

**МАТЕМАТИЧКА ГИМНАЗИЈА**

**У БЕОГРАДУ**

**МАТУРСКИ РАД**

**из физике**

**Монте Карло метода и њена примена у физици**

**ментор: Иван Станић**

**ученик : Никола Ристић 4е**

**Београд, јун, 2016**

# Садржај

<b>1</b>	<b>Увод</b>	<b>1</b>
<b>2</b>	<b>Псеудослучајни бројеви</b>	<b>2</b>
2.1	Историја псеудослучајних бројева . . . . .	2
<b>3</b>	<b>Генератори псеудослучајних бројева</b>	<b>3</b>
3.1	Генератори псеудослучајних бројева са једним чуваним парметром . . . . .	3
3.2	Middle-square и други алгоритми средњих цифара . . . . .	4
3.3	Линеарни конгруентни генератор псеудослучајних бројева (ЛКГ) . . . . .	4
<b>4</b>	<b>Расподеле непрекидних случајних промељивих</b>	<b>6</b>
4.1	Униформна расподела и метода инверзије функције расподеле . . . . .	6
4.2	Централна гранична теорема и нормална дистрибуција . . . . .	7
4.2.1	Процена грешке Монте Карло методе . . . . .	8
<b>5</b>	<b>Монте Карло метода код Марковљевих ланца</b>	<b>9</b>
5.1	Случајан ход . . . . .	9
5.2	Матрично приказивање ланца . . . . .	10
5.3	Инваријантна расподела . . . . .	11
5.4	Метрополис алгоритам . . . . .	11
<b>6</b>	<b>Примена у физици</b>	<b>13</b>
6.1	Симулација дводимензионих система . . . . .	13
6.1.1	Честице у кутији . . . . .	13
6.1.2	Изингов модел . . . . .	16
6.2	Симулација тродимензионих система . . . . .	18
6.2.1	Расподела електрона на сфери . . . . .	19
6.2.2	Чврсте сфере у кутији . . . . .	20
<b>7</b>	<b>Закључак</b>	<b>22</b>
	<b>Литература</b>	<b>22</b>

# 1

## Увод

Монте Карло метода је име за читаву класу математичких метода које су научници почели да користе 40-тих година прошлог века у Лос Алмосу приликом израде нуклеарног оружја. Мада је на почетку изгледало као да играње игара на срећу и коришћење случајних бројева неће имати никакву примену у решавању стварних проблема, можемо лако наћи примере где можемо путем игре решити неки конкретан проблем. Савршен пример би био прва примена ове методе за решавање неког математичког проблема: Буфорова игла. Овај експеримент извео је Буфон још 1777, много пре него што су ове методе добиле своје име. У његовом експерименту бацана је игла дужине  $L$  на равну на којој се налазе паралелне линије међусобно удаљене за  $d$  ( $d < L$ ). Он је показао да је шанса да игла пресече једну од тих линија тачно:  $P = \frac{2L}{d\pi}$ . Неколико година касније Лаплас је предложио да резултат овог експеримента (шанса  $P$ ) може се користити за одређивање броја  $\pi$ .

Ипак овакви проблеми су боље решени другим математичким методама. Данас се Монте Карло метод користи за рачунање вишеструких интеграла, за симулацију стохастичких процеса. Наравно ова врста симулације разликује се од класичне симулације која користи законе система да директно дође до одговора. За разлику од такве врсте симулације Монте Карло користи пробалистичке методе да се реше непробалистички проблеми (детермистички).

Треба напоменути да је моћ ове методе утолико порасла и расте са развојем компјутера. Мада на почетку ове методе нису осмишљене са компјутером у виду, данас се не могу изводити озбиљни Монте Карло огледи без њега. То је зато што се уместо већ наведеног мануелног извођења експеримената користе компјутери који генеришу случајне бројеве који ће послужити уместо случајних догађаја у реалном систему.

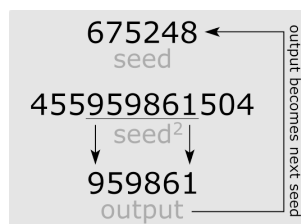
## 2

# Псеудослучајни бројеви

## 2.1 Историја псеудослучајних бројева

У самој основи Монте Карло методе стоје псеудослучајни бројеви. Они се користе да би се симулирали делови посматраних процеса који су насумични. Њихово ефикасно налажење представља један од главних проблема код симулације тих процеса. Ови бројеви су се прво ручно изводили и симулација се изводила механичким калкулаторима које су користили радници. То се радило на самом почетку примене Монте Карло методе још 1943. у Лос Алмосу за решавање проблема неутронског флукса. Ипак овакве симулације би захтевале недеље ако не и месеце тешког мануелног рада да би се извеле. То је навело једног од оснивача ове методе Џон фон Нојмана да раднике замени дигиталним компјутерима, у чији развој је и сам био умешан. Ту се наравно појавио проблем који смо предходно поменули, требало је некако генерисати случајне бројеве путем компјутера. Зато је фон Нојман изумео први генератор псеудослучајних бројева, такозвани *middle-square* алгоритам.

Посматрајући овај алгоритам увиђамо да бројеви који су добијени нису стварно случајни већ зависе од улаза, семена алгоритма. Тако су ови бројеви и добили свој назив, због чињенице да су за једно дато семе унапред одређени бројеви и њихов редослед. Зато се овај проблем лако решава тако што се узима семе које знамо да је насумично, тренутно време на пример. На тај начин коришћењем једног случајног броја добијају се хиљаде псеудослучајних бројева за коришћење.



