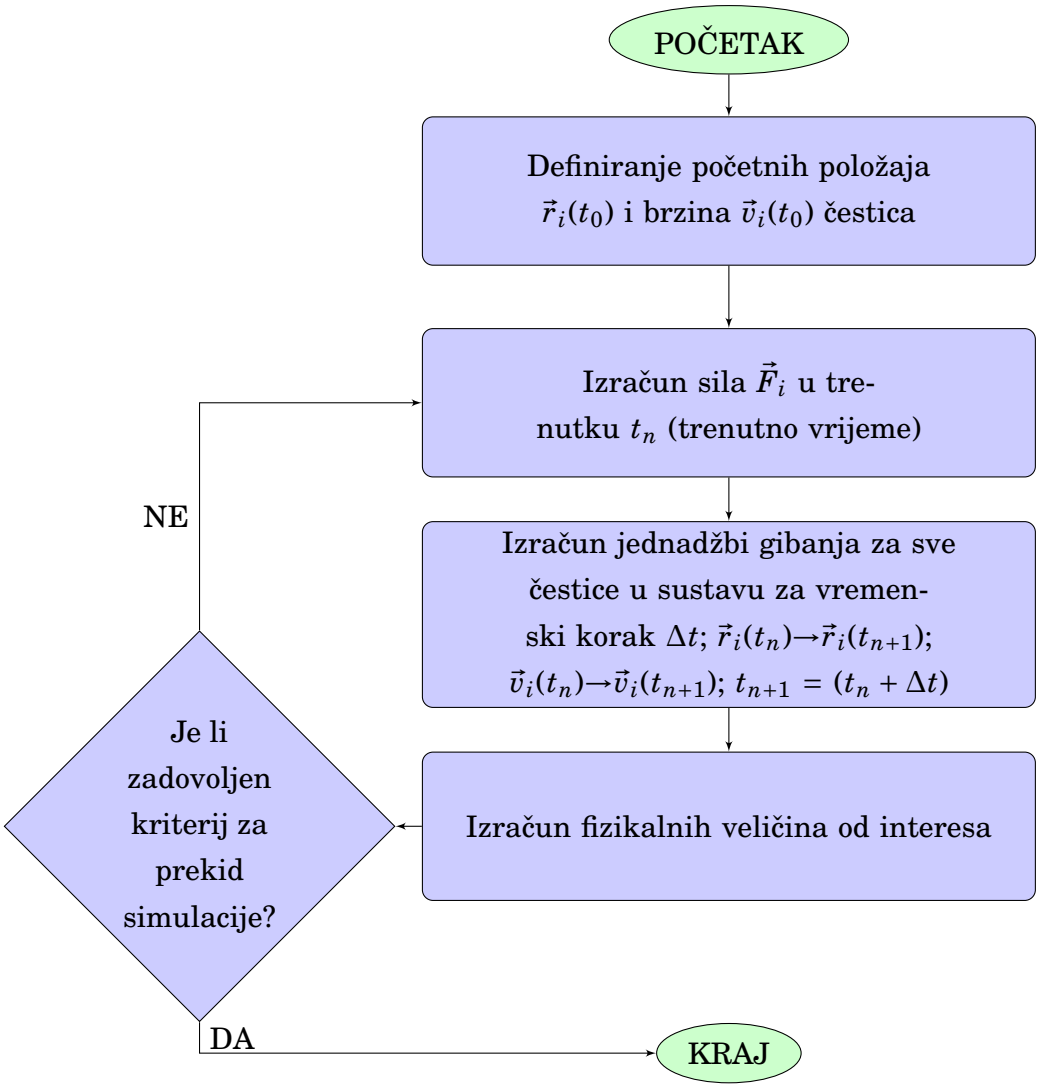
Silu $\vec{F}_i$ općenito izračunamo preko funkcije međučestičnog potencijala kao:

$$\vec{F}_i = -\nabla_i U(\vec{r}_1, \vec{r}_2, \vec{r}_3, ..., \vec{r}_N) \tag{2.2}$$

Odabir međučestičnog potencijala prije svega ovisi o vrsti problema koje želimo proučavati. Funkcionalna forma međučestičnog potencijala najčešće je dobivena eksperimentalno (empirijski) jer alternativni načini (od prvih principa) su najčešće prekompleksni za praktičnu upotrebu. U našem slučaju koristimo Lennard-Jonesov potencijal 2.5 opisan u potpoglavlju 2.2. Ukupnu energiju sustava sastavljenog od $N$ čestica koje međudjeluju nekim empirijskim potencijalom može se razviti u više-čestični razvoj kao:

$$U(\vec{r}_1, \vec{r}_2, \vec{r}_3, ..., \vec{r}_N) = \sum_i U_1(r_1) + \sum_i \sum_{j>i} U_2(\vec{r}_i, \vec{r}_j) + \sum_i \sum_{j>i} \sum_{k>j} U_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) + ... \tag{2.3}$$

Jednočestični član ($U_1$) predstavlja granične uvjete (npr. zidovi spremnika) ili vanjsko polje. Dvočestični član ($U_2$), ili "pair potential" kako mu je čest naziv u literaturi, uzima u obzir međudjelovanje između bilo kojeg para čestica koje ovisi samo o njihovoj međusobnoj udaljenosti. Lennard-Jonesov potencijal je primjer takvog "pair potential"-a. Tročestični član ($U_3$) uzima u obzir međudjelovanje para čestica modificiran zbog prisutnosti treće čestice, itd. Nakon što smo zadali početne uvjete i međučestični potencijal, jednadžbe gibanja riješimo numerički jer ne postoji analitičko rješenje za $N > 3$ čestica. Zatim računamo, ovisno o problemu, veličine koje nas interesiraju (npr. tlak). Kada izračunamo sve veličine od interesa, krećemo s provjerom kriterija za prekid simulacije. Ti kriteriji mogu biti razne vrste. Od onih jednostavnih, poput, ako je proteklo vrijeme veće od nekog zadanog $t_{n+1} > t_{max}$ pa sve do kompliciranih, poput, da li je sustav u termodinamičkoj ravnoteži. U ovom diplomskom radu će nam potonji navedeni kriterij biti od većeg interesa. Ako je kriterij zadovoljen, simulacija je gotova, a u suprotnom ponavljamo proces kao što je prikazano na dijagramu 2.1. Detaljnije o svemu do sada navedenom i ostalim temama slijedi u nastavku diplomskog rada.




Slika 2.1: Dijagram procesa simulacije molekularne dinamike

## *2.1 Povijesni počeci molekularne dinamike*

Prve simulacije molekularne dinamike (MD) koreliraju s razvojem računalne tehnologije. Među prvim istraživanjima je rad "*Equation of State Calculations by Fast Computing Machines*" (1953 g.) autora: Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth i Augusta H. Teller [9]. U njemu opisuju općenitu metodu za proučavanje jednadžbe stanja za materiju koja se sastoji od međudjelujućih čestica. Iako metoda (Monte Carlo) korištena u tom radu nije metoda kojom se pristupa rješavanju problema u molekularnoj dinamici (numeričko rješavanje Newtnovih jednadži gibanja) s takvim istraživanjima i porastom računalne snage otvoren je put istraživanjima metodama MD. Prva simulacija koja je koristila metode MD je objavljena u radu "Phase Transition for a Hard Sphere System" (1957 g.) autora B. J. Alder i T. E. Wainwright [10]. Zatim radovi "Correlations in the Motions of Atoms in Liquid Argon" (1964 g.) [11] te "Molecular Dynamics Study of Liquid Water" (1971 g.) autora A. Rahman [12]. Početnim uspjesima i daljnjim ubrzanim razvojem računalne tehnologije omogućen je daljnji napredak na području molekularne dinamike.

## 2.2 Međučestični potencijal

Dinamiku čestica opisujemo uzimajući u obzir samo klasičnu mehaniku. Čestice su sfernog oblika, kemijski inertne i njihovu unutarnju strukturu zanemarujemo. Međudjelovanje između bilo kojeg para čestica ovisi samo o udaljenosti između njih. U tom slučaju ukupna potencijalna energija $U$ je suma dvočestičnih međudjelovanja:

$$U = u(r_{12}) + u(r_{13}) + ... + u(r_{23}) + ... = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} u(r_{ij}) \tag{2.4}$$

gdje $u(r_{ij})$ ovisi samo o međusobnoj udaljenosti između čestica $i$ i $j$. Forma potencijala $u(r_{ij})$ za električki neutralne čestice može se izvesti iz prvih principa kvantne mehanike, međutim to je vrlo zahtjevno i za većinu slučajeva nepotrebno [2]. Pogodnije je odabrati jednostavniju fenomenološku[6] formu potencijala $u(r)$. Najvažnija svojstva takvog potencijala su:

- snažno odbijanje na maloj udaljenosti $r$ između čestica
- slabo privlačenje na velikoj udaljenosti $r$ između čestica

Odbijanje na maloj udaljenosti je posljedica Paulijevog principa isključenja. Slabo privlačenje na velikoj udaljenosti je posljedica obostrane polarizacije čestica. U ovom diplomskom radu proučavamo međudjelovanje čestica u Lennard-Jonesovom potencijalu $u(r)$:

$$u(r) = 4\epsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{2.5}$$

Graf Lennard-Jonesovog potencijala je prikazan na slici 2.2. Odbojni dio međudjelovanja $r^{-12}$ nema fundamentalnog značenja i takav je zbog matematičke pogodnosti. Privlačni dio $r^{-6}$ odgovara Van der Waalsovom međudjelovanju među česticama. Parametar $\sigma$ i $\epsilon$ karakteriziraju dotičnu tvar (npr. plin argon). Parametar $\epsilon$ ima dimenziju energije i predstavlja minimum potencijala pri ravnotežnoj udaljenosti između dvije čestice $r = 2^{1/6}\sigma$. Parametar $\sigma$ ima dimenziju duljine i predstavlja udaljenost na kojoj je međučestični potencijal jednak nuli.

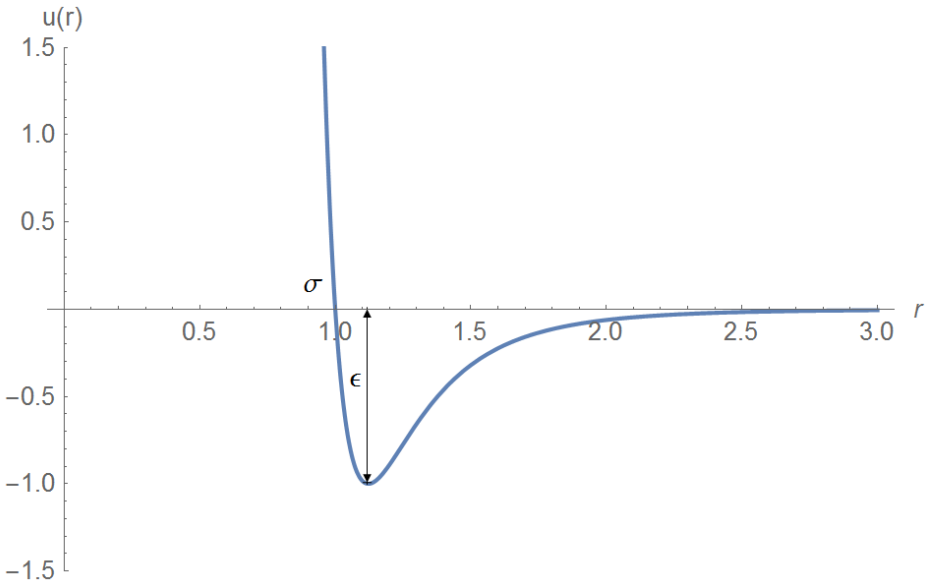
## 2.3 Sila na česticu u Lennard-Jonesovom potencijalu

Sila koja djeluje na česticu u Lennard-Jonesovom potencijalu 2.5 jest:

$$\vec{F} = -\nabla u(r) \tag{2.6}$$

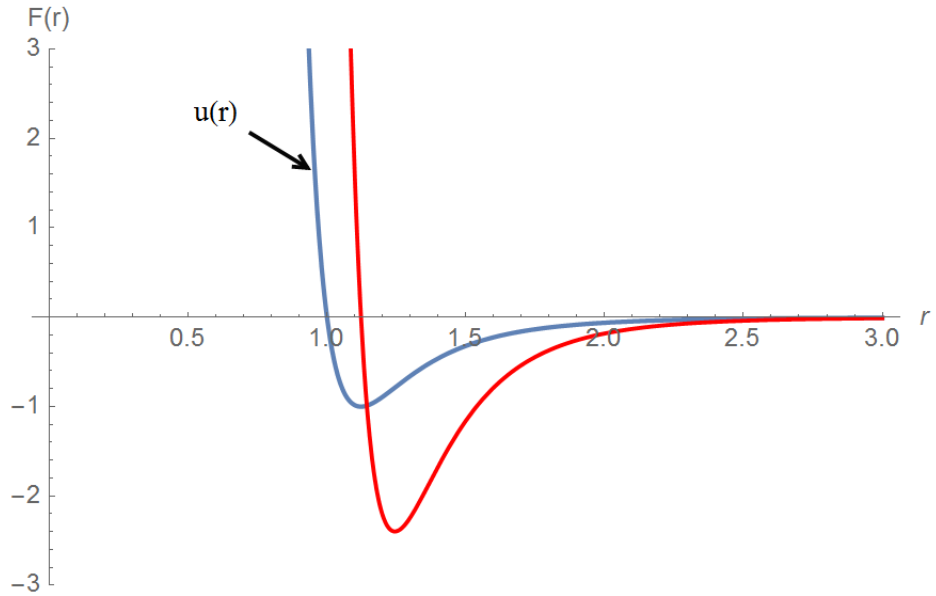
$$\vec{F_{ij}} = \frac{24\epsilon}{|\vec{r_{ij}}|} \left[ 2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] \vec{r_{ij}} \tag{2.7}$$

[6] Pod time se misli da opis pojave nije izveden iz prvih principa




Slika 2.2: Lennard-Jones potencijal ($\sigma = 1, \epsilon = 1$)

gdje je $\vec{F}_{ij}$ sila kojom čestica $i$ djeluje na česticu $j$. Graf sile je prikazan na slici 2.3. Vidimo da na udaljenosti manjoj od $r = 2^{1/6}\sigma$ sila između čestica postane odbojna (posljedica Paulijevog principa isključenja), dok je na udaljenosti većoj od $r = 2^{1/6}\sigma$ sila između čestica privlačna (posljedica uzajamne polarizacije molekula).




Slika 2.3: Sila na česticu $i$ od čestice $j$ u Lennard-Jones potencijalu ($\sigma = 1, \epsilon = 1$)

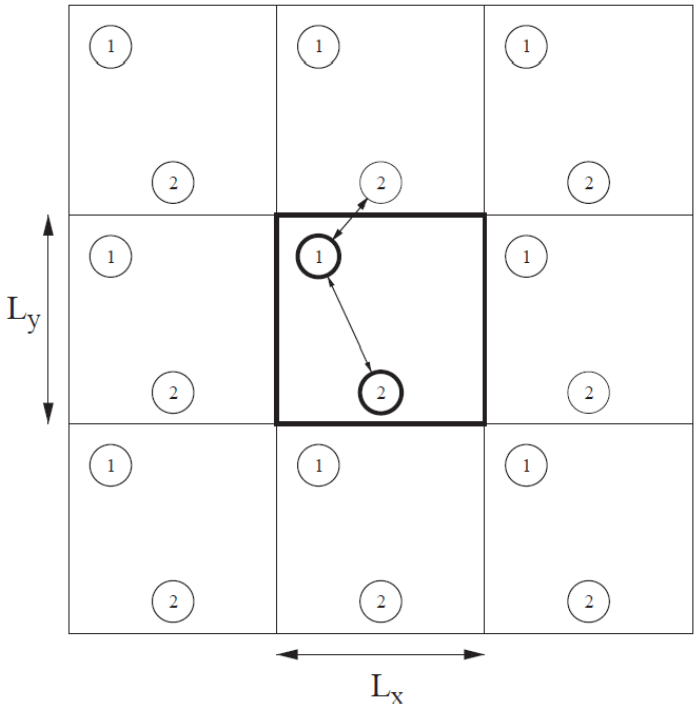
Jednadžba gibanja čestice $i$ prema drugom Newtonovom zakonu jest:

$$m\ddot{\vec{r}}_i = \vec{F}_i = \sum_{j=1, j\neq i}^{N} \vec{F}_{ij} \tag{2.8}$$

gdje suma ide preko svih parova čestica (bez čestice $i$).

## 2.4 *Rubni uvjeti*

Materija koju želimo simulirati je sustav sastavljen od $N \sim 10^{23}$ čestica koje se nalaze u spremniku. U takvim sustavima samo se maleni dio čestica nalazi na zidovima spremnika. Broj čestica u tipičnim simulacijama molekularne dinamike su reda veličine $10^3 - 10^6$ čestica (na superračunalima nekoliko redova veličine više). U takvim malenim sustavima broj čestica koji se nalaze na granici sustava i okoline je značajan i ako ne proučavamo ponašanje sustava kojim dominiraju granična svojstva, želimo minimizirati te efekte. To se ostvaruje pomoću periodičkih rubnih uvjeta (PRU). PRU




Slika 2.4: Primjer periodičkih rubnih uvjeta u dvije dimenzije

implicira da centralna ćelija mora biti replicirana beskonačno mnogo puta da ispuni površinu. Na primjer, na slici 2.4[7] prikazujemo centralnu ćeliju s dvije čestice koja je jednom kopirana u obje dimenzije sustava. PRU u dvije dimenzije je sustav kod kojeg si zamišljamo $N$ čestica (na našem primjeru N=2) koje se nalaze u ćeliji (središnja ćelija 2.4) kod kojeg su nasuprotne strane tako spojene da formiraju površinu torusa. U tom slučaju čestica koja bi izašla na jednom kraju te ćelije bi se vratila na drugom kraju. Relativne udaljenosti između čestica u kopijama centralne ćelije trebaju biti jednake. To ujedno implicira da je svaka točka u ćeliji ekvivalentna te da nema kraja [2]. Iz razloga što gibanje čestice u jednoj ćeliji je ekvivalentno gibanje čestica u bilo kojoj drugoj ćeliji, promatranje gibanja čestica u ćelijama se svodi na promatranje gibanja čestica u samo jednoj ćeliji. Kada jedna čestica $i$ napusti centralnu ćeliju, to se događa u svim kopijama te tako dobijemo česticu koja uđe u centralnu ćeliju s nasuprotne strane i predstavlja česticu $i$. Ukupna sila na česticu $i$ je jednaka sumi sila

[7] Slika preuzeta iz [2]

svih preostalih čestica $j$ na tu česticu $i$ uključujući i doprinose od svih $j$ čestica u kopijama centralne ćelije. To znači da imamo beskonačno članova koji doprinose sili na jednu česticu. Za međučestična međudjelovanja kratkog dometa uzimamo da čestica $i$ međudjeluje samo s najbližim česticama $j$ u susjednim kopijama (vidi 2.4), a za sve ostale čestice $j$ koje su udaljene više $L/2$ ($L$ je dužina stranice ćelije) doprinos sili je jednak nuli [2].

## *2.5 Kratki uvod u statističku fiziku*

### 2.5.1 Potreba za statistikom

Iako možemo izračunati putanju bilo koje čestice, to ne radimo iz više razloga. Sasvim praktičan razlog je da bi takav pristup generirao vrlo veliku količinu informacija koje bi ubrzo premašile kapacitet bilo kojeg sustava za pohranu podataka od kojih ne bi bilo mnogo koristi. Fizikalniji razlog je taj što je sustav, koji se sastoji od mnogo čestica, vrlo osjetljiv na početne uvjete. To znači da će se putanje čestica od dva identična sustava koji počinju sa malo drugačijim početnim uvjetima drastično razlikovati. Iako su Newtonove jednadžbe vremenski reverzibilne, ta invarijantnost ne može biti realizirana u praksi (zbog nemogućnosti beskonačno preciznog mjerenja) te nas prisiljava da problemu pristupimo statistički [2].

### 2.5.2 Ansambl

Različita makroskopska ograničenja (ukupna energija $E$, broj čestica $N$, volumen $V$, kemijski potencijal $\mu$) koja nametnemo sustavu vode različitim tipovima ansambla sa specifičnim statističkim karakteristikama. Tri važna ansambla definirana su od strane J.W. Gibbsa[8]; mikrokanonski $(E,N,V)$, kanonski $(N,T,V)$ i velekanonski ansambl $(\mu,T,V)$. Zamislimo sustav nad kojim vršimo eksperiment i opservacije. Umjesto ponavljanja pokusa identičnim sustavom, možemo zamisliti da imamo veliki broj jednakih sustava. Jednakih u smislu nametnutih ograničenja sustavu (npr. konstantan broj čestica $N$). Takav zamišljeni skup jednakih sustava naziva se ansambl.

### 2.5.3 Mikrokanonski ansambl (E,N,V)

Član mikrokanonskog ansambla je sustav koji je potpuno izoliran od okoline ima konstantnu ukupnu energiju $E$ i konstantan broj čestica $N$. Na njega stavljamo dodatni uvjet za konstantnim volumenom $V$. Pod takvim uvjetima pretpostavljamo da su sva mikroskopska stanja[9] jednako vjerojatna.