Слика 2.1: Middle-square алгоритам, један пролаз , семе 675248

## 3

# Генератори псеудослучајних бројева

Као што смо већ поменули генератори псеудослучајних бројева нису ништа више од компјутерских алгоритама. Зато као и сваки компјутерски алгоритам они дају детермистичко решење. Зато они никако не могу бити случајни бројеви, али ипак могу да играју улогу случајних бројева у нашим симулацијама. Зато сваки генератор мора да задовољи дате услове:

1. Да буде брз. За добру симулацију потребни су милиони псеудослучајних бројева.
2. Да буде поновљив, да бисмо могли да тражимо грешке.
3. Да се може анализирати, да би били сигурни да добро дистрибуира бројеве.
4. Да има дугачак период. Сваки генератор се на крају понавља и нама је потребно да то буде што мање тј. период да буде што већи.
5. Да се понаша као случајан број за процес који симулира, што би била најважнија ставка.

Колико је важно пратити ове услове можемо видети на примеру Кнутовог првог алгоритама за генерисање псеудослучајних бројева. Његов алгоритам је био врло компликован и самим тим није био анализиран. Када га је испробао увидео је да брзо конвергира на једну вредност која онда сама себе генерише.

### 3.1 Генератори псеудослучајних бројева са једном чуваном променљивом

Сваки алгоритам за генерисање псеудослучајних бројева се може записати у облику: резултат =  $f$ (променљиве, параметри)

Параметри су задати унапред док променљиве могу бити генерисане у самом алгоритму или дате екстерно, на пример тренутно време или температура или резултат неког другог алгоритама или процеса.

Променљиве се мењају из итерације у итерацију па се често узимају они који алгоритам

сам генерише. У том случају само почетни параметар се уноси споља и он се зове *семе алгоритма*. Ово се ради због услова 2, да би алгоритам био поновљив, јер у случају да семе узима из спољашности за сваку итерацију ми не бисмо могли да поново генеришемо исте псеудослучајне бројеве. Овако је само потребно прво семе узети из спољашности и њега запамтити.

### 3.2 Middle-square и други алгоритми средњих цифара

Ови алгоритми се базирају на идеји да после неколико цифара резултата функција као што су  $\sin(x)$  или  $\log(x)$ , остатак цифара се понашају као случајни бројеви. Ипак већ на почетку се увиђа мана оваквих алгоритама, превише су спори. То следи из познате чињенице да рачунање ових функција на велики број децимала захтева пуно времена.

Алгоритам за генерисање 10 псеудослучајних бројева алгоритмом средњих цифара

**begin**

$x = 0.3$

Семе алгоритма

**for**  $i := 1$  **to** 10 **step** 1 **do**

$x = 10^5 * \sin(x)$

Издавају се првих 5 цифара ван децималног зареза

$w = \text{floor}(x)$

Узима се првих 5 цифара

$x = x - w$

Узимају се цифре после пете као случајне

**od**

**end**

Због спорости овог алгоритма фон Нојман је користио свој већ поменути алгоритам квадрирања семена. Ипак тај алгоритам има један велики недостатак, веома мали период. То наравно доводи до тога да се брзо понављају једни те исти бројеви што уништава сву привидну случајност тих бројева. Због тога је било потребно наћи бољи алгоритам.

### 3.3 Линеарни конгруентни генератор псеудослучајних бројева (ЛКГ)

Линеарни конгруентни генератор је један од најпопуларнијих, најкоришћењих и најједноставнијих генератора псеудослучајних бројева. Он користи просту рекурзију да би из претходног резултата добио следећи ( и наравно из семена добио први псеудослучајан број).

**Дефиниција 3.3.1.** Општа формула за линеарни конгруентни генератор је :

$$f(x) = (ax + c) \bmod m$$

где су  $a$ ,  $c$  и  $m$  параметри а  $x$  промељива.

Добијање низова псеудослучајних бројева овим алгоритмом ради се путем једноставне рекурзије :  $x_{n+1} = f(x_n)$

Наравно ми желимо да добијемо највећи период овим алгоритмом, па се поставља питање какви треба да буду параметри  $a$ ,  $c$  и  $m$  да бисмо то постигли. На то питање нам одговара следећа теорема:

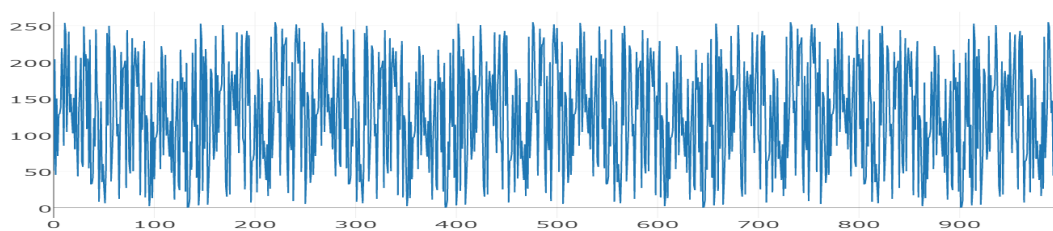
**Теорема 3.3.1.** Линеарни конгруентни генератор псеудослучајних бројева

$$f(x) = (ax + c) \bmod m$$

има период  $m$  ако и само ако

1. највећи заједнички делилац за  $c$  и  $a$  је 1
2.  $a - 1$  је дељив са  $p$  за сваки прости број  $p$  који дели  $m$  и на крају
3.  $a - 1$  је дељив са 4 ако је  $m$  дељив са 4.

**Пример 1.** Узмимо да је  $m = 2^{32}$ , тада вредности параметра  $c = 1$  и  $a = 4l + 1$  дају период  $m$  за сваки цео број  $l > 0$ .



**Слика 3.1:** Линеарни конгруентни генератор псеудослучајних бројева за параметрима  $m=256$ ,  $c=1$ ,  $a=9$ ,  $x=34$ , 1000 итерација

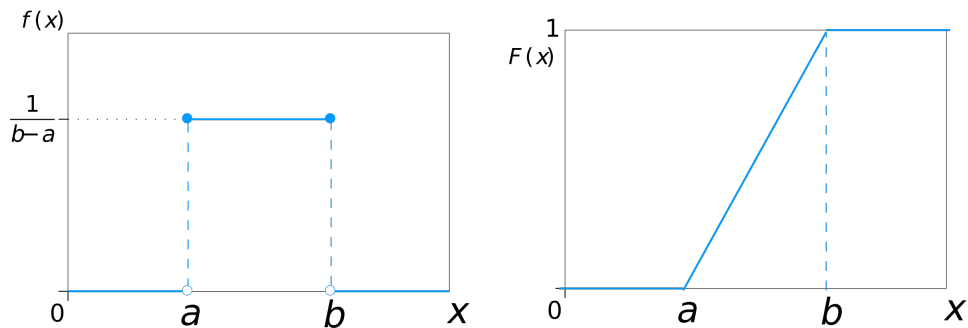
Пошто је ово релативно једноставан алгоритам лако можемо учити дистрибуцију псеудослучајних бројева које ствара. Због саме чињенице да је период једнак пуном  $m$  (предходна теорема) следи да је у свакој дужини од једног периода сваки од ових бројева различит док су крајеви исти. У противном би због саме чињенице да следећа итерација зависи само од претходне следило да је период мањи од  $m$ . Сваки од бројева се онда појављује тачно једном у периоду. Често се резултати деле са  $m$  да би се добили бројеви из сегмента  $[0, 1)$  који се после могу користити за добијање бројева из било ког другог сегмента. Треба споменути још једну предност код ЛКГ. Ако узмемо  $m = 2^{32}$  или  $m = 2^{64}$  због архитектуре рачунара, првенствено чињенице да се подаци чувају у битовима, операција модула се извршава у једном кораку. Једна од највећих мана овог генератора је што се у датом периоду никад не понавља исти број. То указује да бројеви генерисани овим путем нису добри за коришћење као случајни.

# 4

## Расподеле непрекидних случајних промељивих

### 4.1 Униформна расподела и метода инверзије функције расподеле

У Монте Карло методама главну улогу игра нам униформна расподела. Представља једну од најосновнијих али и најмоћнијих расподела у вероватноћи. Означава се са  $U(a, b)$  где су  $a$  и  $b$  границе униформне расподеле. Као што јој назив налаже сваки број између  $a$  и  $b$  је једнако вероватан у овој расподели.



Слика 4.1: Функција расподеле и расподела вероватноће за униформну расподелу од  $a$  до  $b$

Ипак, дешава се да у нашим симулацијама хоћемо да наши случајни бројеви имају другачију дистрибуцију. Уместо да правимо нове алгоритме за генерисање тих бројева можемо да искористимо следеће својство униформне расподеле.

**Теорема 4.1.** (Инвертовање непрекидне функције расподеле) Нека је  $F(x)$  функција расподеле случајне промељиве  $X$  и да је непрекидна и нека је да је  $U = F(X)$ . Тада је  $U$  случајна промељива која има униформну расподелу на сегменту  $[0, 1)$ .



*Доказ:* Треба да докажемо да је  $P(U \leq u) = u$  за  $0 \leq u \leq 1$ . Пошто је  $F$  непрекидна и растућа знамо да постоји  $F^{-1}$ . Ако је  $x = F^{-1}(u)$  онда је:

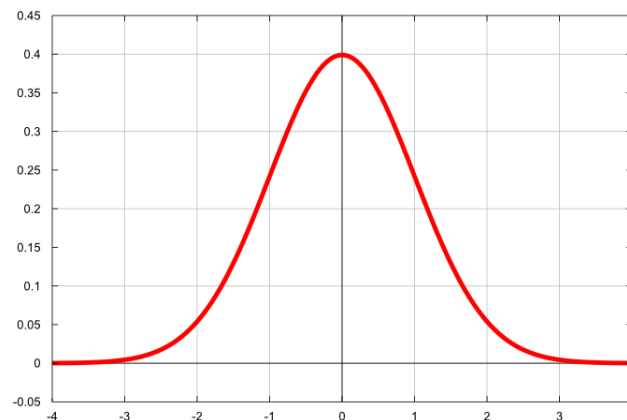
$$P(U \leq u) = P(F(X) \leq u) = P(X \leq F^{-1}(u)) = P(X \leq x) = F(x) = u,$$

што је и тврђено. Самим тим ако је  $U$  узорак из  $U(0, 1)$ , онда је  $X = F^{-1}(U)$  узорак из расподеле са функцијом расподеле  $F$ .