[8] Američki fizičar iz 19 st. koji je zajedno s J.C. Maxwellom i L. Boltzmannom stvorio statističku mehaniku

[9] Pod pojmom mikroskopskog stanja podrazumijevamo da je na razini čestica točno utvrđena jedna od mogućih raspodjela ukupne energije $E$ na razna pobuđenja

### 2.5.4 Ergotska hipoteza

Iako eventualno izračunate putanje ne bi bile one koje bismo mislili da računamo, izračunati položaji i brzine čestica su konzistentni s ograničenjima nametnutim sustavu. U našem slučaju, ukupna energija $E$, broj čestica $N$ i volumen $V$. Zbog pretpostavke o jednakim vjerojatnostima za sva mikroskopska stanja izoliranog sustava, sve putanje s tim ograničenjima će jednako doprinositi srednjim vrijednostima makroskopskih veličina. Putanje koje ćemo izračunavati u simulaciji su korisne iako se razlikuju od točnih putanja koje bismo dobili s računalom beskonačne preciznosti. Numeričkim rješavanjem Newtonovih jednadžbi, kao što ćemo mi učiniti, dopušta nam da računamo vremenske prosjeke putanja u faznom prostoru kroz konačne vremenske intervale. Iz razloga što je vrijeme nebitno za sustave u termodinamičkoj ravnoteži, vremensko usrednjavanje i usrednjavanje pomoću ansambla su ekvivalentni. Ta ekvivalencija se zove *Ergotska hipoteza*. Pretpostavka je da ako čestičnu (molekularnu) simulaciju izvodimo dovoljno dugo, da će sustav čestica "istražiti" sav raspoloživ fazni prostor te će mjerena veličina predstavljati prosjek. Jedan od načina kako to eksperimentalno provjeriti je da izračunamo veličine od interesa pomoću mnogo nezavisnih ansamblova od kojih svaki ansambl ima različite početne konfiguracije. Direktnija mjera "ergotizma" se temelji na usporedbi vremenski usrednjene veličine $\overline{f_i(t)}$ od $f_i$ za česticu $i$ naspram srednje vrijednosti te veličine od svih ostalih čestica [2]. Ako je sustav ergotski, tada će sve čestice "vidjeti" isto prosječno okruženje i vremeski prosjek $\overline{f_i(t)}$ za svaku od čestica će biti jednak nakon dovoljno mnogo vremena (npr. simuliramo vrlo veliki sustav te usporedimo ponašanje različitih dijelova sustava). Vremenski prosjek veličine $f_i$ je definiran kao:

$$\overline{f_i(t)} = \frac{1}{t}\int_0^t f(t')dt' \tag{2.9}$$

a prosjek $\overline{f_i(t)}$ za sve čestice kao:

$$\langle f(t)\rangle = \frac{1}{N}\sum_{i=1}^{N}\overline{f_i(t)} \tag{2.10}$$

### 2.5.5 Idealni plin

Najjednostavniji model makroskopskog sustava je onaj kod kojeg je međudjelovanje među česticama sustava zanemarivo. U tom slučaju potencijalna energija međudjelovanja između čestica je zanemariva naspram kinetičke energije. Da bi taj uvjet bio zadovoljen, gustoća čestica sustava mora biti dovoljno malena te će zbog toga sudari između čestica biti rijetki i u većini slučajeva zanemarivi. U slučaju kada potpuno zanemarimo međudjelovanje između čestica sustava, taj sustav možemo modelirati kao *idealni plin* koji je opisan jednadžbom:

$$pV = nRT \tag{2.11}$$

Tu jednadžbu (2.11), koja daje vezu između makroskopskih varijabli stanja sustava, tlaka $p$, volumena $V$, temperature $T$ i količine tvari $n$ ($R$ je plinska konstanta) zovemo

jednadžbom stanja idealnog plina i primjenjiva je na bilo koji sustav koji zadovoljava gore navedene uvjete.

### 2.5.6 Realni plin

Kod modeliranja realnih plinova uzimamo u obzir međudjelovanje među česticama. Još u 19 st. fizičar Van der Waals proučavao je moguću jednadžbu za opisivanje realnog stanja plina koja uključuje odstupanja od idealnog plina. Za jedan mol tvari ($n = 1$), Van der Waalsova jednadžba glasi:

$$(p + \frac{a}{v^2})(v - b) = RT \tag{2.12}$$

gdje je s $v$ označen volumen koji se odnosi na jedan $mol$ tvari, a pozitivne konstante $a$ i $b$ se utvrđuju empirijski za svaku pojedinu tvar. Parametar $a$ je povezan s privlačnim dijelom međudjelovanja među česticama, dok parametar $b$ uzima u obzir da čestice ne tretiramo kao točke već imaju volumen i to na način da im umanji raspoloživ volumen. Logika postavljene jednadžbe je u tome da za dovoljno veliki volumen $v$, član $\frac{a}{v^2}$ možemo zanemariti naspram tlaka plina $p$, te isto tako $b$ naspram $v$. U tom slučaju dobijemo jednadžbu stanja idealnog plina $pV = nRT$, što se i očekuje kada jednom molu tvari stavimo na raspolaganje dovoljno veliki volumen. Iako jednadžba (2.12) predstavlja samo jednu od mnogih fenomenoloških jednadžbi za opisivanje stanja materije, njezina prednost prilikom opisa rijetkih plinova naspram drugih fenomenoloških jednadžbi je prije svega njezina jednostavnost i mogućnosti izvođenja iz virijalne jednadžbe stanja (3.40) čiji se rezultati mogu potkrijepiti sa statističkom fizikom.

## *2.6 Raspodjela brzina čestica*

Brzine čestica klasičnog (nekvantnog) plina u termodinamičkoj ravnoteži opisana je Maxwellovom raspodjelom brzina. Za trodimenzionalan slučaj Maxwellova raspodjela brzina dana je izrazom 2.13.

$$f(v)dv = 4\pi v^2 \Big(\frac{m}{2\pi k_B T}\Big)^{\frac{3}{2}} e^{-mv^2/2k_B T} dv \tag{2.13}$$

Iz jednadžbe 2.13 proizlazli da vjerojatnost da čestica ima brzinu u intervalu $[v_1, v_2]$ jest:

$$\int_{v_1}^{v_2} f(v)dv \tag{2.14}$$

Pomoću funkcije gustoće vjerojatnosti $f(v)$ možemo izračunati razne korisne veličine, a neke od njih koje su nam od interesa su srednja brzina čestica koju izračunamo pomoću izraza definiranog u jednadžbi 2.15,

$$\langle v \rangle = \int_0^{\infty} v f(v)dv \tag{2.15}$$

najvjerojatnija brzina čestice $v_{nv}$ dana izrazom 2.16,

$$\frac{df(v)}{dv} = 0 \tag{2.16}$$

te korijen iz srednje kvadratne brzine (RMS - Root Mean Square) definiran izrazom 2.17.

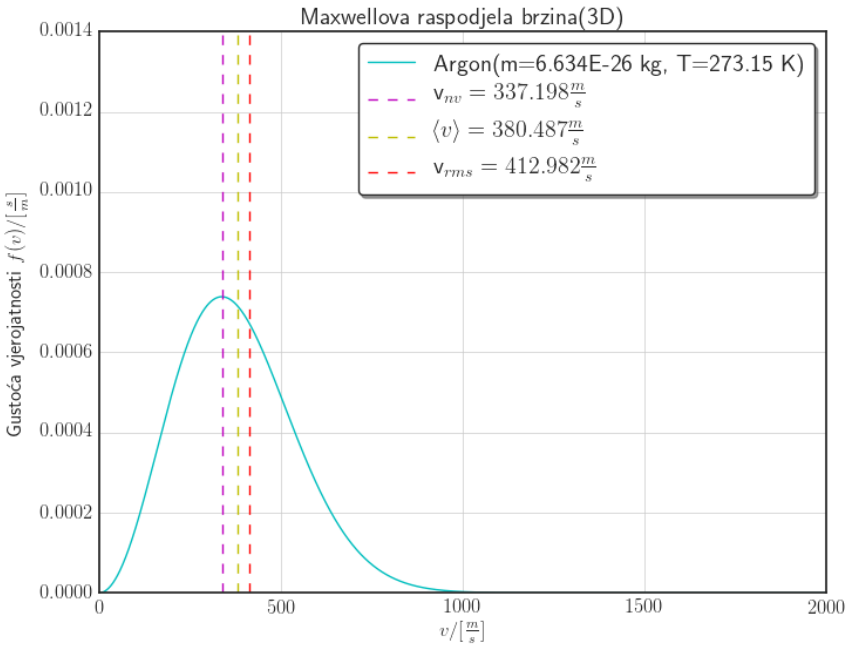
$$v_{rms} = \sqrt{\langle v^2 \rangle} = \left( \int_0^\infty v^2 f(v) dv \right)^{\frac{1}{2}} \tag{2.17}$$

Za 3D Maxwellovu raspodjelu brzina za $\langle v \rangle$, $v_{nv}$, $v_{rms}$ nakon izračuna dobijemo redom izraze 2.18, 2.19, 2.20. Na slici 2.5 možemo vidjeti odnos tih veličina na primjeru sustava od atoma argona na temperaturi od $T = 273.15K$.

$$\langle v \rangle = \sqrt{\frac{8}{\pi}} \sqrt{\frac{k_B T}{m}} \tag{2.18}$$

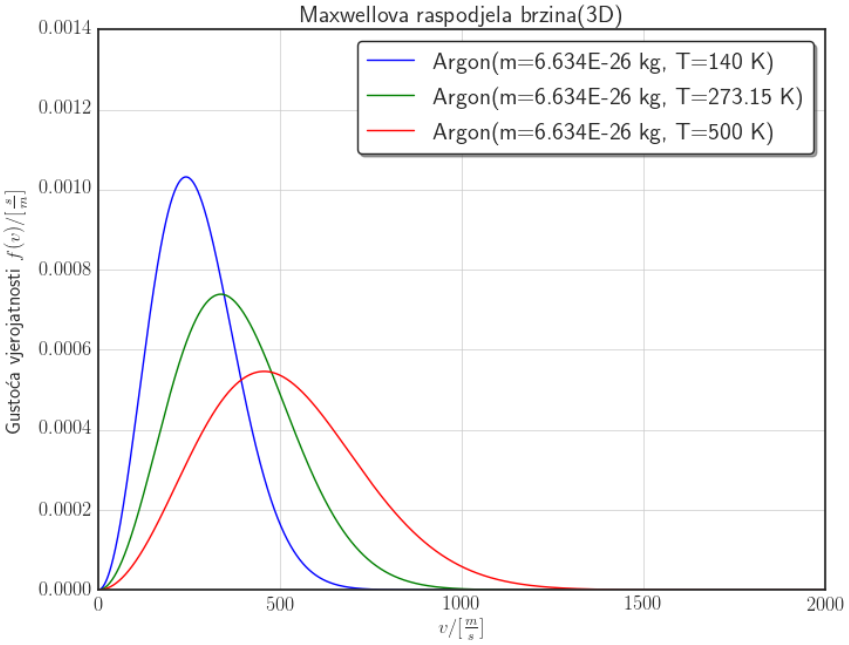
$$v_{nv} = \sqrt{2} \sqrt{\frac{k_B T}{m}} \tag{2.19}$$

$$v_{rms} = \sqrt{3} \sqrt{\frac{k_B T}{m}} \tag{2.20}$$

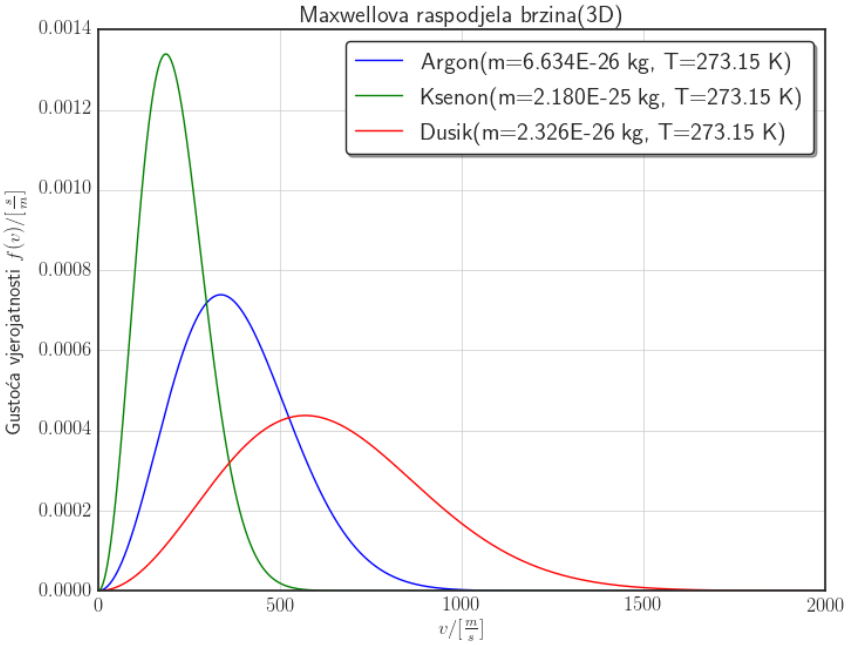



Slika 2.5: Raspodjela gustoće vjerojatnosti $f(v)$, $\langle v \rangle$, $v_{nv}$, $v_{rms}$

Vidimo da raspodjela gustoće vjerojatnosti $f(v)$ ovisi još i o masi čestica $m$ i temperaturi sustava čestica $T$. Na slici 2.6 su prikazane raspodjele gustoće vjerojatnosti u ovisnosti o brzini čestice za nekoliko različitih temperatura sustava čestica sastavljenog od atoma argona.




Slika 2.6: Raspodjela gustoće vjerojatnosti $f(v,T)$




Slika 2.7: Raspodjela gustoće vjerojatnosti $f(v,m)$

Na slici 2.7 su prikazane raspodjele gustoće vjerojatnosti o brzini čestice za čestice različitih masa (vrsta čestica) koje će nam biti zanimljive.

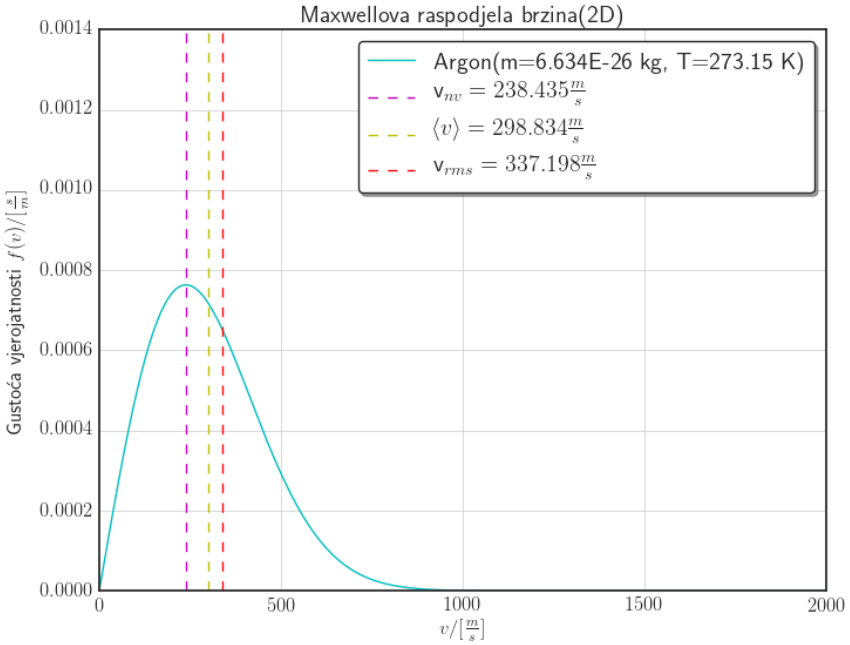
Za 2D slučaj (slike 2.8,2.9,2.10) Maxwellova raspodjela brzina je dana izrazom 2.21, a pripadne veličine od interesa $\langle v \rangle$, $v_{nv}$, $v_{rms}$, redom u izrazima 2.22, 2.23, 2.24.

$$f(v)dv = 2\pi v\left(\frac{m}{2\pi k_B T}\right)e^{-mv^2/2k_B T}dv \tag{2.21}$$

$$\langle v \rangle = \sqrt{\frac{\pi}{2}}\sqrt{\frac{k_B T}{m}} \tag{2.22}$$

$$v_{nv} = \sqrt{2}\sqrt{\frac{k_B T}{m}} \tag{2.23}$$

$$v_{rms} = \sqrt{\frac{k_B T}{m}} \tag{2.24}$$



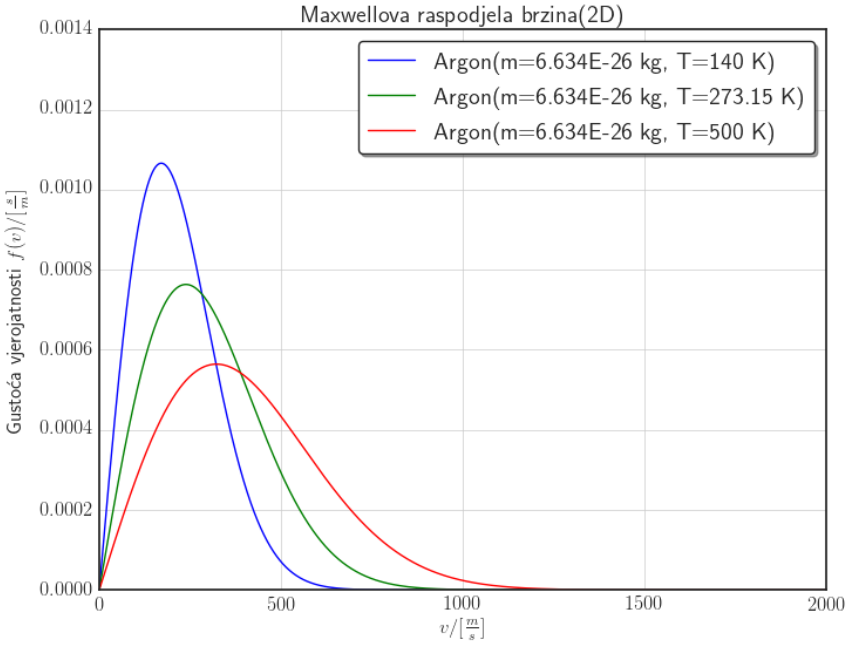

Slika 2.8: Raspodjela gustoće vjerojatnosti $f(v)$, $\langle v \rangle$, $v_{nv}$, $v_{rms}$

Iz razloga što nam je dostupan podatak o broju čestica u sustavu, zornije nam je promatrati veličinu $N_v$ definiranu izrazom 2.25.

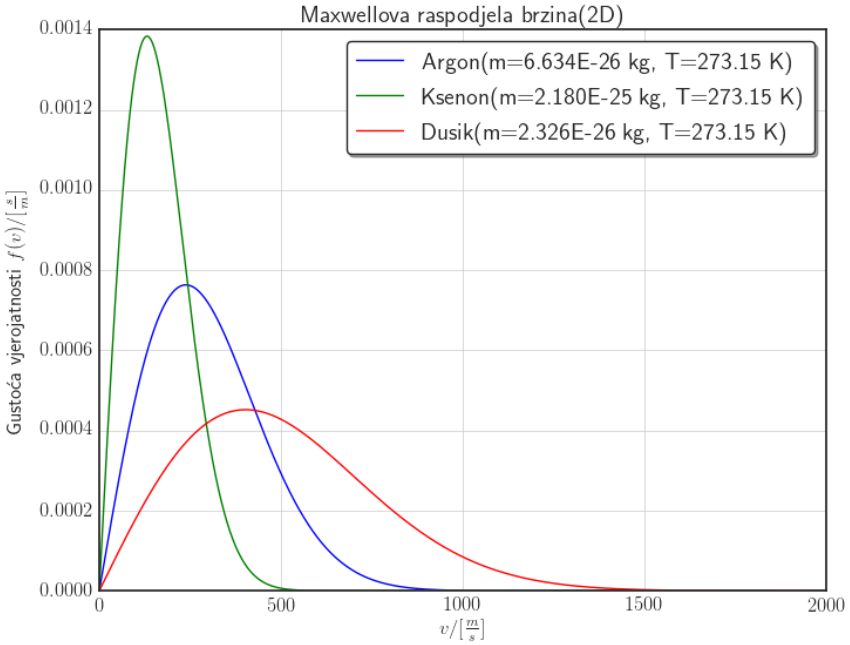
$$N_v dv = N 2\pi v \left(\frac{m}{2\pi k_B T}\right) e^{-mv^2/2k_B T} dv \tag{2.25}$$

U tom slučaju broj čestica koji ima brzinu u intervalu $[v_1, v_2]$ dan je izrazom 2.26. Ako izraz 2.26 integriramo od 0 do $\infty$, dobit ćemo ukupan broj čestica $N$ u sustavu. Na primjeru od $N = 10000$ čestica na slici 2.11 ukupna površina ispod krivulje predstavlja upravo ukupan broj čestica $N$ u sustavu.

$$\int_{v_1}^{v_2} N(v) dv \tag{2.26}$$




Slika 2.9: Raspodjela gustoće vjerojatnosti $f(v,T)$

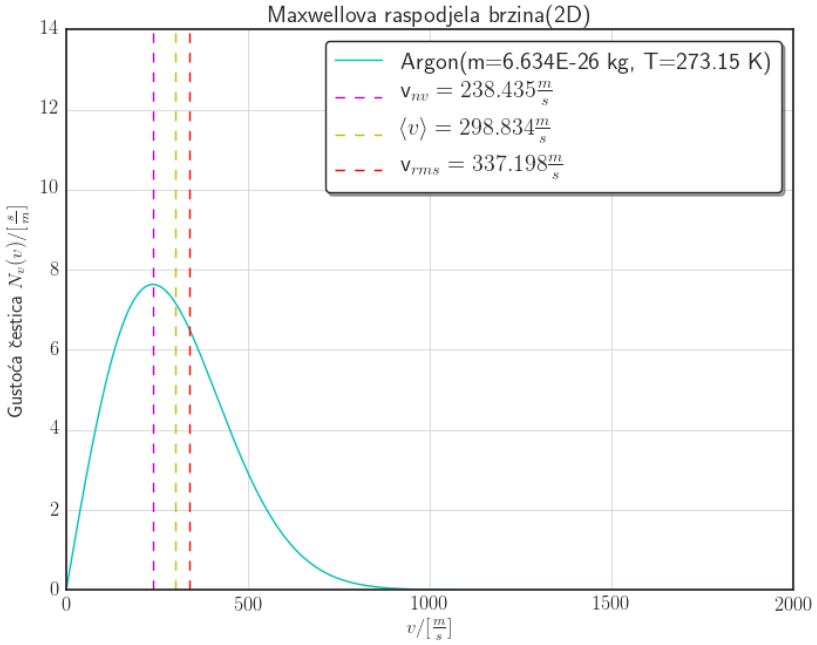



Slika 2.10: Raspodjela gustoće vjerojatnosti $f(v,m)$

## *2.7 Mjerene termodinamičke veličine*

### 2.7.1 Temperatura

Prema ekviparticijskom teoremu, srednja kinetička energija čestice po stupnju slobode iznosi $k_B T/2$, gdje je $k_B$ Boltzmannova konstanta, a $T$ je temperatura. Ako poopćimo




Slika 2.11: Raspodjela gustoće čestica $N(v)$, $\langle v \rangle$, $v_{nv}$, $v_{rms}$

tu relaciju da definiramo temperaturu u trenutku $t$, dobijemo relaciju

$$k_B T(t) = \frac{2}{d}\frac{K(t)}{N} = \frac{1}{Nd}\sum_{i=1}^{N} m_i \vec{v}_i \cdot \vec{v}_i \tag{2.27}$$

gdje je $K$ ukupna kinetička energija sustava, $\vec{v}_i$ je brzina čestice $i$ s masom $m_i$, a $d$ je prostorna dimenzija sustava. Temperatura koju mjerimo u laboratorijskom eksperimentu jest srednja temperatura koja odgovara vremenskom prosjeku od $T(t)$ preko mnogo konfiguracija čestica:

$$k_B T = \frac{1}{Nd}\sum_{i=1}^{N} m_i \overline{\vec{v}_i \cdot \vec{v}_i} \tag{2.28}$$

Veličina $\overline{Y}$ predstavlja vremenski prosjek veličine $Y(t)$. Relacija 2.28 je primjer veze makroskopske veličine (srednje temperature) s vremenskim prosjecima trajektorija čestica. U simulaciji stavljamo dodatni uvjet da nam je količina gibanja centra mase sustava jednaka nuli. Ne želimo da nam gibanje centra mase sustava mijenja temperaturu. U tom slučaju sustav ima $Nd - d$ neovisnih komponenti brzine te sukladno tome korigiramo relaciju 2.28:

$$k_B T = \frac{1}{(N-1)d}\sum_{i=1}^{N} m_i \overline{\vec{v}_i \cdot \vec{v}_i} \tag{2.29}$$

Faktor $(N-1)d$ naspram $Nd$ u relaciji 2.29 je primjer korekcije sustava *konačne* veličine koja postane nebitna za veliki $N$ [2]. Za potrebe simulacije ćemo zanemariti tu korekciju.