**Пример 2.** Рецимо да хоћемо да наши псеудослучајани бројеви имају експоненцијалну расподелу са параметром  $\alpha$ . Функција расподеле је онда  $F(x) = 1 - e^{-\alpha x}$  и коришћењем теореме 4.1 добијамо  $X$  решавањем једначине  $U = F(X)$  где је  $U$  псеудослучајан број узет из униформне расподеле. Решење ове једначине је  $X = \frac{-1}{\alpha} \ln(1 - U)$ , и њиме добијамо псеудослучајане бројеве  $X$  са експоненцијалном расподелом.

## 4.2 Централна гранична теорема(ц.г.т) и нормална дистрибуција

Током извођења симулација користећи велики број псеудослучајних бројева можемо приметити да резултати полаку 'конвергирају' ка одређеној расподели. Ово је наравно није ништа више од очекиване последице централне граничне теореме. Тако да током израде хистограма за наше симулације можемо приметити да добијају облик симетричног зфона које тежи ка 0 кад  $x \rightarrow + - \infty$ . Такво исто зфоно има и график функције  $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ .



**Слика 4.2:** Функција расподеле и расподела вероватноће за униформну расподелу од  $a$  до  $b$

Доказано је да вредност интеграла:  $\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = 1$ , што је услов да функција буде густина вероватноће неке расподеле. Та расподела се зове нормална расподела и означава се са  $N \sim (\mu, \sigma^2)$  где је  $\sigma$  стандардна девијација а  $\mu$  очекивана вредност расподеле. У нашем случају  $\mu$  је 0 а  $\sigma$  1, а за друге вредности параметра користи се трансформација  $X * \sigma + \mu \sim N(\mu, \sigma^2)$  за  $X \sim N(0, 1)$ .

Колико је ова дистрибуција важна види се после навођења централне граничне теореме.

**Теорема 4.2.1.** (Централна гранична теорема) Нека су  $X_1, X_2, \dots, X_n$  независни бројеви узети из расподеле са очекивањем  $\mu$  и коначном дисперзијом  $\sigma^2$ . У симулацијама то су псеудослучајни бројеви које смо генерисали да прате ту расподелу. Тада

$$Y = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{(n\sigma^2)}}$$

где је  $Y$  случајна промељива добијена горе наведеном формулом има нормалну дистрибуцију са очекивањем 0 и дисперзијом 1 када  $n \rightarrow \infty$ .

Мада на први поглед ова теорема изгледа само да нам гарантује како се понаша збир наших псеудослучајних бројева, треба се сетити да и резултат функције бројева из неке расподеле даје неку другу расподелу, самим тим и збир тј. просек наших симулација често узима нормалну расподелу за довољно велик број симулација. Ипак поставља се питање како добити нормалну расподелу из оне које нам је на располагању, униформне. Не може се користити теорема 4.1 пошто се не може аналитички решити неодређени интеграл од  $e^{-x^2/2}$ . За то се користи Централну граничну теорему.

**Пример 3.** Узмемо  $n$  псеудослучајних бројева из расподеле  $U(0, 1), U_1, U_2, \dots, U_n$  и онда израчунамо  $X = \frac{\sum_{i=1}^n U_i - n/2}{\sqrt{n/12}}$  (Пошто униформна расподела  $U(0,1)$  има очекивање  $1/2$  и дисперзију  $1/12$ ). По ц.г.т  $X$ -еви добијени на овај начин имаће приближно нормалну расподелу са очекивањем 0 и дисперзијом 1. Често је довољно добра апроксимација кад узмемо  $n = 12$  и тиме још избегавамо дељење са кореном.

#### 4.2.1 Процена грешке Монте Карло методе

Када Монте Карло симулацијом тражимо неку скаларну вредност, било то време чекања неког догађаја или неутронски флукс кроз одређену дебљину препреке битно је знати прецизност добијеног резултат. Нека су  $X_1, X_2, \dots, X_n$   $n$  процена дате вредности добијене симулацијом. Ако су те вредности независне и имају једнаку расподелу са очекивањем  $\theta$ , нашим жељеним резултатом, тада по ц.г.т добијамо да просек тих симулација приближно прати нормалну дистрибуцију са очекивањем  $\theta$  и дисперзијом  $\sigma^2_X/n$  где је  $\sigma$  нама непозната стандардна девијација резултата симулације.

**Пример 4.** Хоћемо на пример да будемо 95% сигурни да је наша процена резултата симулације тачна. Узмемо да је  $Y = \frac{\bar{X} - \theta}{\sqrt{\sigma^2_X/n}}$  и то приближно прати расподелу  $N(0, 1)$ . Посматрајући ту расподелу можемо приметити да се број налази са шансом 0.954% у околини од две стандардне девијације од очекивања:

$$Pr(-2 < Y < 2) = 0.954$$

и заменом  $Y$  добијамо границе нашег резултата

$$\bar{X} - 2\sqrt{\sigma^2_X/n} < \theta < \bar{X} + 2\sqrt{\sigma^2_X/n}$$

Коришћењем овог резултата налазе се такозвани интервали поузданости. Ипак у самом резултату видимо недостатак овог поступка, у већини случајева дисперзија резултата је непозната. Зато се та дисперзија већином процењује на основу резултата.