### 2.7.2 Tlak

Tlak je fizikalna veličina koje je definirana kao iznos okomite komponente sile po jedinici površine. Uzimajući u obzir drugi Newtonov zakon, sila je povezana s tokom količine gibanja po jedinici vremena. Pomoću takve relacije bismo mogli odrediti tlak, ali ona koristi informacije od samo jednog dijela čestica koje prolaze kroz zamišljenu površinu u trenutku $t$ pa ćemo koristiti relaciju 2.30 koja uzima u obzir sve čestice u sustavu:

$$P(t)V = Nk_BT(t) + \frac{1}{d}\sum_{i<j} \vec{r_{ij}} \cdot \vec{F_{ij}}(t) \tag{2.30}$$

gdje je $\vec{r_{ij}} = \vec{r_i} - \vec{r_j}$, a $\vec{F_{ij}}$ je sila kojom čestica $j$ djeluje na česticu $i$. Drugi član u relaciji 2.30 (tzv. *Virial*) jest korekcija jednadžbe stanja idealnog plina uslijed međudjelovanja među česticama [2]. Srednji tlak $P = \overline{P(t)}$ dobivamo tako da izračunamo vremenski prosjek desne strane relacije 2.30. U poglavlju Virijalni razvoj i odabrana simulacija, kada ćemo proučavati odabran primjer u skali, koristit ćemo dimenzionalne SI(fra. Système International d'Unités) jedinice te sukladno tome računati tlak $P$ s pripadnom SI jedinicom. Izvod virijalnog teorema priložen u dodatku B.

### 2.7.3 Energija

Iako je u našem slučaju ukupna energija $E$ sustava konstantna, u simulaciji vršimo njezino mjerenje jer je to jedan od načina kako provjeravamo da li simulacija radi ispravno.

## *2.8 Algoritmi integracije*

Iz razloga što sustav jednadžbi 2.8 nema analitičkog rješenja, gibanje čestica moramo riješiti numeričkim metodama. Kriterije koje zahtijevamo da dobar algoritam za numeričko integriranje zadovoljava su sljedeći:

- očuvanje volumena faznog prostora,
- konzistentnost s poznatim zakonima očuvanja,
- invarijantnost na vremenske translacije,
- točnost za relativno velike korake $\Delta t$.

U ovom diplomskom radu je korišten $Verlet$ algoritam za numeričko integriranje. On zadovoljava sve prethodno navedene kriterije i jednostavan je za implementaciju na računalu. Verlet algoritam za jednodimenzionalno gibanje:

$$x_{n+1} = x_n + v_n\Delta t + \frac{1}{2}a_n(\Delta t)^2 \tag{2.31}$$

$$v_{n+1} = v_n + \frac{1}{2}(a_{n+1} + a_n)\Delta t \tag{2.32}$$

Novi položaj je korišten da bismo izračunali novo ubrzanje $a_{n+1}$, što je korišteno zajedno s ubrzanjem $a_n$ da izračunamo novu brzinu $v_{n+1}$. Izvod Verletovog algoritma priložen u dodatku A. Drugi algoritam koji se često koristi je Runge-Kutta algoritam za numeričko integriranje [21]. Iako postoje mnoge varijante ona koje se najčešće koristi je RK4 varijanta. Ona radi na principu da $\frac{dy}{dt} = f(y,t)$ izvrijedni četiri puta unutar jednog koraka $h$. Za korak $h > 0$ imamo

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.33}$$

$$t_{n+1} = t_n + h \tag{2.34}$$

za $n = 0, 1, 2, 3, ...$

$$k_1 = f(t_n, y_n) \tag{2.35}$$

$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \tag{2.36}$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \tag{2.37}$$

$$k_4 = f(t_n + h, y_n + hk_3) \tag{2.38}$$

Različitim odabirom težinskih konstanti odgovaraju različite varijante algoritma RK4. Runge-Kutta metode zahtjevaju izvođenje većeg broj operacija no isto tako daju veću stabilnost rješenja.

## *2.9 Jedinice*

U simulaciji koristimo bezdimenzionalne (reducirane) jedinice i SI jedinice kada će sustav biti sastavljen od konkretne vrste čestica (npr. atoma argona). Bezdimenzionalne (reducirane) jedinice koristimo zbog pogodnosti prilikom programskog ostvarenja simulacije jer su sve veličine u toj formi jednostavnije za implementaciju na računalu te sukladno tome smanjena mogućnost pogreške prilikom programiranja simulacije. Sljedeći razlog je taj što na taj način možemo proučavati cijelu klasu problema i jednom kada izvršimo mjerenje, veličinu skaliramo na željene jedinice. Jedinice za udaljenost i energiju nam predstavljaju Lennard-Jonesovi parametri $\sigma$ i $\epsilon$, a jedinica za masu $m$ jest masa jedne čestice. Sve ostale veličine možemo izraziti preko tih jedinica. U tablici 2.1 su priložene neke od tih veličina [2]. Na primjer, ako simulaciju izvodimo s česticama argona (tablica 2.1), nakon 20000 koraka s vremenskim korakom $\Delta t = 0.005$, ukupno proteklo vrijeme u reduciranim jedinicama iznosi 100 ili $2.16 \cdot 10^{-10} s$ u SI jedinicama.

| **Veličina** | **Jedinica** | **Argon** |
|---|---|---|
| Duljina | $\sigma$ | 3.504 Å |
| Energija | $\epsilon$ | $1.625 \cdot 10^{-21}$ J |
| Masa | $m$ | $6.634 \cdot 10^{-26}$ kg |
| Vrijeme | $\sqrt{\frac{m \cdot \sigma^2}{\epsilon}}$ | $2.17 \cdot 10^{-12}$ s |
| Brzina | $\sqrt{\frac{\epsilon}{m}}$ | $1.57 \cdot 10^{2}$ m/s |
| Sila | $\frac{\epsilon}{\sigma}$ | $4.85 \cdot 10^{-12}$ N |
| Temperatura | $\frac{\epsilon}{k_B}$ | 120 K |
| Tlak | $\frac{\epsilon}{\sigma^2}$ | $1.43 \cdot 10^{-2}$ N/m |

Tablica 2.1: Bezdimenzionalne veličine i pripadne jedinice s numeričkim vrijednostima za argon

## *2.10 Simulacija molekularne dinamike u dvije dimenzije*

Primarni fokus nam je simuliranje rijetkih plinova na relativno visokoj temperaturi za koje Lennard-Jonesov potencijal daje najbolje rezultate. Iako nećemo proučavati pojedina stanja materije (osim plinovitog) i *fazne prijelaze* između njih, demonstrirat ćemo da i tako relativno jednostavan potencijal omogućava njihovo kvalitativno proučavanje. U ovom poglavlju, gdje proučavamo općenito kvalitativno ponašanje sustava, koristimo bezdimenzionalne jedinice iz razloga navedenih u potpoglavlju jedinice. Proces simuliranja grubo dijelimo u tri faze:

- početni uvjeti,
- termodinamička ravnoteža,
- mjerenje.

### 2.10.1 Početni uvjeti

u toj fazi odlučujemo o inicijalnim postavkama sustava koji želimo simulirati:

- $N$ - broj čestica,
- početni položaj čestica,
- $\rho$ - gustoća sustava,
- $\frac{U_{kin-ini}}{N}$ - početna kinetička energija po čestici,
- početna raspodjela brzina čestica,
- $\Delta t$ - vremenski korak integracijskog algoritma,
- $r_{cutoff}$,

- "koža",
- usrednjavanje.

***Broj čestica*** - tipčan broj čestica s kojim barataju današnje simulacije molekularne dinamike se kreće između $10^4 - 10^8$ čestica. To stavlja ograničenje na pojave koje možemo proučavati u smislu da to što želimo proučavati se može "sastaviti" od dotičnog broja čestica. S tim brojem čestica tipična prostorna veličina sustava je reda veličine od nekoliko 10 nm. U našem slučaju to neće predstavljati problem jer ćemo proučavati rijetke plinove koji već s malim brojem čestica ($\sim 10^3$) daju ne samo kvalitativne već i kvantitativne rezultate.
***Početni položaj čestica*** - u našem slučaju, gdje ne proučavamo nikakvu kompleksnu strukturu sastavljenu od više čestica, to će se svesti na jednostavnu kvadratnu matricu položaja čestica zbog minimizacije zakazivanja integracijskog algoritma.
***Gustoća sustava*** - parametar s kojim za zadan broj čestica definiramo granice sustava. U poglavlju Virijalni razvoj i odabrana simulacija, gdje proučavamo sustav sastavljen od pojedine vrste čestica, ta veličina je definirana kao količnik ukupne mase čestica i površine sustava s pripadnom SI jedinicom $\frac{kg}{m^2}$.
***Početna kinetička energija po čestici*** - parametar s kojim, u slučaju bezdimenzionalnih veličina, zadajemo početnu temperaturu sustava (Boltzmannova konstanta $k_B = 1$).
***Početna raspodjela brzina čestica*** - s tim parametrom odlučujemo o početnoj raspodjeli brzina čestica (Uniform, Gauss, Maxwell)
***Vremenski korak integracijskog algoritma*** - maksimalni iznos tog parametra je ograničen brzinom pojave koju proučavamo. To povlači da bi minimalni broj "uzoraka", između dva stanja pojave koju pročavamo, trebao biti dovoljno velik da možemo donositi zaključke o pojavi (npr. 100 uzoraka). Tipičan iznos u simulacijama molekularne dinamike 0.005 (u bezdimenzionalnim jedinicama) što na našem primjeru s česticama argona iznosi ~ 10 fs. Ako uzmemo u obzir da se tipičan broj koraka kreće između $10^4 - 10^8$ to predstavlja vremenski limit za proučavanje pojava koje se događaju u vremenskom intervalu koji nije veći od ~ 100 ns.
$\boldsymbol{r_{cutoff}}$ - udaljenost (najčešće $2.5-3\sigma$ ) na kojoj je doprinos potencijalu skoro jednak nuli. Uvedeno zbog optimizacije brzine izvođenja na računalu. Doprinos ukupnoj energiji sustava od čestica koje nisu uzete u obzir (udaljenost veća od $r_{cutoff}$) jest vrlo malena ali pod raznim okolnostima (npr. velika gustoća, visoka temperatura, velik broj koraka) ne i zanemariva. Zbog toga s tim parametrom treba biti oprezan jer s premalenom vrijednošću može narušiti očuvanje ukupne energije sustava. Kako općenito riješiti potencijalni problem oko očuvanja ukupne energije sustava, možemo vidjeti u potpoglavlju Potencijalne optimizacije (4.2).
***Usrednjavanje*** - s tim parametrom odlučujemo koliko posljednjih mjerenja uzimamo u obzir prilikom izračunavanja veličine (N-usrednjavanje preko svih koraka)
***Koža*** - još jedan optimizacijski parametar s kojim indirektno odlučujemo koliko često

će se izrađivati lista susjednih čestica.
O svim tim parametrima i njihovim implementacijama će biti više riječi kasnije kroz primjere (Primjer 1 (2.10.5), Primjer 2 (3.2)) i u poglavlju Implementacija na računalu (4).

### 2.10.2 Termodinamička ravnoteža

Iako je općenito teško točno definirati kada je sustav u termodinamičkoj ravnoteži, u našem slučaju, gdje je sustav izoliran od okoline (nema izmjene energije ni mase), odgovor je sasvim jasan. Drugi zakon termodinamike kaže da izolirani sustav koji je u termodinamičkoj ravnoteži ima maksimalnu entropiju. Međutim, iz razloga što računanje te veličine u simulacijama molekularne dinamike je pothvat sam po sebi, u okviru ovog diplomskog rada zadovoljit ćemo se sa sljedećom definicijom termodinamičke ravnoteže:
Sustav je u termodinamičkoj ravnoteži ako se mjerljive veličine (temperatura, tlak, potencijalna energija, kinetička energija) ne mjenjaju u vremenskom intervalu tokom kojeg proučavamo željenu pojavu. Iz razloga što navedene veličine zapravo fluktuiraju (iz raznih razloga koje ćemo navesti i obrazložiti u sljedećem potpoglavlju) oko neke srednje vrijednosti, za praktične potrebe simulacije ćemo se zadovoljiti ako se vrijednost mjerene veličine nalazi u željenom intervalu tokom izvođenja simulacije. Na primjeru koji sljedi će to postati jasnije.

### 2.10.3 Mjerenje

Prilikom "konvencionalnog" mjerenja veličinu nikad ne izmjerimo s apsolutnom preciznošću te su pogreške u mjerenju neizbježne. Mjerenja koja se vrše u simulacijama molekularne dinamike isto nisu izvedena s apsolutnom preciznošću. Jedan od razloga je taj što računalo ima konačnu preciznost pa je zbog toga rezultat zaokružen na konačan broj. Takve pogreške možemo minimizirati, ali ih ne možemo izbjeći i to radimo na način da veličinu više puta izmjerimo i rezultate mjerenja statistički obradimo. Sljedeći razlog nije pogreška ali nas isto tako onemogućuje da veličinu odredimo potpuno točno. U našem slučaju, gdje su broj čestica, volumen i unutarnja energija zadani, veličine poput temperature fluktuiraju oko određene vrijednosti. Za temperaturu razlog leži u njenoj definiciji (2.28) te sukladno tome i takvu veličinu statistički obrađujemo . Sva mjerenja veličina smatramo neovisnima i prikazana su u formi $(\langle X\rangle \pm M_n)$ gdje $\langle X\rangle$ predstavlja srednju vrijednost mjerene veličine, a $M_n$ nepouzdanost definiranu kao:

$$M_n = \sqrt{\frac{\sum_{i=1}^{n}\left(x-\bar{x}\right)^2}{n\,(n-1)}} \tag{2.39}$$

gdje je $n$ broj izvršenih mjerenja. Srednje vrijednosti svih veličina zaokružujemo na prvu sigurnu znamenku nepouzdanosti mjerene veličine.

### 2.10.4 LAMMPS

Prije nego počnemo s primjerima simuliranja, objasnit ćemo načine provjere rezultata mjerenja i općenito ispravnosti rada simulacije. Jedan od načina koji ćemo koristiti je taj da ćemo rezultate mjerenja dobivenih simuliranjem usporediti s rezultatima mjerenja dobivenih klasičnim eksperimentom. To ćemo učiniti u poglavlju Virijalni razvoj i odabrana simulacija. Drugi način je taj da ćemo usporediti naše rezultate mjerenja s rezultatima druge simulacije izvršene u profesionalnoj aplikaciji za simuliranje molekularne dinamike LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [8]. To ćemo učiniti tako da ćemo iz naše simulacije eksportirati, u datoteku, sve relevantne parametre (početne uvjete) potrebne za izvršavanje simulacije na LAMMPS-u. Nakon izvedene simulacije na LAMMPS-u rezultate importiramo u našu simulaciju te ih statistički obrađujemo i zajedno grafički prikazujemo.

### 2.10.5 Primjer

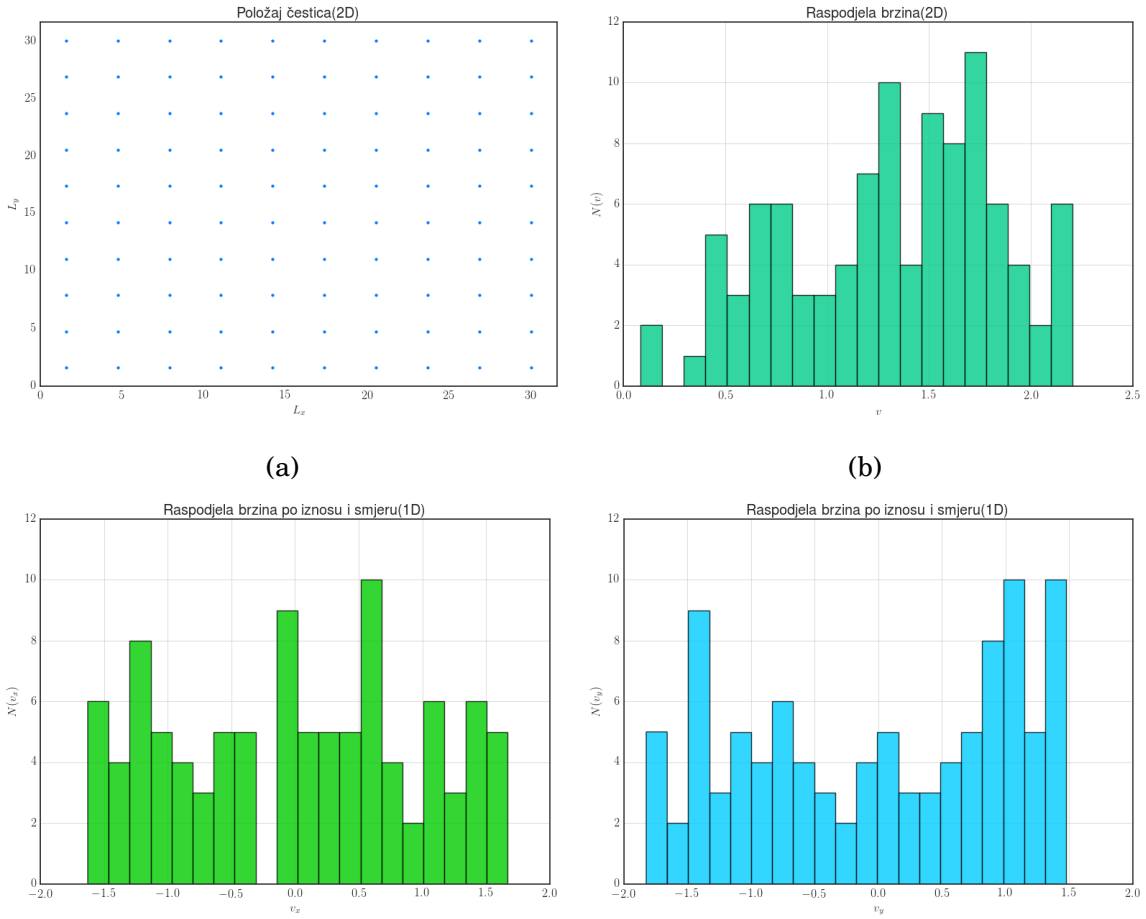
Na ovom primjeru demonstriramo i komentiramo mogućnosti i potencijalne probleme računalne simulacije molekularne dinamike na sustavu od stotinu čestica. Početni uvjeti su priloženi u tablici 2.2.

| **Početni uvjeti** | |
|---:|---|
| Broj čestica ($N$) | 100 |
| Korak ($\Delta t$) | 0.005 |
| Gustoća ($\rho$) | 0.01 |
| Temperatura ($U_{kin}/N$) | 1 |
| Početna konfiguracija čestica | $kvadratno$ |
| Početna raspodjela brzina | $uniform$ |
| $r_{cutoff}$ | 6 |
| Koža | 3 |
| Usrednjavanje | N |

Tablica 2.2: Primjer - početni uvjeti

Početni položaj čestica zajedno s početnom raspodjelom brzina po česticama prikazan je na slici 2.12. Inicijalna raspodjela brzina je uniformno (s jednakom vjerojatnošću) raspodjeljena po česticama u intervalu $(-0.5, 0.5)$ za $v_x$ i $v_y$. Zatim postavimo ukupnu količinu gibanja sustava na nulu te nakon toga skaliramo u skladu s početnom temperaturom (tj. početnom kinetičkom energijom po čestici).

Prije nego prijeđemo na sljedeću fazu, na potprimjeru ćemo demonstrirati zašto uopće trebamo usrednjavati veličine. Početni uvjeti su identični onima u tablici 2.2. Sva mjerenja veličina za potrebe ovog potprimjera (prikazanih na slikama 2.13, 2.14 a), b), 2.15 a), b)„ 2.16 a), b)) ne usrednjavamo te im je nepouzdanost mjerenja jednaka nuli (samo jedno mjerenje). Iz razloga što srednje vrijednosti mjerenih veličina




(a) (b)

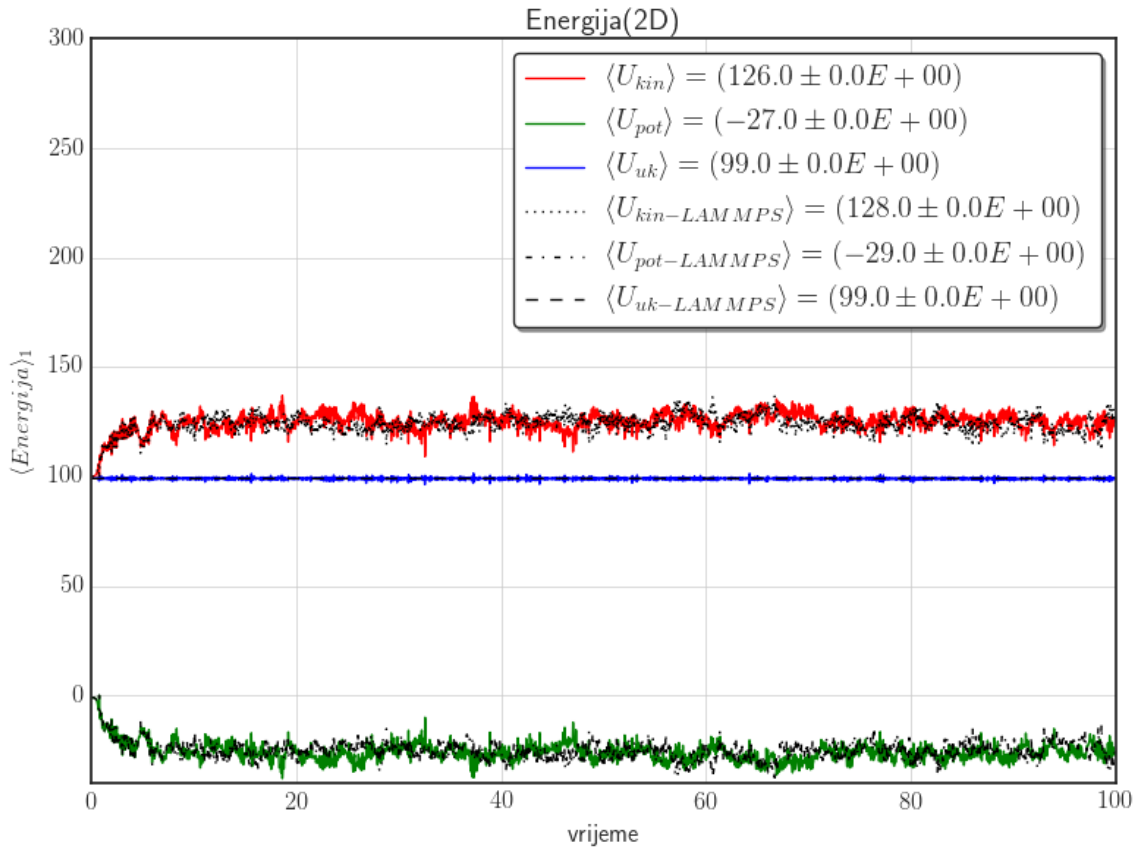
(c) (d)

Slika 2.12: Početna stanja sustava od 100 čestica

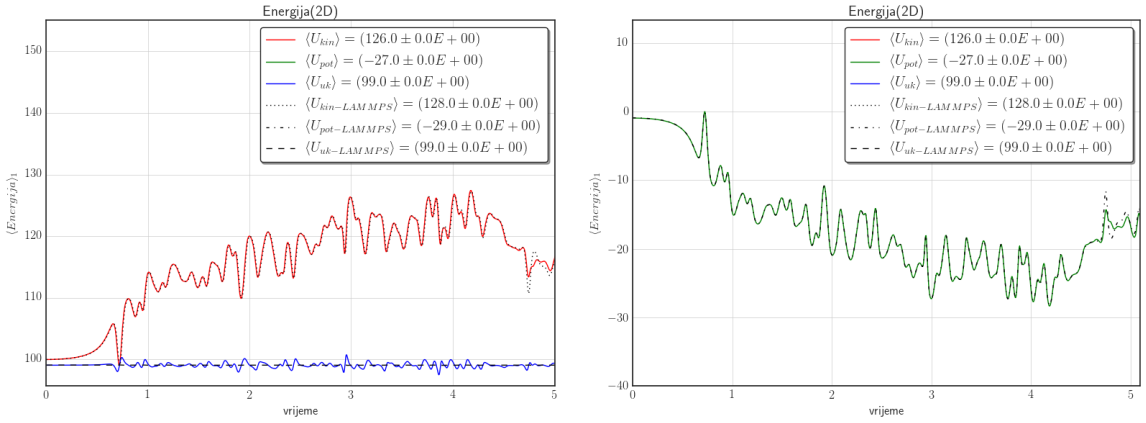
zaokružujemo na prvu značajnu znamenku nepouzdanosti (u ovom slučaju nula) sve veličine su sukladno tome automatski zaokružene. Na svim grafovima koje ćemo prikazati, a sadrže veličine koje usrednjavamo, možemo vidjeti na ordinati u indeksu veličine preko koliko posljednjih mjerenja vršimo usrednjavanje. Na slici 2.13 možemo vidjeti da za malen broj čestica ($N = 100$) fluktuacije kinetičke i potencijalne energije su značajne. Fluktuacije ukupne energije postoje zbog zaokruživanja na konačni broj znamenaka i one su slučajne prirode te se međusobno poništavaju. Jednako je i s temperaturom $T$ (sl.2.15) i tlakom $p$ (sl.2.16). Čak i kod relativno značajno većeg broja čestica, kao što ćemo vidjeti na kasnijim primjerima, fluktuacije su neizbježne te zbog toga veličine moramo usrednjavati.

Vraćamo se na prvi primjer. Na njemu vršimo usrednjavanje veličina preko svih koraka. U trenutku $t = 0$ energetsko stanje sustava je prikazano u tablici 2.3.

Vidimo da se gotovo sva energija nalazi u kinetičkoj energiji. To je posljedica inicijalno kvadratne raspodjele položaja čestica i malene gustoće sustava. Simulacija traje $t = 100$ vremena ili 20000 koraka ($\Delta t = 0.005$). Prvo ćemo obratiti pozornost na što se događa sa sustavom prvih par stotina koraka. Ono što želimo demonstrirati se zapravo najbolje vidi na primjeru bez usrednjavanja na slikama 2.14a, 2.14b, 2.15b i



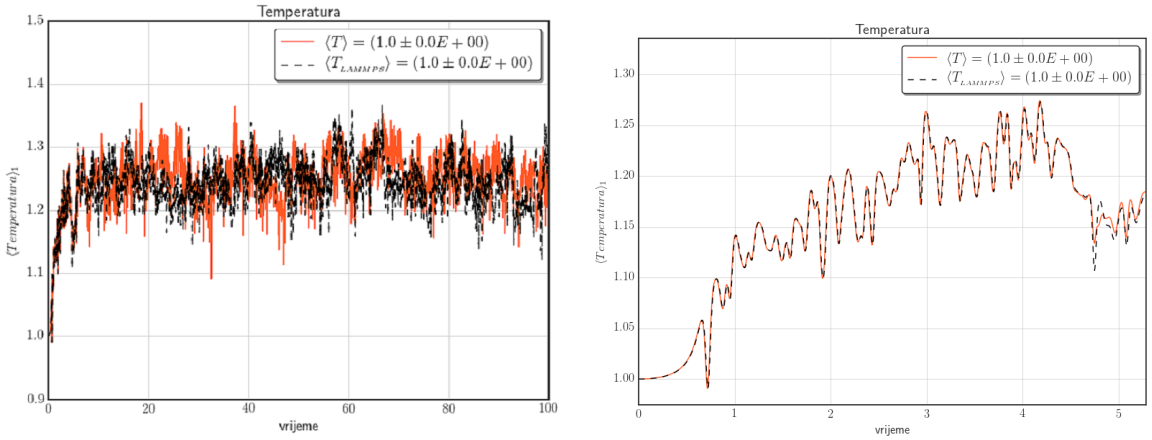

Slika 2.13: Usrednjavanje energije (za objašnjenje zašto je nepouzdanost nula vidi potprimjer u potpoglavlju Primjer 2.10.5)




(a) (b)

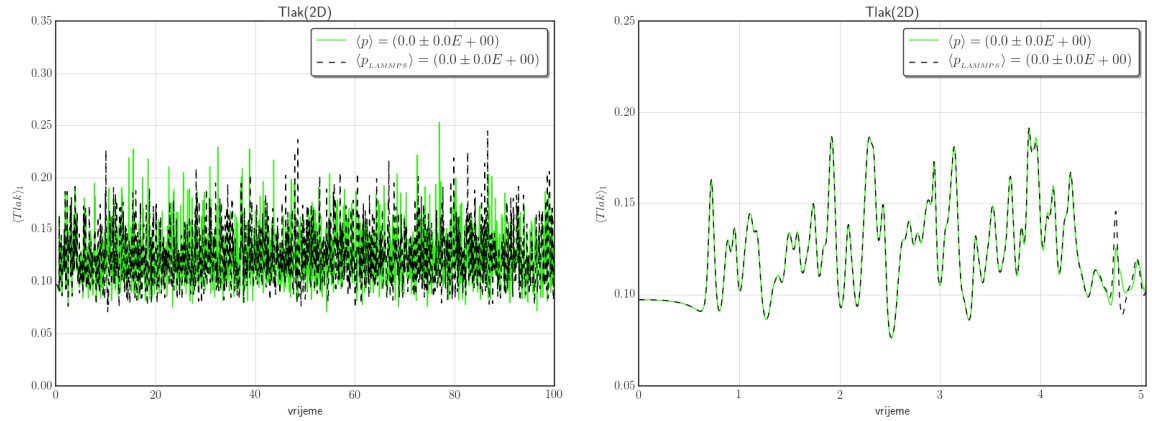
Slika 2.14: a), b) Usrednjavanje energije - vremenska skala (0-5) (za objašnjenje zašto je nepouzdanost nula vidi potprimjer u potpoglavlju Primjer 2.10.5)

2.16b. Ono što primjećujemo je da je evolucija našeg sustava čestica identična onoj iz LAMMPS sustava sve do $t \sim 4.8$. Tu možemo vidjeti osjetljivost sustava na početne uvjete. Ponavljamo, i jedna i druga simulacija počinju s identičnim početnim uvjetima do na pogrešku zaokruživanja. Zapravo, to je još vrlo dugo vrijeme slaganja, ~ 1000 koraka, samo iz razloga što je sustav malene gustoće i niske temperature. Dodatnim




(a) (b)

Slika 2.15: a) Usrednjavanje temperature, b) vremenska skala (0-5) (za objašnjenje zašto je nepouzdanost nula vidi potprimjer u potpoglavlju Primjer 2.10.5)




(a) (b)

Slika 2.16: a) Usrednjavanje tlaka, b) vremenska skala (0-5) (za objašnjenje zašto je nepouzdanost nula vidi potprimjer u potpoglavlju Primjer 2.10.5)

| | |
|---|---|
| $\langle U_{kin} \rangle$ | 100 |
| $\langle U_{pot} \rangle$ | -1 |
| $\langle U_{uk} \rangle$ | 99 |
| $\langle U_{kin-LAMMPS} \rangle$ | 100 |
| $\langle U_{pot-LAMMPS} \rangle$ | -1 |
| $\langle U_{uk-LAMMPS} \rangle$ | 99 |

Tablica 2.3: Energija - početno stanje

povećanjem preciznosti bi samo odgodili neslaganje, ali ne i eliminirali. Na te rezultate možemo gledati kao da smo simulaciju pokrenuli dva puta zaredom. U tom smislu one predstavljaju članove ansambla kojima želimo odrediti najvjerojatnije stanje sustava s obzirom na zadane početne uvjete. Analogno zaključivanje možemo primijeniti i na putanje čestica. Unatoč tome što ne možemo odrediti apsolutno točnu vrijednost

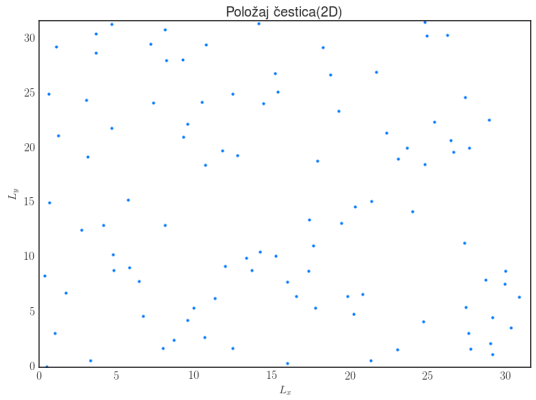
neke veličine u željenom trenutku, ta informacija nije bezvrijedna. Vrijednost takvih informacija o sustavu leži u njihovim prosječnim, a ne pojedinačnim vrijednostima. Ako pogledamo slike 2.13, 2.15a i 2.16a, na njima možemo odmah primijetiti da sve veličine nakon nekog vremena počinju fluktuirati u rasponu koji je manje više stalan u vremenu. Ako ih usrednjimo, kao što smo učinili u ovom primjeru, slike 2.17, vidimo da poprimaju relativno dobro definiranu vrijednost. Ono što se i ovdje primjećuje (sl. 2.17) je da iako ih usrednjavamo preko svih koraka i dalje su fluktuacije izražene. Ako želimo imati dobro definirane veličine tokom simulacije, to možemo učiniti na dva načina. Jedan je taj da povećamo broj čestica, dok je drugi da akumuliramo dovoljno mnogo mjerenja da nam srednja vrijednost veličine bude definirana s dovoljnom preciznošću. Oba rješenja su u konačnici skupe u smislu računalnih resursa. Dok nas prva košta manje u simulacijskom vremenu, to kompenzira skupljim procesorskim vremenom i obratno za drugi način. To nas prisiljava da primijenimo kompromisno rješenje, povećamo broj čestica i usrednjavamo rezultate toliko da promatranu pojavu možemo odsimulirati u razumnom vremenu sa željenom točnošću. To ćemo i učiniti na primjeru u potpoglavlju Primjer 2 (3.2). Međutim, čak i ako su nam vrijednosti veličina dobro definirane prilikom simuliranja, i dalje trebamo biti oprezni kada tu informaciju o sustavu ima smisla koristiti. Primjerice, iako temperaturu računamo u svakom trenutku tokom simualcije, ta veličina je općenito valjana tek kada je sustav u termodinamičkoj ravnoteži. Zato, prije nego pristupimo bilo kakvom mjerenju, moramo biti sigurni da je sustav u termodinamičkoj ravnoteži. Kao što smo već rekli u potpoglavlju 2.10.2, sustav ćemo smatrati termodinamički uravnoteženim ako mu mjerljive veličine fluktuiraju ispod zadanog praga. Konkretno, u simulaciji smo to riješili tako da veličinu izmjerimo u trenutku $t_1$ i u nekom kasnijem trenutku $t_2$ te provjerimo da li se veličina u trenutku $t_2$ nalazi unutar željene tolerancije. Koliko točno vremena treba proći između dva uzorka veličine, ovisi prije svega o tome koliko želimo biti sigurni da se sustav nalazi u termodinamičkoj ravnoteži. Naravno, moguće su razne komplikacije u toku simuliranja. Primjerice, prešutno pretpostavljamo da se između ta dva trenutka mjerena veličina ne mijenja previše jer u suprotnom se može dogoditi da se veličina nađe slučajno unutar intervala tolerancije. Protumjera za takav slučaj je povećati broj intervala nad kojima provjeravamo toleranciju te zahtjevati da se u "n" intervala veličina nalazi unutar zadane tolerancije. Uostalom, uvijek možemo grafički prikazati evoluciju svih veličina u sustavu od interesa te iz toga pokušati izvući korisne zaključke. Veličine koje mi možemo pratiti u toku simuliranja te provjeravati da li se sustav nalazi u termodinamičkoj ravnoteži su temperatura, tlak, ukupna potencijalna energija, ukupna kinetička energija, gustoća te raspodjela brzina. Dok je postupak za prve četiri veličine jasan iz prethodnog primjera, gustoću ćemo sada pojasniti. Kako točno provjeravamo posljednji kriterij, raspodjela brzina, će biti jasno na primjeru u potpoglavlju Primjer 2 (3.2). Ono što kvalitativno, u ovom primjeru, od njih očekujemo je to da brzine čestica sustava u termodinamičkoj ravnoteži u 1D po iznosu i smjeru poprime formu normalne (Gaussove) raspodjele oko $v_x = 0$ i $v_y = 0$, dok u 2D

poprime formu Maxwellove raspodjele brzina prikazanu na slici 2.11. Ako na slikama 2.12 i 2.17 pogledamo raspodjele brzina čestica iz ovog primjera, vidimo da je broj čestica premalen za izvlačenje bilo kakvih zaključaka. Dok ostale veličine na grafovima predstavljaju njihove usrednjene vrijednosti, grafovi položaja čestica i raspodjela brzina prikazuju trenutno stanje tih veličina. Gustoću kao kriterij primjenjujemo na sljedeći način. Sustav podijelimo na $k$ kvadranata i za svaki podsustav izračunamo gustoću. Korisna informacija o sustavu preko tog kriterija nam leži u standardnoj devijaciji srednje gustoće od $k$ podsustava, a ne u ukupnoj gustoći sustava jer nju zadamo sustavu na početku i ne mijenjamo tokom simuliranja. Ako pogledamo sliku 2.17 h) iz primjera na njoj prikazujemo evoluciju te veličine u vremenu. Na legendi u indeksu srednje standardne devijacije gustoće prikazujemo preko koliko posljednjih koraka smo ju usrednjavali. Iznad srednje gustoće sustava (puna crta), prikazujemo usrednjenu veličinu dok ispod prikazujemo njezino trenutno stanje. Provjeru da li se sustav nalazi u termodinamičkoj ravnoteži radimo na srednjim vrijednostima veličina, a ne trenutnim. Konkretno, na našem primjeru sustav se nalazi u termodinamičkoj ravnoteži ako zadovoljava sljedeće kriterije:

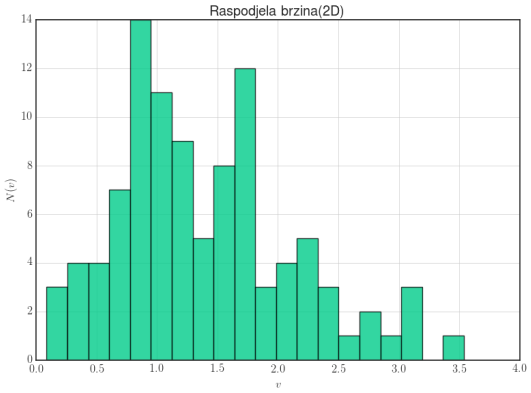
- temperatura: provjera svakih 2000 koraka, tolerancija 5%,
- tlak: provjera svakih 2000 koraka, tolerancija 5%,
- kinetička energija: provjera svakih 2000 koraka, tolerancija 5%,
- potencijalna energija: provjera svakih 2000 koraka, tolerancija 5%,
- gustoća: provjera svakih 2000 koraka, tolerancija 10%.

Po tim kriterijima naš sustav iz primjera je ušao u termodinamičku ravotežu u trenutku $t = 40$ te nakon toga možemo započeti s mjerenjem. U praksi, kao što ćemo i mi učiniti, to zapravo znači da smu ujedno i završili s mjerenjem jer su nam podaci preko kojih uopće odlučujemo da li se sustav nalazi u termodinamičkoj ravnoteži već statistički obrađeni. Sve te kriterije možemo kombinirati s njihovim raznim parametrima, međutim, može se dogoditi da ako pretjeramo sa zahtjevima da će simuliranje potrajati veoma dugo ili neće uopće moći zadovoljiti kriterije. Zato nam je korisno, što i radimo, pratiti u realnom vremenu u kojem se stanju sustav nalazi pa eventualno intervenirati te ublažiti neke kriterije ili, kao što je često slučaj, možemo povećati broj čestica jer su one uzrok (njihova malobrojnost) nemogućnosti zadovoljavanja kriterija. Alternativa je povećati interval nad kojim vršimo usrednjavanje veličina. Za kraj ćemo još prokomentirati konačna stanja pojedinih veličina sustava (slika 2.17). Ako obratimo pozornost na vrijednosti veličina iz naše simulacije i onih iz LAMMPS-a vidimo da zajedno konvergiraju prema istim vrijednostima. Iako smo temperaturu indirektno zadali na početku simulacije ($E_{kin}/N = 1$) na slici 2.17 f) možemo primjetiti da kada je sustav ušao u ravnotežu, da je poprimila neku drugu vrijednost. To je

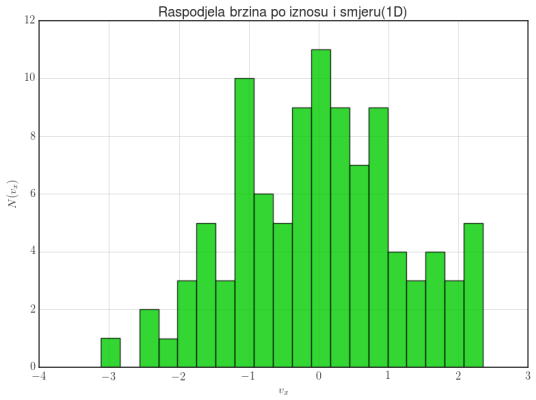
iz razloga što naš sustav čestica predstavlja mikrokanonski sustav kojemu je zadana $U_{uk}, N, V$ te sukladno tome temperatura fluktuira oko takve vrijednosti da naš sustav u termodinamičkoj ravnoteži može i imati te zadane $U_{uk}, N, V$ vrijednosti.



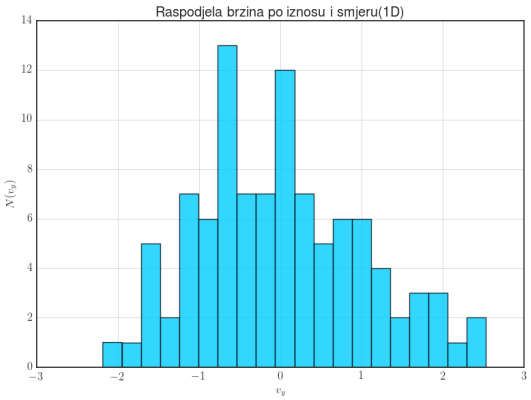

(a)



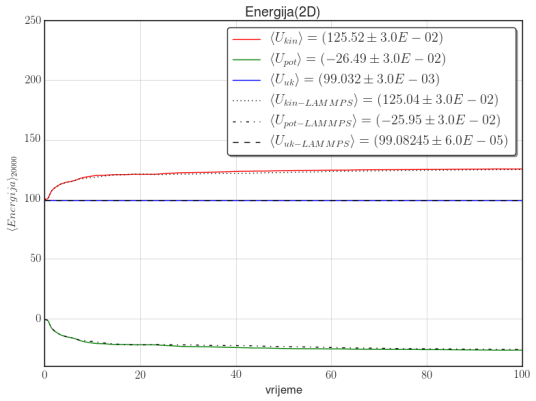

(b)



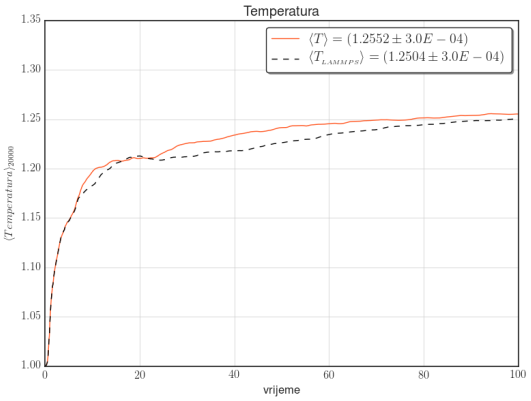

(c)



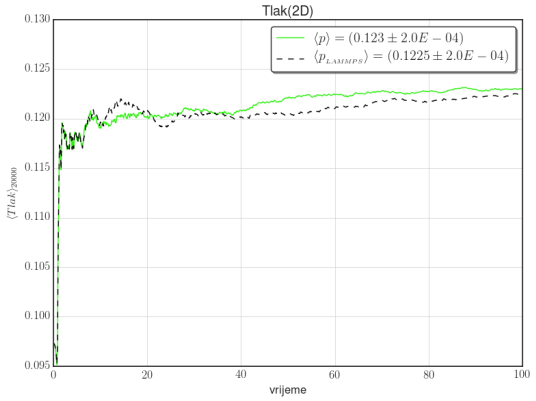

(d)



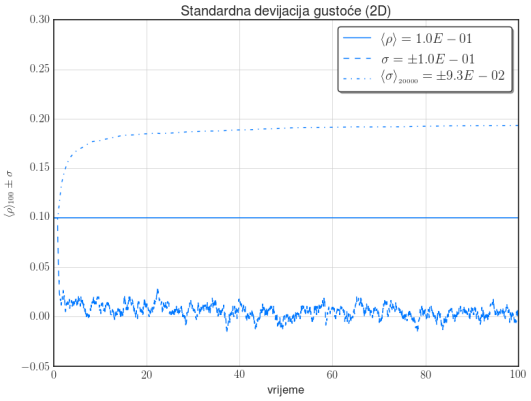

(e)




(f)




(g)

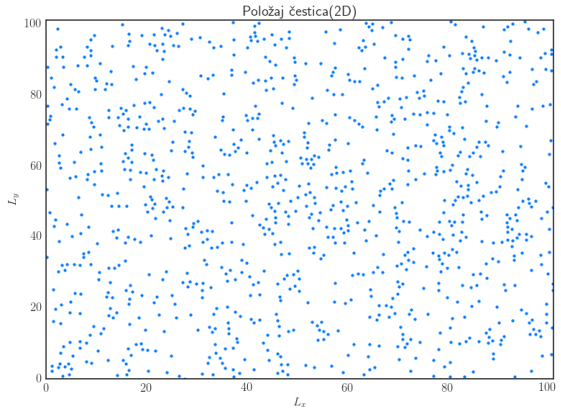



(h)

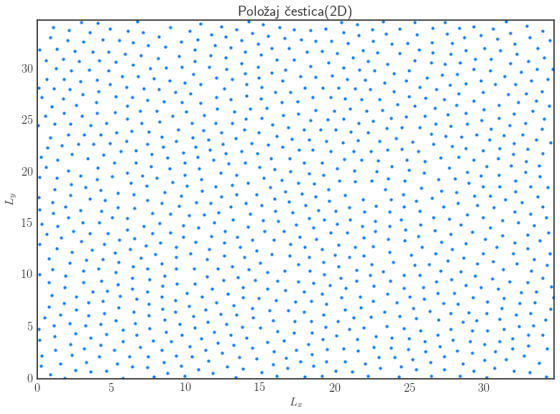
Slika 2.17: Evolucija sustava - konačno stanje

### 2.10.6 Kvalitativni prikaz agregatnih stanja

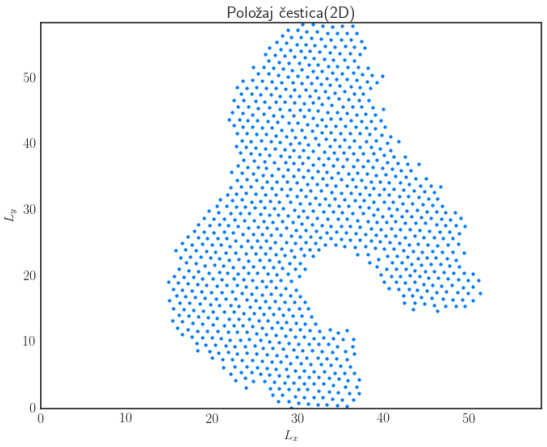
Na skupini slika 2.18 redom možemo vidjeti prikaz položaja čestica kada se sustav nalazi u pojedinom agregatnom stanju. Da prilikom njihove izrade ne bismo isprobavali razne početne uvjete te iz samog promatranja evoluiranja sustava u realnom vremenu donosili kvalitativne zaključke o stanju sustava, poslužili smo se Lennard-Jones faznim dijagramom[10]. Što bi bilo potrebno optimizirati za analitičko proučavanje faznih prijelaza vidjeti poglavlje Potencijalne optimizacije (4.2).