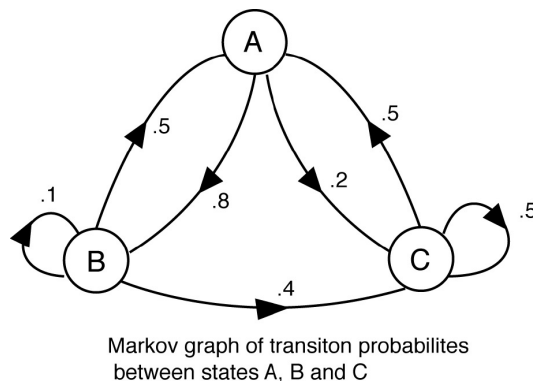
## 5

# Монте Карло метода код Марковљевих ланца

Марковљеви ланци представљају гледање случајних догађаја не из перспективе појединачних догађаја него из перспективе секвенце случајних догађаја. Ако представимо са  $\Omega$  скуп свих могућих исхода, у једном случајном догађају можемо да остваримо само мали подскуп тих исхода. Ако узмемо да  $X_k$  означава стање после  $k$ -те итерације онда је то стање добијено из стања  $X_{k-1}$  неким случајним догађајем. Све могућности прелаза из стања  $X_k$  у следеће стање називамо околином  $X_k$ . Овакаво постављање симулација има бројне употребе у физици. На пример промене у термодинамичким системима, као и Марковљев ланац, се дешавају на случајан начин и зависе само од тренутног стања система.

### 5.1 Случајан ход

Случајан ход представља процес који Марковљеви ланци описују. Сам назив нам говори да се ради о процесу где се стање једног система мења "случајним кораком", неким случајним догађајем који зависи од тренутног стања система. То се најлакше види на графику једног Марковљевог ланца.



Слика 5.1: Марковљев ланац са 3 могућа стања.

Свака линија овог цртежа представља један могућ прелаз стања у једној итерацији, тј. могуће кораке у датом стању. Свако могуће стање до ког систем може доћи из стању  $X$  означавамо са  $N_x$ . На свакој линији на датом цртежу има број који представља вероватноћу тог корака у датом стању које ћемо. Ту вероватноћу означавамо са  $p_{xy}$  где је  $X$  тренутно стање а  $Y$  могуће. Сам ланац почиње у стању  $X_0$  које можемо узети из неке расподеле или просто унапред га задати. Одатле се ланац наставља у складу са вероватноћама прелазу у суседна стања, стања која се налазе у  $N_x$ . Следеће стање се бира на основу расподеле добијене  $p_{xy} : y \in N_x$ . Пошто захтевамо да увек постоји следећа итерација, следећи корак следи да је укупна вероватноћа прелазу једнака 1 за свако  $x \in \Omega$  или формалније:

$$\sum_{y \in N_x} p_{xy} = 1 \text{ за свако } x \in \Omega$$

## 5.2 Матрично приказивање ланца

Марковљевски ланци такође се могу приказати као матрица  $P = (p_{xy})$  где је  $p_{xy}$  већ поменуто вероватноћа преласка из  $x$  у  $y$  у једном кораку. Ова матрица се назива матрица транспорта. Из предходне суме следи да сума сваке врсте матрице  $P$  мора да буде једнака 1. Главни разлог зашто се ланац представљам на овај начин је могућност рачунања вероватноће могућих стања. Узмимо да је  $p_t$  вектор вероватноће случајне промелјиве  $X_t$ . То је вектор чије су компоненте ненегативне и њихов збир је 1. Сваки број представља вероватноћу система да буде у одређеном стању. Тада користећи матрицу транспорта можемо израчунати  $p_{t+1}$  као следећи производ:

$$p_{t+1} = p_t P \quad t=1,2,\dots$$

**Пример 5.** Узмимо на пример матрицу транспорта која има 3 могућа стања  $X, Y, Z$ .

$$P = \begin{matrix} & \begin{matrix} X & Y & Z \end{matrix} \\ \begin{matrix} X \\ Y \\ Z \end{matrix} & \begin{pmatrix} 0.3 & 0.2 & 0.5 \\ 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{pmatrix} \end{matrix}$$

И узмимо  $X_0 = X$ , тј да почињемо од стања  $X$ , онда можемо користећи предходну формулу да израчунамо  $p_1$

$$p_1 = [1 \ 0 \ 0] \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} = [0.3 \ 0.2 \ 0.5]$$

Тако рекурзијом добијамо општу

формулу за  $p_t$

$$p_t = p_0 P^t$$

Ако наставимо да рачунамо вредности тих вектора вероватноће:

$$p_3 = [0.332 \ 0.304 \ 0.364], \quad p_5 = [0.33032 \ 0.3036 \ 0.36608] \quad p_{10} = [0.330357 \ 0.303571 \ 0.366072]$$

приметићемо да брзо конвергирају до такозване инваријантне расподеле. Ово је општа одлика великог броја Марковљевих ланца.

## 5.3 Инваријантна расподела

За уобичајене Марковљеве ланце имамо да вектори вероватноће  $p_t$  теже ка инваријантној расподели  $\pi$ . Назив инваријантна расподела је добила по њеном својству :

$$\pi P = \pi$$

Самим тим чим вектор вероватноће дође у близину инваријанте расподеле он ту и остане.

**Теорема 5.3.1.** (Перрон-Фробенусова теорема) Вектор вероватноће тежи инваријантној дистрибуцији ако за Марковљев ланац важи:

1. Иредуцибилан: мора да постоји усмерена путања од једног стања до сваког другог стања (постоји шанса да из било ког стања дође у било које друго)
2. Апериодичан: не сме да постоји ни један цео број  $d > 1$  тако да број итерација за враћање до првобитног стања мора да буде дељив са  $d$ .

## 5.4 Метрополис алгоритам

Сама идеја за коришћење Марковљевих ланца за симулацију процеса потиче још из 1953. када је Н.Метрополис објавио свој рад "Equations of state calculations by fast computing machines". Ту је разматрао проблем из физике везан за кинетичко кретање атома и молекула. У том проблему се мерила и пратила локација, оријентација и брзина атома и молекула. Ту се први пут помиње Метрополис алгоритам за симулацију. Треба напоменути да се он данас сматра једним од најбољих алгоритама који је имао велики утицај на развој науке и инжењерства. Када се користе Марковљеви ланци за симулацију неког процеса у великом броју случајева се користи овај алгоритам да би се то постигло.

Сам алгоритам је врло једноставан и флексибилан. Први корак у алгоритму је да конструишемо Марковљев ланац који задовољава теорему 5.3.1. Користићемо вероватноћу  $P(X'|X)$  односно вероватноћу да се из  $X$  дође у стање  $X'$ , тј вероватноћу трансорта. Ми сад захтевамо да се систем налази у равнотежи подразумевамо да је обрнут процес једнако вероватан тј:

$$P(X'|X)P(X) = P(X|X')P(X')$$

где под  $P(X)$  подразумевамо као вероватноћу да је систем у том стању, тј  $P(X)$  је расподела вероватноће за наш систем. Овај услов назива се услов детаљног баланса.

И записује се у виду

$$\frac{P(X'|X)}{P(X|X')} = \frac{P(X')}{P(X)}$$

Следећи корак алгоритма је раздвајање вероватноће транспорта у производ две вероватноће.

Први део је вероватноћа за прихватање следећег стања  $A(X'|X)$ , ако је систем у стању  $X$  и покушамо да га померимо у стање  $X'$ , која је вероватноћа да се помери. Други део је расподела вероватноће следећих стања  $g(X'|X)$ . Она представља вероватноћу да се предложи покрет из стања  $X$  у стање  $X'$ .

Можемо изабрати било коју расподелу следећих стања која одговара нашим потребама са једним ограничењем.

$$g(X' \| X) = g(X \| X').$$

Вероватноћа транспорта раздвајам на следећи начин:  $P(X' \| X) = g(X' \| X)A(X' \| X)$  и тиме добијамо једначину за  $A(X' \| X)$  а то је:  $\frac{A(X' \| X)}{A(X \| X')} = \frac{P(X')g(X \| X')}{P(X)g(X' \| X)}$

и чест избор за  $A(X' \| X)$  да би ова једначина била задовољена је такозвани Метрополски избор:  $A(X' \| X) = \min(1, \frac{P(X')}{P(X)})$  где се  $g(X \| X')$  и  $g(X' \| X)$  скрате због симетрије. Из ове формуле следи да Метрополски алгоритам аутоматски прихвата вероватније стање, док мање вероватно стање прихвата са вероватноћом  $A(X' \| X)$ .

Метрополис алгоритам

**begin**

**Initialize:**  $X$  Постави тренутно стање система

**for**  $i := 1$  **to**  $n$  **step** 1 **do**

$Y \in N_X$  Предложи ново стање система

$A = \min(1, P(Y)/P(X))$  Вероватноћа да предлог прође

$U = \text{rand}(0, 1)$  Генериши случајан број за проверу предлога

**if**  $U \leq A$  **then**  $X = Y$  Ако предлог прође промени стање

**fi**

**Save:**  $X$  Запамти стање ради израде хистограма

**od**

**end**

При коришћењу овог алгоритма, као и било ког другог у Монте карло методама, циљ је израчунавање очекиване вредности неког својојства датог система. Коришћењем овог алгоритма добија се очекивана вредност Марковљевог ланца који се симулира. То доказује следећа теорема.

**Теорема 5.4.1.** Узмимо да је  $(X_N)$  иредуцибилан, апериодичан Марковљев ланац са коначним простором исхода  $\Omega$  и стационарном дистрибуцијом  $\pi$ . Нека је  $r : \Omega \rightarrow R$  било која функција. Тада важи:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N r(X_i) = E_{\pi}(r), \text{ где } E_{\pi} \text{ означава очекивање за дату дистрибуцију } \pi.$$

Прећутно смо овде узимали да Метрополис алгоритам стварно симулира Марковљев ланац који има инваријанту расподелу  $\pi$ . Треба показати да он задовољава услове које теорема 5.3.1 постаља за такве ланце. Прво знамо да се не добија периодичан систем пошто ће увек постојати стање  $Y$  у чијој околони ће систем остати због постојања  $(f(Y) \geq f(X) \vee X \in N_Y)$ , за други услов треба да одаберемо расподелу следећих стања  $g(X' \| X)$  такву да омогућава прелаз из једног стања система у било које друго на пример константу за сва стања.