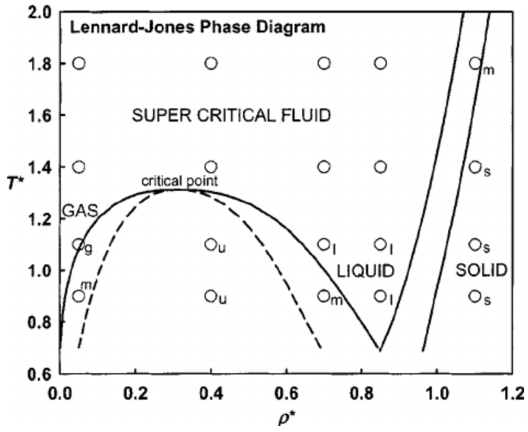

(a) Plinovito stanje




(b) Tekuće stanje




(c) Čvrsto stanje




(d) Fazni dijagram

Slika 2.18: a) Agregatna stanja Lennard-Jonesova sustava

[10]Slika Lennard-Jones faznog dijagrama preuzeta je s web odredišta <https://www.researchgate.net/figure/215775009_fig2_FIG-3-Phase-diagram-of-Lennard-Jones-systems-The-open-circles-represent-the-states>, [accessed 28 Nov, 2016] - Scientific Figure on ResearchGate

# 3 Virijalni razvoj i odabrana simulacija

## *3.1 Virijalni razvoj*

Počinjemo s činjenicom da se svojstva plinova, kada im je gustoća dovoljno mala, mogu opisati jednadžbom stanja idealnog plina 2.11. Zatim ćemo tu jednadžbu modificirati na takav način da ćemo moći opisati stanje plinova kada postoji međudjelovanje među česticama plina. Tako dobivena jednadžba stanja se zove *virijalna jednadžba stanja* (3.40), a postupak kojim ćemo do nje doći *virijalni razvoj*. U okviru ovog diplomskog rada zaustavit ćemo se na modifikacijama koje uključuju samo dvočestična međudjelovanja. Postupak se sastoji od toga da jednadžbu stanja idealnog plina razvijemo u red potencija gustoće.

$$\frac{p}{k_B T} = \frac{N}{V}\left(B_1 + B_2(T)\cdot\left(\frac{N}{V}\right) + B_3(T)\cdot\left(\frac{N}{V}\right)^2 + \cdots\right) \tag{3.40}$$

$B_n$ se zovu *virijalni koeficijenti* i njih se može analitički izvesti pomoću statističke fizike te tako opravdati taj postupak. Ti koeficijenti se mogu interpretirati kao broj čestica koji sudjeluju u međudjelovanju te tako virijalna jednadžba stanja 3.40, kada uzmemo u obzir samo prvi član reda pri čemu je $B_1 = 1$, predstavlja jednadžbu stanja idealnog plina 2.11 (čestice ne međudjeluju same sa sobom). Drugi član reda uzima u obzir dvočestično međudjelovanje, treći tročestično međudjelovanje, itd. Iako se općenito za proizvoljni potencijal ne može doći do analitičkog riješenja za virijalne koeficijente, za Lennard-Jonesov potencijal (2.5), preko klaster ekspanzije (knjiga [5], str. 492-502), možemo doći do analitičkog izraza 3.41[11] za drugi virijalni koeficijent. Ako pogledamo izraz 3.41, vidimo da je sva dimenzionalnost drugog virijalnog koeficijenta sadržana u parametru $\sigma$ koji u tom slučaju predstavlja promjer čestice te ga možemo interpretirati kao volumen po čestici.

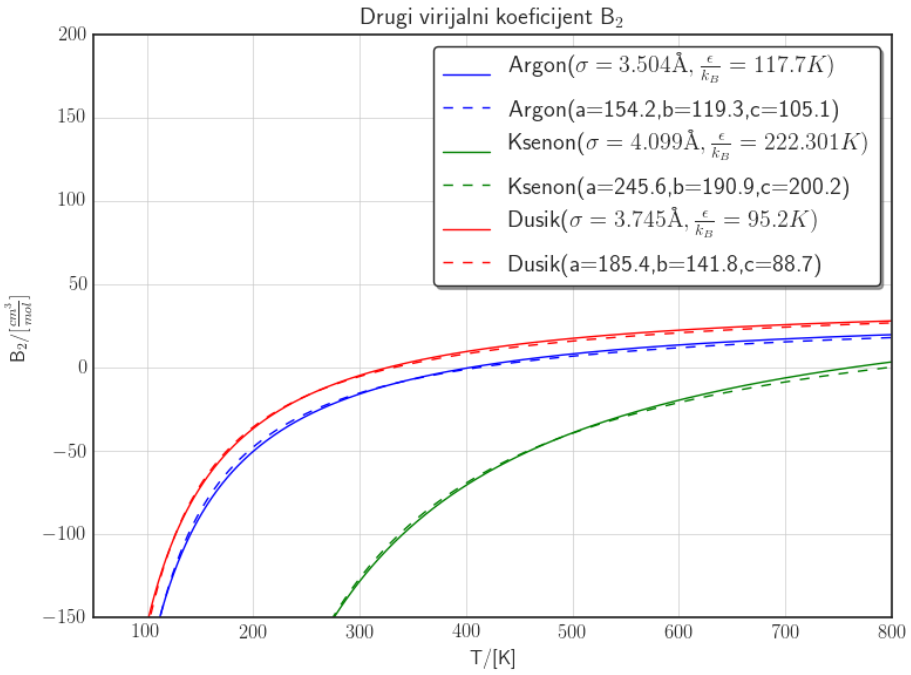
$$B_2(T)_{LJ} = \frac{2\pi\sigma^3}{3}\sum_{i=0}^{\infty} -\frac{2^{(i+1/2)}}{4i!}\cdot\Gamma\left(\frac{2i-1}{4}\right)\cdot\left(\frac{\epsilon}{k_B T}\right) \tag{3.41}$$

Na slici 3.19 prikazujemo drugi virijalni koeficijent za nekoliko vrsta čestica izračunat pomoću izraza 3.41 te drugi virijalni koeficijent $B_{2_{EF}}$ koji je dobiven eksperimentalnim "fitanjem" na izraz 3.42 s parametrima $a, b, c$ koji su priloženi na legendi [15].

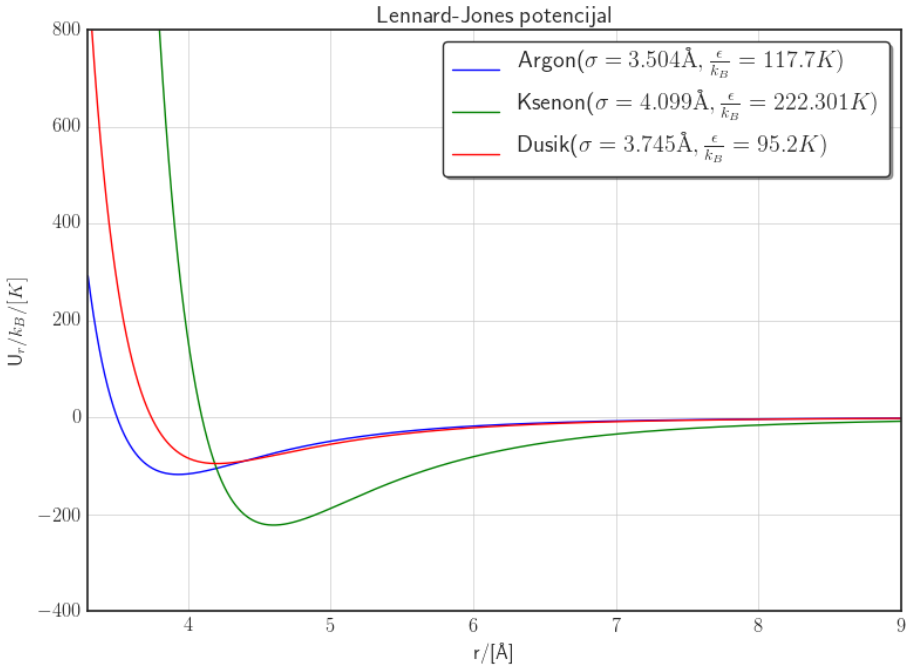
$$B_{2_{EF}} = a - b\cdot e^{c(K/T)} \tag{3.42}$$

Na slici 3.20 prikazujemo pripadni Lennard-Jonesov potencijal za dotičnu vrstu čestica.

[11] $\Gamma(x)$ predstavlja gama funkciju




Slika 3.19: $B_2$ za argon, ksenon, dušik




Slika 3.20: LJ potencijal za argon, ksenon, dušik

Vidimo da postoji temperatura na kojoj je drugi virijalni koeficijent jednak nuli te tada realni plin možemo tretirati kao idealni plin. Ta temeperatura se još zove Boyleova temperatura. Ispod te temperature $B_2$ je negativan te ako nastavimo s istom logikom interpretacije čestice imaju volumen te stoga umanjuju ukupno dostupan volumen. Iznad Boyleove temperature $B_2$ je pozitivan, što je onda posljedica da čestice mogu do neke mjere prodrijeti jedna u drugu, što je opet posljedica toga što Lennard-Jonesov potencijal nije krajnje strm. Za određenu tvar ju možemo izračunati pomoću

izraza $T_B = \frac{a}{bR}$ koji možemo izvesti, uz prethodno navedena ograničenja i uvjete, i iz Van der Waalsove jednadžbe (2.12) i iz virijalne jednadžbe(3.40). To je ujedno i jedan od razloga zašto se Van der Waalsova jednadžba tako dugo "zadržala" bez obzira na to što postoje znatno preciznije fenomenološke jednadžbe stanja.

## *3.2 Primjer 2*

Na ovom primjeru demonstriramo i verificiramo mogućnosti računalne simulacije molekularne dinamike na sustavu od deset tisuća čestica. Želimo provjeriti da li se eksperimentalno izmjerene veličine u simulaciji slažu s teoretski izračunatim vrijednostima. To ćemo pokušati učiniti tako da teoretski izračunamo odstupanje sustava od idealnog sustava, tj. sustava koji bi opisali idealnom jednadžbom plina 2.11. Početni uvjeti su priloženi u tablici 3.4.

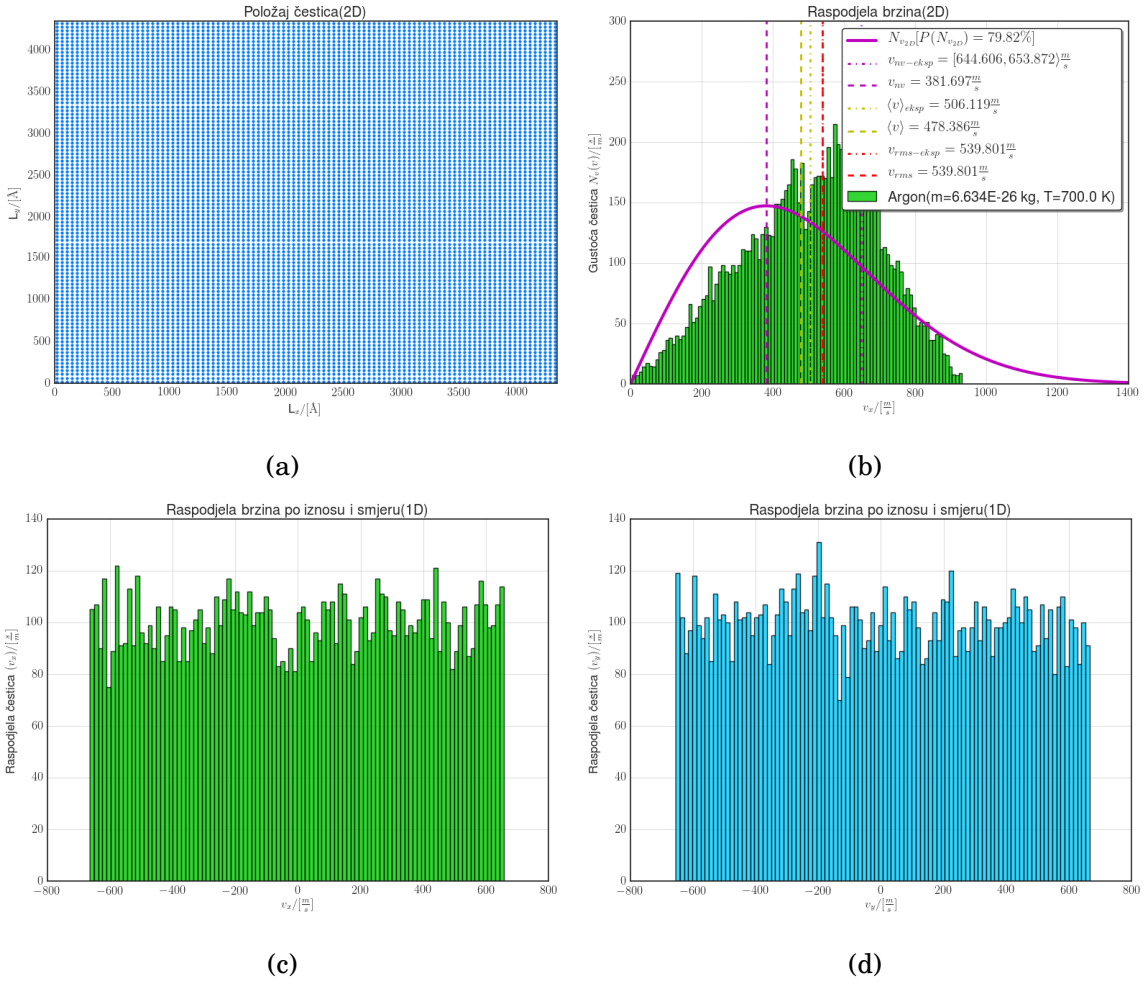
| **Početni uvjeti** | |
|---:|---|
| Broj čestica ($N$) | 10000 |
| Vrsta čestica | Argon |
| $m$ | $6.634 \cdot 10^{-26}$ kg |
| $\sigma$ | 3.504 Å |
| $\epsilon$ | $1.625 \cdot 10^{-21}$ J |
| Korak($\Delta t$) | 10 fs |
| Gustoća ($\rho$) | 10 kg/$m^3$ |
| Temperatura | 700 K |
| Početna konfiguracija čestica | kvadratno |
| Početna raspodjela brzina | uniform |
| $r_{cutoff}$ | 15 Å |
| Koža | 15 Å |
| Usrednjavanje | 10000 |

Tablica 3.4: Primjer 2 - početni uvjeti

Za razliku od primjera (2.10.5) iz prethodnog poglavlja, u ovom primjeru čestice su konkretne vrste, atomi argona, te su sve veličine zadane u SI jedinicama. U tom slučaju Lennard-Jonesov (LJ) potencijal je parametriziran s $m$, $\sigma$ i $\epsilon$ priloženim u tablici 3.4. i prikazanim na slici 3.20. Početna stanja sustava prikazana su na skupini slika 3.21.
Uvjeti koje je sustav morao zadovoljiti da bismo ga smatrali termodinamički uravnoteženim su sljedeći:

- temperatura: provjera svakih 2500 koraka, tolerancija 1%,
- tlak: provjera svakih 2500 koraka, tolerancija 1%,
- kinetička energija: provjera svakih 2500 koraka, tolerancija 1%,

- potencijalna energija: provjera svakih 2500 koraka, tolerancija 2%,
- gustoća: provjera svakih 2500 koraka, tolerancija 5%,
- raspodjela brzina: provjera svakih 2500 koraka, tolerancija 1%.




(a) (b)

(c) (d)

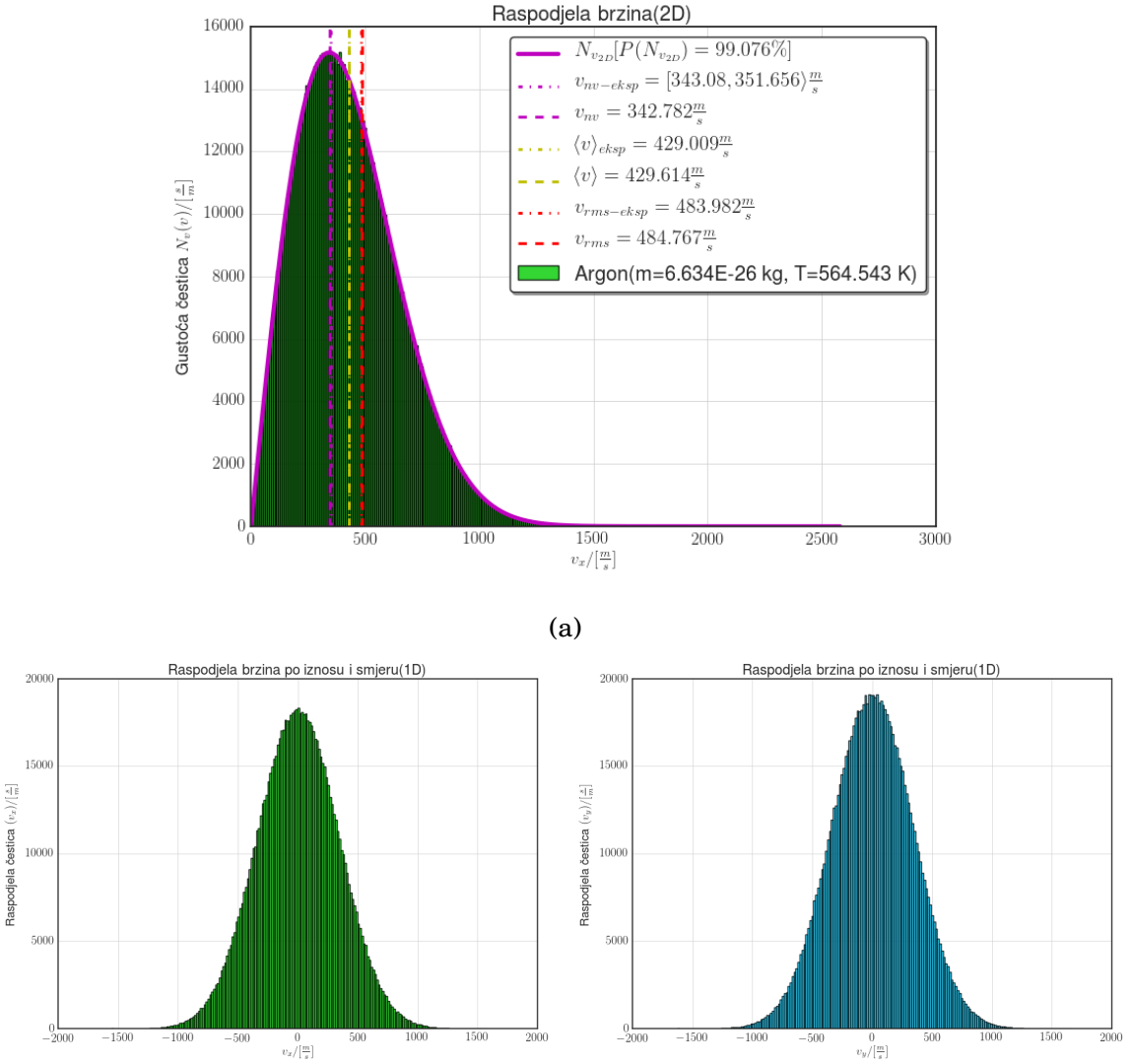
Slika 3.21: Početna stanja sustava sastavljenog od 10000 atoma argona

Posljednji uvjet, raspodjela brzina čestica, izveden je tako da teoretski, pomoću izraza 2.25, izračunamo očekivanu gustoću čestica u ovisnosti o brzini $N_{v2D}$ u intervalu $v+\Delta v$, gdje $\Delta v$ odgovara širini stupca na histogramu čiji broj zadajemo prilikom poziva metode za grafički prikaz prikaz (za detalje pogledati potpoglavlje Stvaranje programa 4.1.3). Zatim, u simulaciji, izračunamo koliko čestica ima brzinu u tako podijeljenim intervalima($\Delta v$) i na kraju usporedimo da li to odgovara očekivanoj teoretskoj vrijednosti u tom brzinskom intervalu. Grafički je to jasno vidljivo ako na svim brzinskim intervalima $\Delta v$ simulacijska vrijednost brzine čestice odgovara teoretskoj vrijednosti, onda će se svi "stupići" na histogramu nalaziti ispod teoretske krivulje na trenutnoj temperaturi sustava. To poklapanje teoretske i eksperimentalne vrijednosti na grafovima označavamo s $P(N_{v2D})$ i izražavamo u postocima. Iz razloga što je $\Delta v$ konačne širine $P(N_{v2D})$ nije nikad 100% ali, kako smanjujemo $\Delta v$, teži k 100%. Vrijednost $\Delta v$ je s jedne strane određena uvjetom za zadovoljavajućom točnošću numeričkog opisa,

a s druge strane brojem čestica. Drugim riječima, $\Delta v$ treba biti toliko širok da sadrži dovoljno čestica u intervalu $v + \Delta v$, a opet dovoljno uzak da može numerički opisati očekivanu teoretsku vrijednost veličine $N_{v2D}$. Za usporedbu, na ovom primjeru koji ima 10000 čestica, maksimalna vrijednost $P(N_{v2D})$ iznosi 97.157% (3.23 b)) dok na slici 3.22 možemo vidjeti sustav od milijun čestica te vidimo da je ta vrijednost bliža 100%. Oba sustava su krenula od jednolike raspodjele brzina čestica. Kratak odgovor na pitanje zašto onda nismo koristili milijun čestica prilikom simuliranja je prije svega iz razloga što bi predugo trajalo. Za potrebe te demonstracije sustav je "ugođen" početnim uvjetima (stotinu puta veća gustoća, dovoljno visoka temperatura, $\Delta t$...), mjeren je samo nužan skup veličina za to konkretno mjerenje i unatoč svemu tome bilo je potrebno pet sati izvođenja za $\sim 600$ koraka na trenutačno dostupnim računalnim resursima[12]. Za ovaj primjer nam je trebalo 30000 koraka i to je potrajalo oko sat vremena. Kao i kod svih prethodnih uvjeta termodinamičke ravnoteže, provjeravamo svakih $k$ koraka da li je ta veličina poprimila maksimalnu vrijednost. Za razliku od svih preostalih veličina, grafovi raspodjele brzina čestica sadrže trenutne brzine čestica u pojedinom koraku, a ne usrednjene veličine. Nakon što je sustav termodinamički uravnotežen, u granicama prethodno navedenih tolerancija, započinjemo s mjerenjem. Mjerenje je strogo gledajući, kao što smo već prije rekli, već izvršeno u toku provjera da li je sustav u ravnoteži. Jedina razlika od primjera 1 (2.10.5), po pitanju mjerenja, je ta da usrednjujemo preko posljednjih 2500 koraka a ne preko svih koraka, te se tako rješavamo akumuliranih pogrešaka u mjerenju koje su posljedica toga što su veličine izmjerene kada sustav još nije bio u termodinamičkoj ravnoteži. To rješavanje akumuliranih pogrešaka se najbolje vidi na grafu temperature (3.23 f)) u formi "skoka" na t~100000 fs iako postoji i kod svih ostalih mjerenih veličina. Na skupini slika 3.23, 3.24 prikazana je evolucija sustava čestica te njegovo konačno stanje. Možemo primijetiti da, iako je sustav sastavljen od 100 puta više čestica nego sustav iz primjera 1, i dalje su prisutne fluktuacije vrijednosti veličina, ali u ovom slučaju možemo izvlačiti ne samo kvalitativne već i kvantitativne zaključke o sustavu. Da bismo ispunili zahtjev za provjerom odstupanja sustava od idealnog u simulaciji izračunavamo veličinu *kompresijski faktor Z* koja je definirana kao $Z = \frac{pV}{Nk_BT}$ i veličinu $Z_{VIRIJAL}$ definiranu kao $Z_{VIRIJAL} = 1 + B_2(T) \cdot \left(\frac{N}{V}\right)$ pri čemu V zapravo predstavlja površinu sustava a ne volumen. Za slučaj kada se sustav čestica može tretirati kao idealni sustav obje te veličine su jednake jedinici. Na slici 3.24a) možemo vidjeti rezultate tih veličina. Iz priloženih rezultata vidimo da sustav čestica zadovoljava teoretski zahtjev da je sljedeći član iz virijalne jednadžbe stanja 3.40 mnogo manji od prethodnog. To je postignuto malenom gustoćom sustava te je vjerojatnost međudjelovanja više od dvije čestice veoma malena, što se u konačnici i vidi iz poklapanja izmjerenih rezultata. Analognim postupkom dolazimo do veličina tlak u 2D i 3D koje su prikazane redom na slikama 3.23g),h). Poopćenje na trodimenzionalne veličine u slučaju tlaka 3D iskorištava činje-

[12] za korištene računalne resurse vidi potpoglavlje Korišteni računalni resursi (4.3)

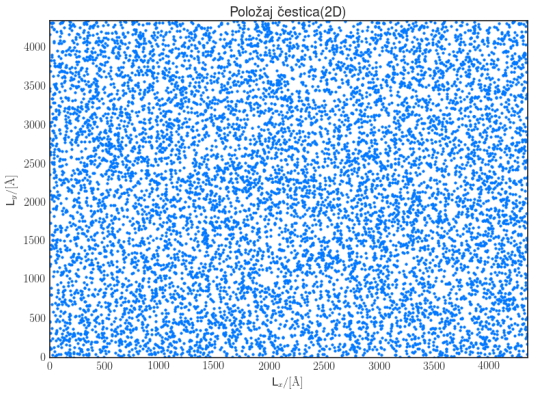
nicu da nam 2D slučaj predstavlja 3D sustav debljine $\sigma$ te su ujedno tako i izračunati rezultati za trodimenzionalni slučaj (3D). Isto takvu logiku smo iskoristili kada smo gustoću ovog sustava zadali u kg/$m^3$ bez obzira što tokom cijele simulacije zapravo baratamo s površinskom gustoćom. Zapravo nas ne treba čuditi tako dobro slaganje rezultata dobivenih računalnim simuliranjem i teorijskih rezultata. Razlog slaganja leži u tome što je sustav u računalnoj simulaciji okarakteriziran s parametrima $\sigma, \epsilon$ za Lennard-Jonesov potencijal koji su upravo ti isti $\sigma, \epsilon$ (za tu tvar) iz teorijskog izvoda za drugi virijalni koeficijent 3.41. Njihove numeričke vrijednosti su određene klasičnim eksperimentom raznim tehnikama (knjiga [5], str. 522, članak [13]). Podjednako dobre rezultate možemo očekivati sa svim česticama iz grupe plemenitih plinova. Helij je izuzetak zbog toga što su kod njega, u tim područjima mjerenja, značajni kvantni doprinosi.



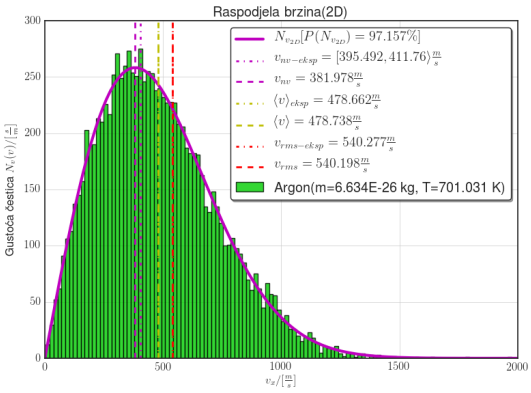

(a)

(b) (c)

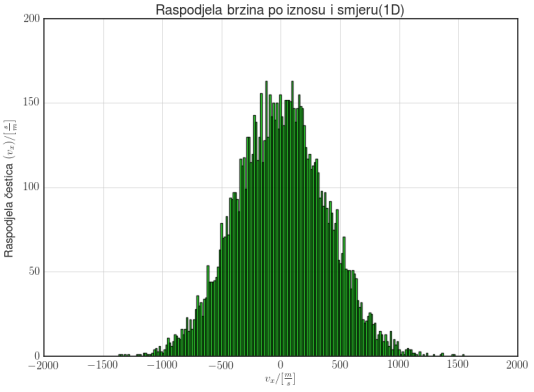
Slika 3.22: Konačno stanje sustava sastavljenog od milijun atoma argona



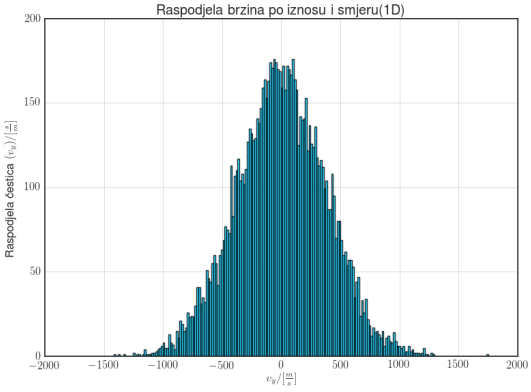

(a)



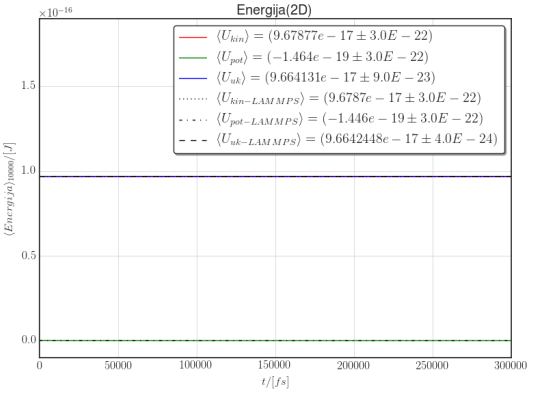

(b)



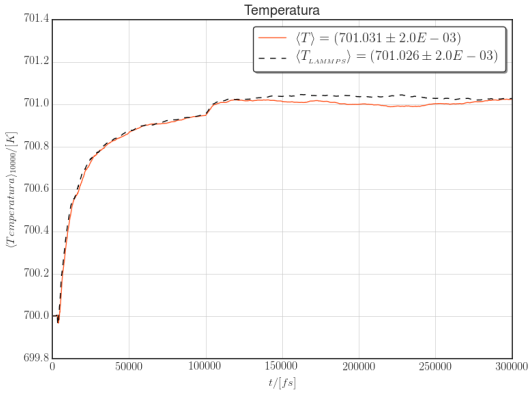

(c)



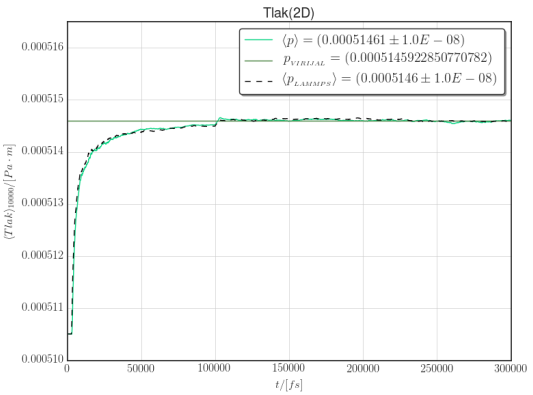

(d)



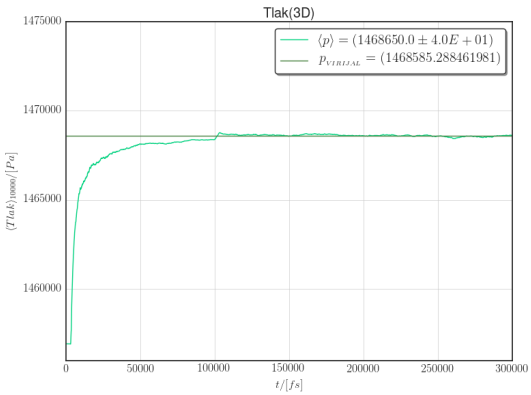

(e)




(f)

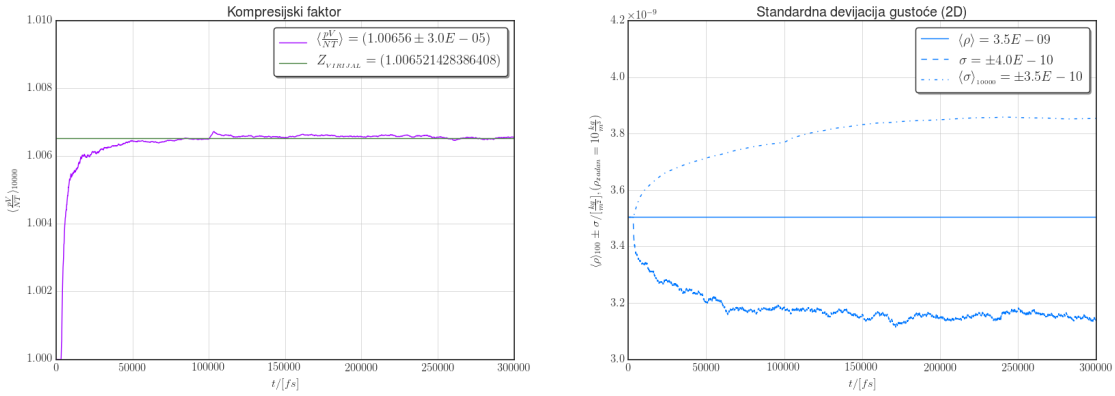



(g)




(h)

Slika 3.23: Evolucija sustava 10000 atoma argona - konačno stanje




(a) (b)

Slika 3.24: Evolucija sustava 10000 atoma argona - konačno stanje

# 4 Implementacija na računalu

## *4.1 Programsko ostvarenje simulacije*

Program za simuliranje molekularne dinamike na računalu je napisan u programskom jeziku Python korištenjem objektno orijentiranih principa. Korištena je programska platforma Anaconda koja uključuje integrirano okruženje za razvoj aplikacija (IDE) Spyder (Scientific PYthon Development EnviRonment) koje uključuje sve potrebne biblioteke za programsko ostvarenje naše aplikacije za simulaciju termodinamike međudjelujućih čestica. Najveći dio programskog koda napisan je korištenjem standardnih funkcija i struktura podataka koje su integrirane u sam programski jezik. U manjoj mjeri je korištena biblioteka NumPy koja proširuje funkcionalnost Pythona na višedimenzionalna polja i matrice te funkcije koje uključuju operacije nad njima. Grafički dio aplikacije se u potpunosti oslanja na biblioteku Matplotlib koje prije svega služi za grafički prikaz podataka. Programski kod je priložen uz diplomski rad u digitalnoj formi jer s preko četiri tisuće linija koda bi nepotrebno ispunio papirnatu formu diplomskog rada.

### 4.1.1 Objektno orijentirano programiranje

Prije nego opišemo strukturu aplikacije ukratko ćemo objasniti korištene objektno orijentirane principe i zašto su uopće korišteni. Intrizično svojstvo objektno orijentiranog programiranja (OOP) je podjela problema na *objekte*, koji manje-više sadrže podatke, te *metode* koje izvode operacije nad tim objektima. Takva podjela u konačnici vodi k većoj iskoristivosti i povećava čitljivost koda, a samim time i lakše održavanje i nadograđivanje jer je sva funkcionalnost objedinjena u jednu logičku cjelinu. Iako je inicijalno teže napisati objektno orijentiran program od klasičnog proceduralnog jer zahtijeva dosta planiranja unaprijed i na problem stavlja još jedan dodatni sloj apstrakcije, međutim, kada je problem koji želimo riješiti iole složeniji (i u konačnici glomazniji) takav pristup počinje vraćati uložen trud. U daljnjem dijelu teksta ćemo se ograničiti kako je to riješeno u Python programskom jeziku iako su osnovni principi slični i kod drugih objektno orijentiranih programskih jezika. U Pythonu su svi gradivni elementi OOP-a objekti i načelno se sastoje od dvije komponente; *atributa* (podataka) koji predstavljaju opis karakteristika objekta i *metoda* (generatora međudjelovanja) [14]. Metode predstavljaju funkcije definirane unutar klase čiji je maksimalni doseg sama klasa, a njihova svrha je izvršavanje operacija nad atributima tog objekta. Da bismo problem smatrali riješenim objektno orijentiranim pristupom, uz već navedeni koncept objekt-metoda, programsko rješenje bi trebalo koristiti sljedeće principe (mehanizme):

- **Učahurivanje** (Enkapsulacija)
- **Apstrakcija**
- **Nasljeđivanje**

- **Polimorfizam**

*Učahurivanje* predstavlja princip "skrivanja" implementacijskih detalja pojedine klase od objekata. Drugim riječima, učahurivanje je mehanizam za restrikciju pristupa nekim komponentama objekta, što zapravo znači da ne možemo vidjeti internu reprezentaciju objekta izvan njegove definicije. *Apstrakcija* predstavlja pojednostavljenje kompleksne realnosti modeliranjem klasa pogodnim za problem od interesa. Iako nije sasvim očito, konkretna realizacija apstrakcije je postignuta primjenom principa učahurivanja te u tom smislu predstavlja skoro sinonim učahurivanja. *Nasljeđivanje* predstavlja mehanizam stvaranja novih klasa pomoću već definiranih klasa. Preko tog mehanizma je ostvarena smanjena redudantnost programskog koda i samim time i veća efikasnost. *Polimorfizam* je mehanizam kod kojeg rezultat korištenih metoda ovisi o vrsti podataka. U praksi to znači da ako klasa B, koja je naslijeđena od klase A, koristi neku metodu $m_B$ nad podacima P, može izvršiti različite operacije nad podacima P od te "iste" $m_A$ metode (definirane istim ključnim imenom) u klasi A. Konkretna realizacija struktura u programskom jeziku koja ostvaruje taj objekt-metoda pristup, omogućuje mehanizme učahurivanja, nasljeđivanja i polimorfizma jest **klasa**. Konkretna realizacija objekta iz pojedine klase se zove ***instanca***. Sasvim konkretna realizacija prethodno navedenih principa u ovom radu je ostvarena prilikom rješavanja problema računalnog opisa termodinamike međudjelujućih čestica čiji je programski kod priložen uz pisani dio diplomskog rada.

### 4.1.2 Struktura programa

Program je strukturalno podijeljen na pet modula; md_pocetni_uvjeti, md, md_SI, Virijal i MBdist, koji u tom smislu predstavljaju najpopćenitije organizacijske jedinice aplikacije. Modul md_pocetni_uvjeti sadrži klasu md_BD_pocetni_uvjeti koja dalje sadrži razne atribute potrebne za rad simulacije od kojih je jedan od najvažnijih *stanje sustava cestica* koji sadrži podatke o trenutnim položajima i brzinama čestica. Klasa md_BD_pocetni_uvjeti isto tako sadrži metode za inicijalizaciju početnih položaja čestica (npr. *pocetni polozaj cestica kvadratno*), inicijalizaciju početnih brzina čestica (npr. *pocetna brzina cestica uniform*) te metode za operacije nad podacima iz i za LAMMPS aplikaciju. Modul *md* sadrži klase md_BD_integrator te md_BD. Prva od navedenih klasa naslijeđuje sve od klase *md_pocetni_uvjeti* te ju proširuje s rješenjima za probleme numeričke integracije, dinamike gibanja čestica te izračunom i održavanjem stanja termodinamičkih veličina. Iako su svi atributi i metode neophodni za rad simulacije spomenut ćemo samo neke od njih koji su od esencijalne važnosti. Metoda *akceleracija_cestica* koja , u našem slučaju za čestice u Lennard-Jonesovom potencijalu, izračunava njihove akceleracije, te metoda *izracunavanje polozaja i brzina Verlet* koja izračunava položaje i brzine čestica pomoću Verletovog algoritma za numeričko integriranje. Izvod Verletovog algoritma je priložen u dodatku A. Zatim metoda *izrada_liste_susjeda*, koja zajedno s metodom *susj_cel* (i još nekoliko metoda),

implementiraju Verletovu listu susjeda u kombinaciji s listom ćelija susjeda kojom je inicijalno kvadratna ovisnost vremena izvođenja o broju čestica linearizirana. Bez njih bi bilo praktički nemoguće izvršiti simulaciju, za sustav koji se sastoji od nekoliko tisuća čestica, u nekakvom razumnom vremenskom roku. Potom razne metode za izračunavanje termodinamičkih veličina (npr. sred_vrijed_temp, sred_vrijed_tlak), statističku obradu rezultata (npr. srednja_vr_nepouzdanost, zaokruzivanje) te između svega ostalog i atributi koji sadrže stanja tih termodinamičkih veličina (npr. uk_en, temp, komp_fak). Druga klasa modula *md* jest klasa md_BD koja naslijeđuje klasu md_BD_integrator i ona ju proširuje za metode koje se bave mjerenjem i statističkom obradom mjerenih veličina (npr. *graf energija podaci, graf temp podaci*) te njihovim grafičkim prikazom (npr. graf_energija, graf_temp) , metode za provjeru da li se sustav nalazi u termodinačkoj ravnoteži (npr. *ravnotezno stanje kin eng*, *ravnotezno stanje gustoca*). Modul md_SI sadrži klasu md_SI koja naslijeđuje sve od klase md_BD te ju proširuje s metodama za prikaz i obradu podataka u SI jedinicama. Modul Virijal sadrži funkcije za numeričku aproksimaciju drugog virijalnog koeficijenta $B_2$ (izraz 3.41) te njegov grafički prikaz, funkcije za numeričko izvrednjavanje Lennard-Jonesovog potencijala (izraz 2.5) te njegov grafički prikaz. Modul MBdist sadrži funkcije za izračunavanje 1D, 2D, 3D Maxwellove raspodjele brzina, funkcije za izračun srednjih, najvjerojatnijih i srednjih kvadratnih brzina svih prethodno navedenih raspodjela te funkcije za njihov grafički prikaz. Za potpun popis svih metoda i atributa korištenih za izradu aplikacije pogledati izvorni kod priložen uz diplomski rad. Sve do sada navedeno ne predstavlja konkretnu formu pojedinih programa za rješavanje termodinamičkih problema, već platformu za njihovu izradu. U daljnjem dijelu ćemo prikazati kako su pomoću te platforme izrađeni programi korišteni u ovom radu.