# 6

## Примена у физици

Када смо упознати са Метрополис методом и самим тим и Марковљевим ланцима можемо почети говорити о примени Монте Карло методе у физици. Главна примена базира се на избегавању проблема класичне физике у системима са великим бројем објеката. Такви системи постају готово немогући да се реше аналитички или чак директном симулацијом. Шта више за системе који се налазе у више димензија проблем постаје још комплекснији. Зато у таквим ситуацијама се користи Метрополис алгоритам. Мада овај алгоритам не даје најбољи увид у то како се систем креће, он нам може дати очекивану вредност неког својства тог система, на пример колики ће бити магнетни момент датог система. То гарантује теорема 5.4.1.

Сам алгоритам се своди на кретање објеката кроз  $n$ -димензијони простор на произвољан начин. Потребна нам је расподела вероватноће  $f(R)$  где је  $R$  вишедимензиони вектор који представља наш објекат. И се користи стандардни Метрополис алгоритам: узмемо произвољан вектор из  $f(R1)$  где се  $R1$  узима из скупа вектора на почетку. Следеће стање се бира на основу вероватноће  $g(R2||R1)$  и проверава се да ли је прелаз прихваћен.  $q(R2||R1) = \frac{f(R2)g(R1||R2)}{f(R1)g(R2||R1)}$  и наравно ако је  $q > 1$  одмах се прихвата иначе се прихвата са шансом  $q$ .

### 6.1 Симулација дводимензионих система

#### 6.1.1 Честице у кутији

Да би боље приказали овај начин резоновања узећемо једноставан пример честица у 2Д кутији. Сваку честицу описујемо двама координатама  $(x_i, y_i)$ . Укупна енергија овог система онда је једнака:  $E = \sum_{i=1}^N \frac{p_i^2}{2m_i} + \sum_{i < j < N} E_p(r_i - r_j)$  где први члан суме представља кинетичку енергију система, а други члан суме представља потенцијалну енергију система

За потенцијалну енергију система ми ћемо користити Lennard-Jones потенцијал

$$E_{LJ}(r) = 4 * e((\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6)$$

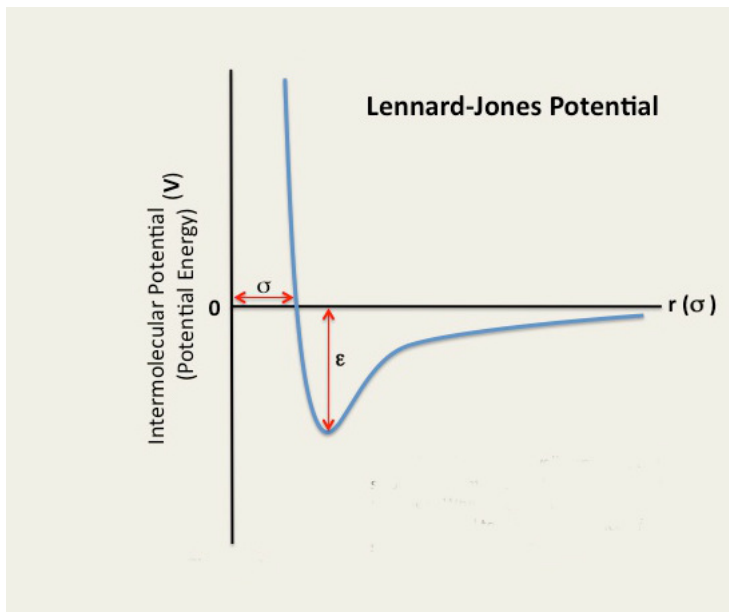
где је  $r$  растојање између честица,  $e$  је дубина потенцијалне јаме а  $\sigma$  растојање на којем је потенцијална енергија између две честице једнака 0.

Расподелу вероватноће честица онда тражимо као функцију позиције и брзине честица.

$$f(v, R) = \exp(-E(v, R)/k_B T) / \int \exp(-E(v, R)/k_B T) dR dv,$$

где је  $k_B$  Болцманова константа,  $T$  температура и  $E(v, R)$  енергија система. Ми из ове расподеле можемо избацити кинетичку енергију и добијамо Болцманову расподелу  $f(R) = \frac{\exp(-\sum F(r_{ij})/k_B T)}{\int \exp(-\sum F(r_{ij})/k_B T)}$  И сада можемо да израчунамо просечну потенцијалну енергију овог система користећи Метрополис алгоритам и ову расподелу.

У овом примеру разматраћемо чврсте дискове у 2Д кутији, и користећемо Болцманову расподелу да би смо симулирали овај процес. Ти дискови због коришћења Lennard-Jones потенцијала симулирају атоме или молекуле који међусовно интерагују и теже ка стабилном стању : Самим тим за вероватнију прихватања промене система добијамо:  $g = \frac{\exp(-\sum F(r'_{ij})/k_B T)}{\exp(-\sum F(r_{ij})/k_B T)} = \exp(\frac{\delta U}{k_B T})$  где је  $\delta U$  промена потенцијалне енергије. Са порастом температуре, прелаз из стања мање потенцијалне енергије у стање веће потенцијалне енергије постаје вероватнији. То значи да са порастом температуре систем тежи да се униформно расподели, из чињенице да сви прелази постају могући. Обрнуто, са смањењем температуре све теже и теже систем прелази из стања мањег потенцијала у стање већег потенцијала и полако његова потенцијална енергија опада.