### 4.1.3 Stvaranje programa

U ovom dijelu ćemo na generičkom predlošku objasniti kako jednostavno kreirati program s kojim možemo izvršiti razna mjerenja i testiranja nad termodinamičkim sustavom. Opća forma programa se sastoji od dijela u kojemu definiramo sustav te od dijelova u kojima mjerimo i provjeravamo da li sustav zadovoljava određene uvjete koje od njega očekujemo da zadovolji. U ovom jednostavnom generičkom primjeru, kriterij se svodi na čekanje da sustav postane termodinamički uravnotežen.

```
1  from md_SI import *
2
3  #za potrebe sigurnog prekida simulacije
4  def signal_prekid_sim(signal, frame):
5      global siguran_prekid_simulacije
6      siguran_prekid_simulacije = True
7  siguran_prekid_simulacije = False
8  signal.signal(signal.SIGINT, signal_prekid_sim)
9
10 ##Pocetni uvjeti
```

```
md = md_SI(vr_cestice="Argon",
           gustoca=(10,"kg/m3"),
           N_x=100,
           N_y=100,
           L_x=None,
           L_y=None,
           dt=(10,"fs") ,
           ini_temp=(700,"K"),
           poc_pol_ces="kvadratno",
           poc_brz_ces="uniform",
           r_Cutoff=(15,"A"),
           koza=(15,"A"),
           izr_lis_susjeda="A",
           usrednjavanje=2500)

md.odabir_jed_prikaz("SI_std")

md.inicijalizacija_sustava(br_celija_gustoca=(10,10))

md.graf_pol()
md.Maxwell_raspodjela_v_x_iz_smj(parametar_dv=100)
md.Maxwell_raspodjela_v_y_iz_smj(parametar_dv=100)

Pocetak=datetime.datetime.now()

#Cekanje termodinamicke ravnoteze(i Mjerenje)
while md.termodin_ravnoteza is False:
    md.korak()
    md.graf_temp_podaci()
    md.graf_komp_fak_podaci()
    md.graf_tlak_podaci()
    md.graf_energija_podaci()
    md.graf_gustoca_podaci()
    md.statistika_pod_prikaz(
    eng="DA",temp="DA", gustoca="DA",tlak="DA",komp_fak="DA")
    md.md_sazetak(osvjezi_svakih_n_kor=1000)

    if md.broj_koraka % 50 ==0:
        md.graf_temp()
        md.graf_komp_fak("+VIRIJAL")
        md.graf_tlak("+VIRIJAL")
        md.graf_tlak_3D("+VIRIJAL")
        md.graf_energija()
        md.graf_gustoca()

    if md.broj_koraka % 1000 == 0:
        stat=md.vrijeme_statistika(Pocetak)
        print("Vrijeme_po_100_kor:",stat[4])
        print("Vrijeme_proteklo:",stat[2])
```

```
60          print("Vrijeme_pocetak:",stat[0])
61          print("Vrijeme_kraj:",stat[1])
62          md.graf_pol()
63          md.Maxwell_raspodjela_v_x_iz_smj(parametar_dv=100)
64          md.Maxwell_raspodjela_v_y_iz_smj(parametar_dv=100)
65          md.Maxwell_raspodjela_v(parametar_dv=100)
66          md.micanje_grafa=False
67
68      md.uk_eng_drift_provjera(
69      provjera_br_kor=2500,tolerancija=0.5,odgoda_br_kor=1)
70
71      t_r = md.ravnotezno_stanje_temp(
72            provjera_br_kor=2500,tolerancija=1,odgoda_br_kor=1)
73      p_r = md.ravnotezno_stanje_tlak(
74            provjera_br_kor=2500,tolerancija=1,odgoda_br_kor=1)
75      ke_r = md.ravnotezno_stanje_kin_eng(
76            provjera_br_kor=2500,tolerancija=1,odgoda_br_kor=1)
77      pe_r = md.ravnotezno_stanje_pot_eng(
78            provjera_br_kor=2500,tolerancija=2,odgoda_br_kor=1)
79      ro_r = md.ravnotezno_stanje_gustoca(
80            provjera_br_kor=2500,tolerancija=5,odgoda_br_kor=1)
81      Max_r = md.ravnotezno_stanje_raspod(
82            provjera_br_kor=2500,tolerancija=1,odgoda_br_kor=None)
83
84  #    md.zeljena_temp(
85  #        T_zeljena=(20,"K"),provjera_br_kor=10,
86  #        tolerancija=1,odgoda_br_kor=None)
87
88      if( t_r is True and
89          p_r is True and
90          ke_r is True and
91          pe_r is True and
92          ro_r is True and
93          Max_r is True): #
94          md.termodin_ravnoteza = True
95      if siguran_prekid_simulacije:
96          print("Simulacija sigurno prekinuta-integritet podataka ocuvan.")
97          break
98
99  md.graf_pol()
100 md.graf_komp_fak("+VIRIJAL")
101 md.graf_tlak("+VIRIJAL")
102 md.graf_tlak_3D("+VIRIJAL")
103 md.Maxwell_raspodjela_v_x_iz_smj(parametar_dv=100)
104 md.Maxwell_raspodjela_v_y_iz_smj(parametar_dv=100)
105 md.Maxwell_raspodjela_v(parametar_dv=md.parametar_dv_opt_prkz())
106
107 print("Simulacija uspjesno izvrsena!")
```

U prvoj liniji programskog koda importiramo sav sadržaj iz modula md_SI u kojem se nalazi sve što nam je potrebno za kreiranje programa. U programu (linija koda 4-8), iako nije nužna za ispravan rad, definirana je funkcija za siguran prekid simulacije. Ona funkcionira na principu da ako želimo izvanredno prekinuti simulaciju (CTRL-C), to napravimo tako da ne oštetimo integritet do tog trenutka obrađenih podataka. To je ostvareno tako da varijablu *siguran_prekid_simulacije* (95 linija koda) stavimo u dio programa gdje smatramo da je najsigurnije mjesto za prekid simulacije te svaki korak provjeravamo da li je podignuta sistemska zastavica za prekid. Zatim slijedi definiranje termodinamičkog sustava te njegovih početnih uvjeta (linije koda 11-24). To učinimo tako da kreiramo instancu klase md_SI zajedno s inicijalnim parametrima sustava. Prvim parametrom, *vr_cestice*, zapravo definiramo parametre $\sigma$, $\epsilon$, $m$. Zadanim parametrima u samoj klasi ("Argon","Ksenon","Dusik") možemo pristupiti s ključnim "stringom" kao u ovom primjeru. Njihovo dodavanje u samu klasu je lako izvedivo. Samo treba dodati atribut u klasu md_SI poput *Argon = [3.504E-10, 1.62502E-21, 6.6335209E-26]* (linija koda 276 u modulu md_SI). Naravno, moguće je i direktno zadavanje parametara $\sigma$, $\epsilon$, $m$ u formi *vr_cestice*=($\sigma$, $\epsilon$, $m$). Preko parametra *gustoca* (linije koda 12) za zadan broj čestica sustava (parametri N_x i N_y) zadajemo gustoću sustava u formatu *gustoca=(x,"kg/m3") ili (y,"kg/m2")*. Drugi način zadavanja gustoće sustava je preko dimenzija sustava, parametri L_x i L_y (linije koda 15-16). Moguće jedinice za vremenski korak $\Delta t$ (linije koda 17) su sekunda (format(x,"s")) i femtosekunda (format(x,"fs")). Zatim parametar kojim definiramo inicijalnu temperaturu sustava (linija koda 18). Mogući izbor za parametar *poc_pol_ces* (linija koda 19) kojim definiramo početni položaj čestica su "kvadratno", "kvadratno_pola", "nasumicno". Mogući izbor za parametar *poc_brz_ces* (linija koda 20) kojim definiramo inicijalnu raspodjelu brzina čestica su "uniform", "Gauss", "Maxwell", "beta". Parametrom *r_cutoff* određujemo maksimalnu udaljenost na kojoj se može nalaziti čestica da bi se našla na listi susjednih čestica. Moguće jedinice i format tog parametra su *r_cutoff*=(x,"m") u metrima i (y,"A") u angstromima. Parametrom *izr_lis_susjeda* (linija koda 23) određujemo koliko često će se izrađivati lista susjeda. Mogući izbor jest konkretan broj kojim zadajemo nakon koliko koraka će se izraditi lista susjeda ili parametar "A" za automatsku izradu liste susjeda. Automatska izrada je izvedena tako da se prate dvije čestice s najvećim pomakom uz uvažavanje pretpostavke da te dvije čestice idu točno jedna prema drugoj te se pomoću toga izračuna maksimalan broj koraka prije nego kompromitiraju listu susjeda. Uz sve navedeno, dinamički (svaki korak) se vrši korekcija te procjene da se odgodi izrada liste za što je moguće kasniji korak. Za optimalnu brzinu izvođenja ostaviti na automatskom načinu rada. Koliko često će se zapravo izrađivati lista susjeda ovisi o parametru *koza* (linija koda 23) kojim određujemo "debljinu sigurne zone" ($r_{\_cutoff}$ + koza - $r_{\_cutoff}$) u kojoj se vrši procjena za izradu liste susjeda navedena u prethodnom parametru u automatskom načinu rada. Za maksimalnu brzinu izvođenja postoji optimalan iznos parametra *koze*, međutim, sasvim općenito ga nije moguće pronaći tako da njegovu vrijednost određu-

jemo na temelju nekoliko probnih izvršavanja simulacije. Posljednjim parametrom *usrednjavanje* određujemo preko koliko posljednjih mjerenja usrednjavamo mjerene veličine. Njegovi mogući parametri su "N" koji predstavlja usrednjavanje preko svih izvršenih koraka tokom simuliranja ili konkretan broj kojim zadajemo koliko posljednjih mjerenja ulazi u izračun srednje vrijednosti mjerene veličine. Taj parametar, *usrednjavanje*, se odnosi na sve veličine koje podržavaju taj parametar i predstavlja njihovu zadanu (default) vrijednost ako nije eksplicitno zadana kao argument pojedine metode za mjerenje (npr. kao parametar metode *graf_temp_podaci* kojom vršimo mjerenje temperature sustava - md.graf_temp_podaci(preko_zad_n)). Iako nije strogo nužno, takvo zadavanje (prilikom instanciranja sustava) svih parametara sustava je preporučljivo iz razloga što ne moramo voditi računa o pretvorbi veličina u bezdimenzionalne veličine s kojima se izvršavaju svi proračuni. Ako ih želimo zadati naknadno ili mijenjati u toku izvođenja programa, moramo se pobrinuti za ispravne pretvorbe između korištenih jedinica. Većina potrebnih konverzija je implementirana u samoj klasi md_SI te se jednostavno izvodi pozivom metode za željenu konverziju jedinica. Primjerice, ako želimo veličinu A zadanu u angstromima pretvoriti u bezdimenzionalnu veličinu, koristit ćemo metodu *SI_duljina_A_BD(A)* ili ako želimo bezdimenzionalnu veličinu, primjerice temperaturu, pretvoriti u dimenzionalnu veličinu, koristit ćemo metodu *BD_SI_temperatura_K(BD)*. Format svih ostalih metoda za pretvorbu jedinica je analogan prethodno navedenim primjerima. Metodom *odabir_jed_prikaz* (linija koda 26) odabiremo skup jedinica u kojima želimo imati pojedine veličine (uz standardnu bezdimenzionalnu formu). Skup tih odabira jedinica je iskorišten prilikom grafičkog prikaza mjerenih veličina. Mogući predefinirani izbori su "SI_std" koje definiraju jedinice za određene veličine koje su uobičajene u tom području fizike (npr. jedinica za duljinu je angstrom, itd) te drugi izbor je "JKU" koji predstavlja set jedinica istih kao i ulaznih jedinica korištenih za definiranje početnih parametara sustava. Metoda *inicijalizacija_sustava()* (linija koda 28) objedinjuje niz poziva inicijalizacijskih metoda kojima se priprema sustav za daljni rad (npr. pocetni_polozaj_cestica(), pocetna_brzina_cestica(), itd.). Njezin jedini parametar jest br_celija_gustoca kojim određujemo na koliko kvadranata je podijeljen sustav za potrebe metode graf_gustoca_podaci(). Iako se sva mjerenja mogu izvršiti bez povratnih informacija u toku mjerenja bilo kakve forme, korisno je imati nekakvu povratnu informaciju o stanju sustava prilikom izvođenja simulacije. To je izvedeno s raznim metodama za grafički prikaz mjerenih veličina. Metoda *graf_pol()* (linija koda 30) je jedna od tih i ona služi za grafički prikaz položaja čestica. *Maxwell_raspodjela_v_x_iz_smj()* (linija koda 31) metoda služi za prikaz raspodjela brzina čestica po iznosu i smjeru po osi x sustava. Njezinim parametrom, parametar_dv, određujemo na koliko će intervala ("stupića" histograma) biti podijeljen ukupan raspon brzine koji "razapinju" čestice sustava. Analogno objašnjenje vrijedi i za metodu *Maxwell_raspodjela_v_y_iz_smj()* (linija koda 32). Kriteriji koje program treba zadovoljiti da bi smatrali da je simulacija ispunila svoj zadatak mogu biti razni. U ovom generičkom primjeru to je uvjet

da sustav izvodi, izračunava dinamiku sustava i vrši mjerenja raznih termodinamičkih veličina, tako dugo dok sustav ne uđe u termodinamičku ravnotežu. U programskom kodu (linija koda 37-97) to je izvedeno pomoću konstantnog provjeravanja uvjeta (while petlja) atributa termodin_ravnoteza koji postane istinit (True) kada sustav uđe u termodinamičku ravnotežu. Da bi atribut termodin_ravnoteza postao istinit moraju biti ispunjeni daljni uvjeti definirani u programskom kodu u linijama 88-94 koji nas vode prema metodama kojim provjeravamo da li se sustav nalazi u termodinamičkoj ravnoteži. Njihova funkcionalnost je već objašnjena na prethodno obrađenim primjerima (Primjer 1 (2.10.5) i Primjer 2 (3.2)). Da bi te metode mogle izvršavati svoju zadaću, prije svega moraju biti dostupni podaci mjerenja željenih veličina od interesa, a to je izvedeno pomoću, primjerice, metode *graf_temp_podaci()* kojom vršimo mjerenja temperature sustava. Svaka od tih metoda (linija koda 39-43) mjeri i statistički obrađuje veličinu. Njihov jedini mogući parametar jest preko_zad_n koji je objašnjen u prethodnom dijelu ovog teksta. Njihove metode za grafički prikaz, graf_tlak(), graf_tlak_3D(), graf_komp_fak() (linija koda 51-53) imaju parametar "+VIRIJAL" koji uz uvjet da je atribut termodin_ravnoteza=True grafički prikazuje njihovu vrijednost, koja je dobivena iz teoretskih izračuna (vidi Primjer 2 (3.2)). Sve do sada navedeno ne bi bilo moguće bez metode *korak()* (linija koda 38), čijim pozivom se izvrši dinamika svih čestica i izračun osnovnih veličina, pomoću kojih izračunavamo i mjerimo sve ostale veličine koje želimo znati o sustavu. Iz toga razloga poziv te metode prethodi svim ostalim pozivima metoda. Redoslijed poziva ostalih metoda je većinom proizvoljan. Naravno, nema smisla, primjerice, prije pozvati metodu za grafički prikaz tlaka (graf_tlak(linija koda 51)) ako prije toga nisu dostupni podaci potrebni za prikaz istih koje stvorimo s pozivom metode graf_tlak_podaci (linija koda 43). Metoda *statistika_pod_prikaz()* se bavi obradom podataka potrebnih za grafički prikaz veličina koje su navedene kao, iz samog naziva jasnih, parametri metode (linija koda 45). Metodom *md_sazetak* (linija koda 46) sažeto prikazujemo sve relevantne vrijednosti mjerenih veličina i početnih postavki sustava. Njezinim parametrom osvjezi_svakih_n_kor, kao što i samo ime govori, osvježavamo sadržaj prikaza. Iz razloga što prikaz takvih informacija na matplotlib grafovima zahtijeva svakakve transformacije i manipulacije sa "stringovima" to radi prilično sporo te (ako se koristi) se preporučuje koristiti s relativno velikim intervalom osvježavanja. Iz istog razloga se grafovi za prikaz mjerenih veličina ne osvježavaju svaki korak jer su relativno "skupi" u smislu potrebnog procesorskog vremena. Jedan od načina kako se to može izvesti dan je u ovom primjeru (linije koda 48-54). Metode za koje je potrebno značajno više procesorskog vremena (poput Maxwell_raspodjela_v() - linija koda 65) stavimo još veći period osvježavanja. Atributom micanje_grafa (linija koda 66) omogućujemo promjenu inicijalnog razmještaja položaja prozora koji sadrže grafove. Metodom uk_eng_drift_provjera() (linija koda 69) provjeravamo da li je ukupna energija sustava očuvana do na parametrom zadanu toleranciju. Ako želimo sustavu zadati temperaturu, to možemo učiniti s metodom *zeljena_temp()* (linija koda 84-86). Način na koji to postiže (reskaliranjem brzina

čestica) ne reproducira sasvim ispravan kanonski ansambl zbog čega treba biti oprezan prilikom njezinog korištenja. Iako ovim kratkim programom nismo upotrijebili sve moguće metode, atribute te opisali sve potencijalne primjene, trebao bi poslužiti kao demonstracija osnovnih mogućnosti te primjer kako iskoristiti prethodno opisane module za stvaranje programa za simulaciju termodinamičkih problema.
U nastavku slijedi popis kreiranih modula i programa za potrebe ovog diplomskog rada:

- md_pocetni_uvjeti.py,
- md.py,
- md_SI.py,
- MBdist.py,
- Virijal.py,
- MB-grafovi.py,
- Primjer_1.py, Usrednjavanje.py,
- Cvrsto_stanje.py, Tekuce_stanje.py, Plinovito_stanje.py,
- Primjer_2.py,
- Maxwellova_raspodjela_1M_cestica.py,
- Genericki_predlozak.py,
- Dirft_energije.

*md_pocetni_uvjeti.py*, *md.py*, *md_SI.py*, *MBdist.py*, *Virijal.py* - moduli čiji je sadržaj opisan u primjeru stvaranja programa (Stvaranje programa,4.1.3)
*MB-grafovi.py* - program s kojim su stvoreni grafovi raspodjele brzina čestica za potrebe istoimenog potpoglavlja (2.6)
*Primjer_1.py* - program korišten za potrebe istoimenog potpoglavlja (2.10.5)
*Usrednjavanje.py* - program korišten za potrebe potprimjera u potpoglavlju Primjer 1(2.10.5)
*Cvrsto_stanje.py*, *Tekuce_stanje.py*, *Plinovito_stanje.py* - programi korišteni za potrebe primjera iz podpoglavlja Kvalitativni prikaz agregatnih stanja (2.10.6)
*Primjer_2.py* - program korišten za potrebe istoimenog potpoglavlja (3.2)
*Maxwellova_raspodjela_1M_cestica.py* - program korišten za potrebe potprimjera u potpoglavlju Primjer 2 (3.2)
*Genericki_predlozak.py* - program korišten za potrebe potpoglavlja Stvaranje programa (4.1.3)
*Dirft_energije.py* - program korišten za potrebe potpoglavlja Drift ukupne energije sustava (4.2.1)

## 4.2 Potencijalne optimizacije

### 4.2.1 Drift ukupne energije sustava

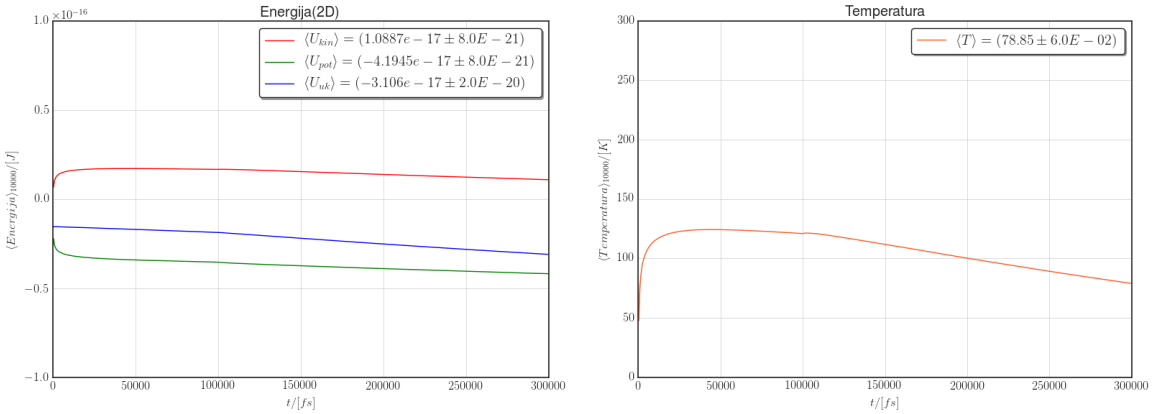
Iako, za probleme kojima smo se mi bavili, ta optimizacija nije nužna svakako je prva na listi mogućih optimizacija. Kada isključimo mogućnosti neočuvanja ukupne energije iz ostalih razloga (npr. premaleni korak $\Delta t$ za integracijski algoritam), još uvijek nam ostane mogućnost narušavanja očuvanja ukupne energije zbog toga što prilikom izračuna ukupne sile na pojedinu česticu ne uzimamo u obzir doprinose svih čestica već samo onih koje se nalaze na udaljenosti manjoj od udaljenosti definirane parametrom $r_{cutoff}$ na kojoj doprinos ukupnoj sili smatramo zanemarivim. Iako u našim prethodno obrađenim primjerima to jest zanemarivo (unutar granica tolerancije), pod raznim okolnostima (npr. velika gustoća,veliki broj čestica, itd) to može značajno narušiti očuvanje ukupne energije sustava. Naravno, uvijek možemo povećati $r_{cutoff}$ da nam je ukupna energija očuvana do na zadanu toleranciju, međutim, time poništavamo ono (optimizaciju brzine izvođenja) zbog čega smo ga i prvobitno uveli. Prilikom povećavanja $r_{cutoff}$ treba imati na umu da zbog periodičkih rubnih uvjeta (2.4), $r_{cutoff}$ + debljina "koze" ne smije premašiti polovicu dužine kraće dimenzije sustava. Na sljedećem primjeru ćemo demonstrirati jednu od situacija u kojima je očuvanje ukupne energije značajno narušeno te kakve su posljedice toga. Početni uvjeti za tu demonstraciju su prikazani u tablici 4.1.

| **Početni uvjeti** | |
|---:|---|
| Broj čestica($N$) | 10000 |
| Vrsta čestica | Argon |
| Masa čestica | $6.634 \cdot 10^{-26}$ kg |
| $\sigma$ | 3.504 Å |
| $\epsilon$ | $1.625 \cdot 10^{-21}$ J |
| Korak($\Delta t$) | 10 fs |
| Gustoća($\rho$) | 1500 kg/$m^3$ |
| Temperatura | 80 $K$ |
| Početna konfiguracija čestica | kvadratno |
| Početna raspodjela brzina | uniform |
| $r_{cutoff}$ | 8.76 Å |
| Koža | 1 Å |
| Usrednjavanje | 10000 |

Tablica 4.1: Drift ukupne energije - početni uvjeti

Ovaj primjer se razlikuje od primjera 2 (3.2) po tome što je 150 puta gušći, inicijalno zadana temperatura mu je $80K$ dok je u primjeru 2 to $700K$. U ovom primjeru, $r_{cutoff}$ smo stavili na $2.5\sigma$ jer u praksi je to najčešće korištena vrijednost. Još smo smanjili vrijednost parametra *koza* za faktor 15 u odnosu na primjer 2 (taj parametar

nema utjecaja na očuvanje ukupne energije). Iako dobiveni rezultati mjerenja (slika 4.1), za korišteni Lennard-Jonesov potencijal s takvim početnim uvjetima (4.1), ne predstavljaju realno upotrebljive podatke, jasno demonstriraju problematiku. Ako pogledamo graf 4.1(a) iz nagiba krivulje $U_{uk}$ je samoevidentno da ukupna energija nije očuvana. Vidimo da za dane početne uvjete (4.1) ukupna energija konstantno "curi" iz sustava. Jedna očita posljedica toga je da sustav ne može postići termodinačku ravnotežu jer mu se, kao što možemo vidjeti na grafu 4.1(b), temperatura konstano mijenja te su nam u tom slučaju praktički sva mjerenja dobivena iz takvog sustava bezvrijedna. Bez obzira što kod tako velikih gustoća Lennard-Jonesov potencijal ne opisuje realno stanje sustava i dalje predstavlja vrijedan model s kojim možemo proučavati kvalitativno ponašanje termodinamičih sustava (npr. fazni prijelazi).




(a) (b)

Slika 4.1: Posljedice neočuvanja ukupne energije

Međutim, da bi to bilo moguće moramo ga tako modificirati da spriječimo, riješimo prethodno prikazan problem. To je riješeno tako da prvobitno definirani Lennard-Jonesov potencijal (2.5) modificiramo tako da nema prekid u točki $U_{LJ}(r_{cutoff})$.

$$U_{LJ-K}(r) = \begin{cases} U_{LJ}(r) - U_{LJ}(r_{cutoff}), & \text{za } r \leq r_{cutoff} \\ 0, & \text{za } r > r_{cutoff} \end{cases} \tag{4.1}$$

Iako tako korigirani potencijal $U_{LJ-K}$ nema prekid u točki $r = r_{cutoff}$, sila ima te ga trebamo dodatno modificirati. To riješimo s dodatnim faktorom tako da je i sila u točki $r = r_{cutoff}$ neprekinuta.

$$F_K(r) = -\frac{dU}{dr} = \begin{cases} F(r) - F(r_{cutoff}), & \text{za } r \leq r_{cutoff} \\ 0, & \text{za } r > r_{cutoff} \end{cases} \tag{4.2}$$

$$U_{LJ-K}(r) = \begin{cases} U_{LJ}(r) - U_{LJ}(r_{cutoff}) - \frac{dU}{dr}\Big|_{r=r_{cutoff}}, & \text{za } r \leq r_{cutoff} \\ 0, & \text{za } r > r_{cutoff} \end{cases} \tag{4.3}$$

Sukladno predloženim modifikacijama potencijala i sile morali bismo modificirati i veličine koje smo pomoću njih i izveli kao primjerice izraz za tlak. Treba voditi računa

da u tom slučaju (korigirani potencijal, itd) parametri ($\sigma$,$\epsilon$) više ne predstavljaju ove do sada korištene kojima smo neposredno verificirali ispravan rad simulacije.

### 4.2.2 Ostale optimizacije i mogućnosti

U nastavku ukratko navodimo potencijalne optimizacije s kojima bi porasla iskoristivost i učinkovitost platforme za izvođenje termodinačkih simulacija.

- proširenje na tri dimenzije
- proširenje s drugim potencijalima
- omogućiti paralelno izvođenje bitnijih dijelova programskog koda
- omogućiti pohranu stanja sustava
- sučelje za izradu i/ili učitavanje konfiguracija čestica (molekula)
- grafičko sučelje (GUI) s mogućnošću upravljanja u realnom vremenu (korisno za edukacijske svrhe)

## *4.3 Korišteni računalni resursi*

Izvođenje bilo kakvih pokusa iz područja molekularne dinamike (MD) imaju vrlo visoke zahtjeve za računalnim resursima. Sve simulacije korištene za potrebe ovog diplomskog rada su izvođene na sljedećoj konfiguraciji računala čije bitne komponente za izvođenje simulacija MD su:

- procesor: Intel Core i7 4790K (4.2 GHz)
- RAM: DDR3 Corsair 8GB, XMP 1.3(2132) PC3-17100 (1066 MHz)

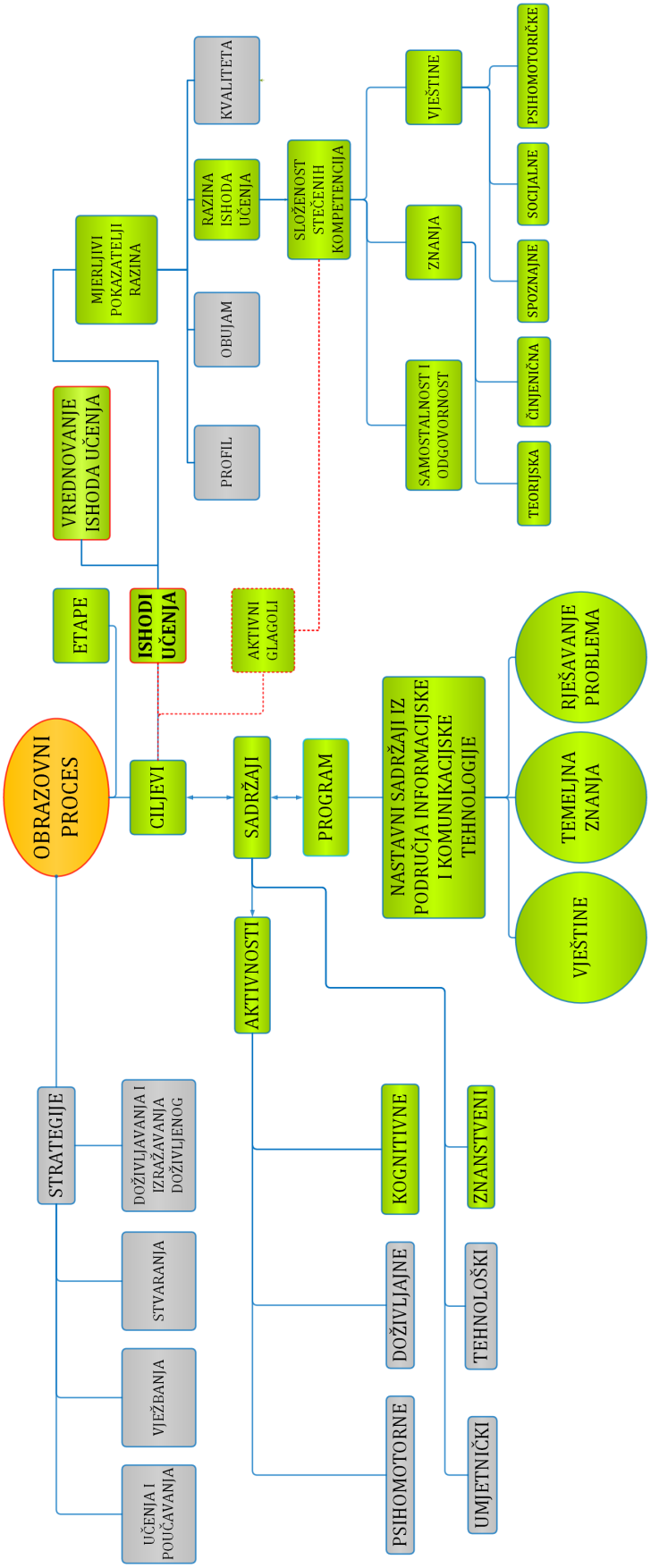
# 5 Metodički dio

Metodika, kao što i samo ime sugerira, se bavi pronalaženjem (pružanjem) odgojno-obrazovnih metoda kako učinkovito prenijeti znanje između ljudi. Iako se sve metodike slažu oko općeg cilja (maksimizacija prijenosa znanja), ono oko čega se ne slažu su metode kako to postići. Iz tog razloga što ne postoji opća teorija metodike, njezine metode su vezane uz određenu struku. Tradicionalne metode prijenosa znanja (kreda-ploča) se najčešće svode na formu u kojoj je predavač u fokusu tog procesa, dok je učenik većinom pasivni sudionik. Takva metoda u praksi pokazuje brojne nedostatke u ispunjavanju gore navedenog cilja. Suvremena rješenja se nude u formi metoda *edukacijskog konstruktivizma*[13] koje, ukratko, fokus s predavača stavljaju na učenika. S učenikom u fokusu, sve metode i načini prijenosa znanja bi trebale biti tako formirane da potiču njegov maksimalan *intelektualan angažman*, sve u cilju optimalnog transfera znanja. Sve do sada rečeno vrijedi općenito za bilo koju metodiku. Metodika fizike nudi metode kako gore navedeno postići u okviru fizike kao znanstvene grane. Iako ima raznih metoda koje potiču intelektualni angažman, mi ćemo se fokusirati na računalne simulacije (modele) kao jedne od metoda za ostvarenje tog cilja. Iz razloga što se fizikalni modeli iskazuju matematičkim formalizmom, to u principu omogućuje njihovu računalnu reprezentaciju u formi simulacije. S razvojem visoke tehnologije to sve više postaje i praksa. Pri tome se ne misli na simulacije izvođene na (po današnjim standardima) superračunalima, nego na malo bržim osobnim računalima. To otvara sasvim novi način za učenje fizike. U okviru ovog diplomskog rada izrađena je aplikacija koja simulira međudjelovanje među česticama iz kojih u konačnici možemo objasniti mnoge termodinamičke veličine te nam može poslužiti kao alat za postizanje gore navedenih ciljeva. Iako su svi fizikalni koncepti apstraktni, koncepti termodinamike i statističke fizike, zbog svoje općenitosti, podižu razinu apstrakcije još više. U tom smislu, imati rješenje koje pokazuje kako te veličine nastaju iz međudjelovanja među sastavnim dijelovima materije (što je sve samo ne očito) može biti od neprocjenjive vrijednosti. Iz razloga što se rezultati mjerenja u simulaciji, u granicama primjene, mogu provjeriti s mjerenjima dobivenih klasičnim eksperimentom predstavljaju neupitno uvjerljiv način učenja što često nedostaje kod ostalih metoda. Takva metoda može poslužiti kao mehanizam za detekciju pretkoncepcija te olakšati njihovo ispravljanje.

## *5.1 Struktura obrazovnog procesa*

"Odgojno obrazovni proces je planska i cilju usmjerena djelatnost koja polazeći od određenih društvenih i individualnih pretpostavki teži ostvarenju društveno i individualno relevantnih postignuća" [22]. Na slici 5.2 je prikazana shema tog procesa.

[13] Idejni začetnik modernog konstruktivizma je švicarski psiholog Jean Piaget (1896.-1980.)




Slika 5.2: Struktura obrazovnog procesa

Na slici 5.2 (osjenčano zeleno) su sve segmenti koje bi ispravno izrađena priprema za nastavni sat iz prirodoslovnih i informatičkih predmeta neke određene nastavne jedinice trebala uzeti u obzir. Ciljevi obrazovanja iskazuju namjeru [22]. Oni su polazište svakog obrazovnog procesa [22]. Obrazovni proces počinje određenom namjerom, a završava provjerom koliko je ta namjera ostvarena, tj. ishodima učenja [22]. Ishodi učenja su znanja i vještine, te pripadajuća samostalnost i odgovornost koje je osoba stekla učenjem i dokazuje nakon postupka učenja [22]. Ishodi učenja predstavljaju jasno definiranu tvrdnju o tome što bi učenik trebao "znati" po završetku procesa učenja. S time da se glagol "znati" iskazuje u formi aktivnih glagola koje možemo direktnije provjeriti (npr. identificirati, predvidjeti, prikazati, itd.). U tom smislu možemo vidjeti da ishodi učenja predstavljaju vrlo bitnu ulogu u obrazovnom procesu jer njihovim dovoljno preciznim definiranjem imamo mogućnost njihovog vrednovanja što predstavlja vrijednu informaciju s kojom se može verificirati i unaprijediti proces obrazovanja. Razina ishoda učenja označava složenost i doseg stečenih kompetencija, a opisuje se skupom mjerljivih pokazatelja [22]. Mjerljivi pokazatelji razina su opisi ishoda učenja određene razine [22]. Minimalan broj (mjerljivih) osnovnih svojstava ishoda učenja su razina, obujam, profil, kvaliteta. Razina ishoda učenja označava složenost stečenih kompetencija, neovisno o drugim osnovnim svojstvima. Mjerljive pokazatelje razina ishoda učenja prikazujemo složenošću sljedećih kompetencija: znanja, vještina te pripadajuće samostalnosti i odgovornosti. Ono na čemu se ostvaruju zadani ishodi učenja su sadržaji. Sadržaji i aktivnosti su integralni dio programa koji se donosi kao državni dokument ili dokument na razini škole [22]. Nastavni sadržaji iz područja informacijske i komunikacijske tehnologije i računarstva moraju učenicima omogućiti: stjecanje vještina uporabe današnjih računala i programske potpore, upoznavanje sa osnovnim načelima i idejama na kojima su sazdana računala odnosno informacijska i komunikacijska tehnologija (temeljna znanja), razvijanje sposobnosti za primjene informacijske i komunikacijske tehnologije u različitim područjima (rješavanje problema) [22]. Iako to predstavlja nesumnjivo zahtjevan način poučavanja koji sigurno maksimizira intelektualni angažman obje strane, ponekad, kada je cilj dovoljno plemenit, sredstva opravdavaju ciljeve.

# 6 Zaključak

U ovom diplomskom radu izrađena je programska platforma za izvođenje simulacija dvodimenzionalne molekularne dinamike čije čestice međudjeluju u Lennard - Jonesovom potencijalu. Vidimo da je računalna tehnologija dovoljno razvijena i pristupačna da izvođenje realno vrijednih simulacija u razumnom vremenskom roku nije više ekskluzivna domena superračunala. Iz računalno izvedenog eksperimenta možemo vidjeti da Lennard-Jonesov potencijal u granicama primjene ima realnu upotrebljivost. Program za simuliranje molekularne dinamike na računalu je napisan u programskom jeziku Python korištenjem objektno orijentiranih principa. Ovim radom demonstrirano je da programski jezik Python predstavlja dobar izbor za programsko ostvarenje i kompleksnijih problema te svojom jednostavnošću i fleksibilnošću predstavlja jasnu razliku između *moguće je* i *vjerojatno je moguće*. Eventualni nedostatak brzine izvođenja kompenzira s bogatim fondom raznih biblioteka s kojima su rješenja kompleksnih problema često nadohvat par programskih linija koda. Primjerice, u ovom diplomskom radu s bibliotekom matplotlib, koje prije svega služi za grafički prikaz podataka, u svega nekoliko linija programskog koda omogućeno je da na jednostavan način vizualiziramo podatke izračunatih veličina i još k tome u realnom vremenu, što je često kompliciran pothvat. Iako se rješenja izvedena u Pythonu ne mogu mjeriti, barem što se brzine izvođenja tiče, s profesionalnim rješenjima (kao što smo i demonstrirali (LAMMPS)), međutim, možda njegova daleko veća prednost leži u tome što tako kompleksan koncept, fizikalni model, dovodi u domenu edukacije gdje predstavlja realno upotrebljiv i nezamjenjiv alat za njegovo shvaćanje i učinkovito učenje. U tom smislu bi se, za kraj, potaknut osobnim iskustvom usudio preformulirati poslovicu koja kruži u edukacijskim krugovima; *"ako nešto želiš naučiti, pokušaj to objasniti nekom drugom"* u *"ako nešto želiš zaista naučiti, pokušaj to objasniti računalu"*.

# Dodaci

## Dodatak A Izvod Verletovog algoritma

Imamo Newtonove jednadžbe gibanja napisane u formi A.1, A.2 gdje je $a(t) \equiv (v(t), x(t), t)$.

$$\frac{dx}{dt} = v(t) \tag{A.1}$$

$$\frac{dv}{dt} = a(t) \tag{A.2}$$

Cilj nam je odrediti $x_{n+1}$ i $v_{n+1}$ u trenutku $t_{n+1} = t_n + \Delta t$. Ako izraze $v_{n+1} = v(t_n + \Delta t)$, $x_{n+1} = x(t_n + \Delta t)$ i $x_{n-1} = x(t_n - \Delta t)$ redom razvijemo u Taylorov red dobit ćemo sljedeće:

$$v_{n+1} = v_n + a_n \Delta t + O((\Delta t)^2) \tag{A.3}$$

$$x_{n+1} = x_n + v_n \Delta t + \frac{1}{2} a_n (\Delta t)^2 + O((\Delta t)^3) \tag{A.4}$$

$$x_{n-1} = x_n - v_n \Delta t + \frac{1}{2} a_n (\Delta t)^2 + O((\Delta t)^3) \tag{A.5}$$

Ako zbrojimo i oduzmemo izraze A.4, A.5 dobijemo:

$$x_{n+1} + x_{n-1} = 2x_n + a_n (\Delta t)^2 \tag{A.6}$$

$$x_{n+1} - x_{n-1} = 2v_n \Delta t \tag{A.7}$$

Uvrstimo izraz A.7 po $x_{n-1}$ u izraz A.6 te riješimo po $x_{n+1}$ i dobijemo sljedeće:

$$x_{n+1} = x_n + v_n \Delta t + \frac{1}{2} a_n (\Delta t)^2 \tag{A.8}$$

Tada pomoću izraza A.7 dobijemo:

$$v_{n+1} = \frac{x_{n+2} - x_n}{2\Delta t} \tag{A.9}$$

i iskoristimo izraz A.4 da dobijemo izraz za $x_{n+2}$:

$$x_{n+2} = 2x_{n+1} - x_n + a_{n+1} (\Delta t)^2 \tag{A.10}$$

Uvrstimo izraz A.10 u A.9 i dobijemo:

$$v_{n+1} = \frac{x_{n+1} - x_n}{\Delta t} + \frac{1}{2} a_{n+1} \Delta t \tag{A.11}$$

I na kraju uvrstimo izraz A.4 u izraz A.11 da se riješimo $x_{n+1} - x_n$ i nakon sređivanja dobijemo:

$$v_{n+1} = v_n + \frac{1}{2}(a_{n+1} + a_n)\Delta t \tag{A.12}$$

# Dodatak B Izvod virijalnog teorema

Imamo sustav točkastih masa s vektorom položaja $\vec{r}_i$ na koje djeluje sila $\vec{F}_i$. Fundamentalna jednadžba gibanja je dana izrazom B.13.

$$\dot{\vec{p}}_i = \vec{F}_i \tag{B.13}$$

Uvodimo pomoćnu veličinu $G = \sum_i \vec{p}_i \cdot \vec{r}_i$. Taj izraz deriviramo po vremenu i dobijemo:

$$\frac{dG}{dt} = \sum_i \dot{\vec{r}}_i \cdot \vec{p}_i + \sum_i \dot{\vec{p}}_i \cdot \vec{r}_i \tag{B.14}$$

Prvi član izraza B.14 transformiramo u:

$$\sum_i \dot{\vec{r}}_i \cdot \vec{p}_i = \sum_i m_i \dot{\vec{r}}_i \cdot \dot{\vec{r}}_i = \sum_i m_i v_i^2 = 2K \tag{B.15}$$

a drugi član je:

$$\sum_i \dot{\vec{p}}_i \cdot \vec{r}_i = \sum_i \vec{F}_i \cdot \vec{r}_i \tag{B.16}$$

Tada se izraz B.14 reducira na:

$$\frac{d}{dt} \sum_i \vec{p}_i \cdot \vec{r}_i = 2K + \sum_i \vec{F}_i \cdot \vec{r}_i \tag{B.17}$$

Zatim izraz B.17 usrednjimo preko vremenskog intervala $\tau$. To napravimo tako da obje strane izraza B.17 integriramo po $t$ od 0 do $\tau$ i podjelimo s $\tau$ i dobijemo:

$$\frac{1}{\tau} \int_0^\tau \frac{dG}{dt} dt \equiv \overline{\frac{dG}{dt}} = \overline{2K} + \overline{\sum_i \vec{F}_i \cdot \vec{r}_i} \tag{B.18}$$

ili:

$$\overline{2K} + \overline{\sum_i \vec{F}_i \cdot \vec{r}_i} = \frac{1}{\tau}[G(\tau) - G(0)] \tag{B.19}$$

Za vezani sustav, brzine i položaji su konačne veličine, postoji gornja granica veličine $G$ te sukladno tome uvijek možemo odabrati dovoljno dugi vremeski interval $\tau$ da nam je desna strana izraza B.18 po volji malena. U tom slučaju srednja kinetička energija sustava čestica je:

$$\overline{K} = -\frac{1}{2} \overline{\sum_i \vec{F}_i \cdot \vec{r}_i} \tag{B.20}$$

# Literatura


[1] Gould, H.; Tobochnik, J., Statistical and Thermal Physics, Princeton University Press, 2010.

[2] Gould, H.; Tobochnik, J.; Christian, W., Introduction to Computer Simulation Methods, Addison-Wesley, 2006.

[3] Rapaport, D.C., The Art of Molecular Dynamics Simulation, 2nd edition, Cambridge University Press, 1995.

[4] Rief, F., Fundamentals of Statistical and Thermal Physics. McGraw-Hill, 1965.

[5] Reichl, L. E., A Modern Course in Statistical Physics, 2nd edition.John Wiley & Sons, 1998.

[6] Goldstein, H.; Poole, C.; Safko, J., Classical Mechanics, 3rd edition, Addison-Wesley, 2000.

[7] Rudolf Krsnik, Suvremene ideje u metodici nastave fizike, Zagreb, Školska knjiga, 2008.

[8] LAMMPS Documentation, <http://lammps.sandia.gov/doc/Manual.html>, 15.8.2016.

[9] Nicholas Metropolis; Arianna W. Rosenbluth; Marshall N. Rosenbluth; Augusta H. Teller; Equation of State Calculations by Fast Computing Machines, <http://bayes.wustl.edu/Manual/EquationOfState.pdf>, 15.8.2016.

[10] B. J. Alder; T. E. Wainwright; Phase Transition for a Hard Sphere System, <http://www.pbx-brasil.com/outrasDisciplinas/DinMol/Notas/IIarea/aula202/artigos/JChemPhys_27_1208.pdf>, 15.8.2016.

[11] A. Rahman; Correlations in the Motions of Atoms in Liquid Argon, <http://journals.aps.org/pr/abstract/10.1103/PhysRev.136.A405>, 15.8.2016.

[12] A. Rahman; Molecular Dynamics Study of Liquid Water, http://171.65.102.199/class/public/readings/Molecular_Simulation_I_Lecture4/Rahman_Stillinger_JCP_71_Water_Dynamics.pdf>, 15.8.2016.

[13] Motohisa Oobatake; Tatsuo Ooi; Determination of Energy Parameters in Lennard-Jones Potentials from Second Virial Coefficients, <http://ptp.oxfordjournals.org/content/48/6/2132.full.pdf>, 15.8.2016.

[14] Python Documentation, <https://www.python.org/, 15.11.2016.

[15] Critical constants and second virial coefficients of gases, <http://www.kayelaby.npl.co.uk/chemistry/3_5/3_5.html>, 6.02.2017.

[16] Anaconda Documentation, <https://docs.continuum.io/>, 15.11.2016.

[17] Demokrit , Leukipov, <https://hr.wikipedia.org/wiki/Demokrit>, 10.1.2017.

[18] Isaac Newton, <https://hr.wikipedia.org/wiki/Isaac_Newton>, 10.1.2017.

[19] Albert Einstein, <https://hr.wikipedia.org/wiki/Albert_Einstein>, 10.1.2017.

[20] Richard Feynman - citat, <http://polymer.bu.edu/vmdl/>, 10.1.2017.

[21] Runge Kutta metoda, <https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods>, 13.2.2017.

[22] Metodika nastave informatike - nastavni materijali, <http://www.phy.pmf.unizg.hr/~gorjana/nastava/Informatika/nastavni%20materijali/index.htm>, 13.2.2017.