Слика 6.1: Lennard-Jones потенцијал



Метрополис алгоритам модела Чврстих дискова

**begin**

**Initialize:**  $(X_1, X_2, \dots, X_n)$  Узмимо неку униформну расподелу  $N$  дискова на графику

**Initialize:**  $T$  Поставимо температуру нашег система

$b = \frac{1}{k_B T}$  Рачунање инверзне температуре

Семе алгоритма

**for**  $i := 1$  **to**  $N$  **step** 1 **do**

$X$  Неки произвољан диск

$x' = x + \delta x$  Предложимо нову  $x$  координату

$y' = y + \delta y$  Предложимо нову  $y$  координату

$\delta E_p = E_{px'y'} - E_{pxy}$  Одредимо разлику потенцијалне енергије тренутног и предложеног система

$q = \min(1, \exp(-\delta E_p * b))$  Израчунај шансу да се прихвати промена

$a = \text{rand}(0, 1)$  Изабери из униформне расподеле број од 0 до 1

**if**  $(a \leq q)$  **then** Провери да ли је корак прихваћен

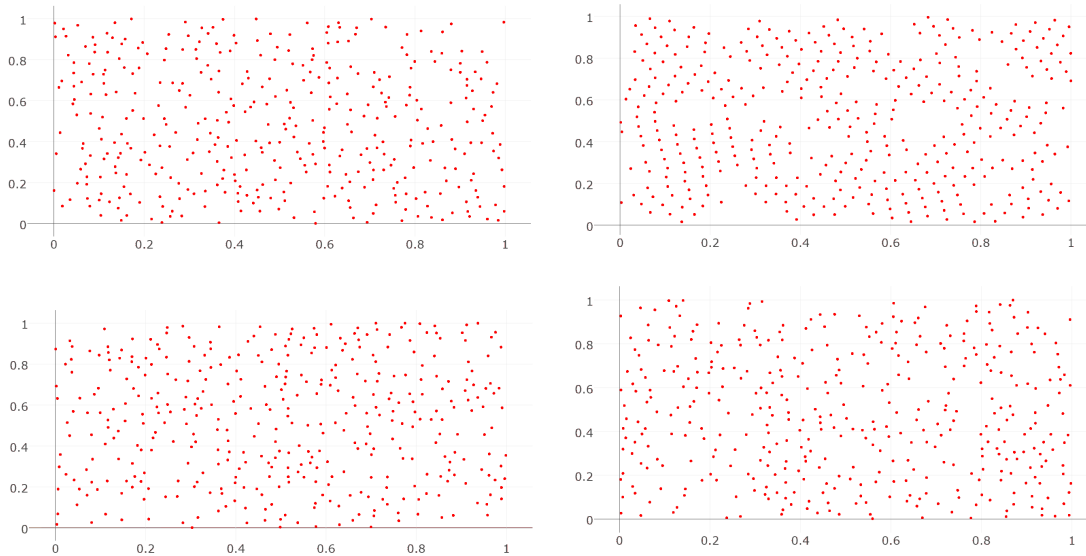
$x = x', y = y'$  Промени положај

**fi**

**od**

**end**

**Пример 6.** Нека је кутија ширине и дужине 1, а дискови полупречника 0.01. Тада би могли максимално 10000 дискова да сместимо у кутију. Узмимо онда да је 10/у оквиру квадрата странице 0.04, и да је сваки померај једнако вероватан. За констате узимамо  $e = 2.9391541 * 10^{-21}$ ,  $\sigma = 0.04$ , и  $K_b$  ће бити  $1.38064 * 10^{-19}$ . Иначе битна напомена за овај пример: Скоро све вредности параметра су процењене отприлике на основу више пробних симулација. Максималан померај диска за једну итерацију узет је да буде реда величине полупречника. Параметар  $e$  узет је на основу папира за сам Lennard-Jones потенцијал. Исто важи и за параметар  $\sigma$ . Температуру третирамо као параметар.



**Слика 6.2:** Лево слике су случајно распоређени дискови у кутији, горе десна слика је после симулације са температуром  $T = 0.0001$  у којој се добија потенцијална енергија од  $-5.7587410^{-18}$ , а доле десно после симулације на температури  $T = 10^{10}$  где добија се потенцијална енергија од  $1.0609610^{-15}$

Гледајући цртеже примећујемо како наш модел има униформну расподелу на великој температури, и у том стању има огромну потенцијалну енергију. Али када смањимо температуру нама се потенцијална енергија смањује и примећујемо како дискови изграђују веома организовану структуру која личу на кристалну. Наравно ова решења су и очекивана са обзиром на то да формула за потенцијалну енергију овог система је добра апроксимација одбојне тј привлачне силе између два молекула или атома. Самим тим ми ове чврсте дискове можемо сматрати да су атоми или молекули у затвореном систему. Изоставља се да овај модел одлично симулира међусобну интеракцију молекула.

### 6.1.2 Изингов модел

Изингов модел добио је име по физичару Ернсту Изингу, и представља математички модел феромагнетизма у статистичкој механици. Модел се састоји из промеливих које означавају спин електрона у атому, и чије су вредности  $+1$  или  $-1$ . Убацујемо их у график као тачке и омогућавамо интеракцију само између тачака које су најближи суседи. Најближи суседи у једној димензији су тачке одмах лево или десно, док у две димензије то су тачке које су горе, доле, лево и десно од посматране тачке. Овај модел нам дозвољава да уочимо фазне прелазе који се дешавају на одређеној температури. На тој температури овај модел тежи униформно да се расподели. Модел је врло једноставан и аналитички решив за 1Д и 2Д систем, док се за остале користе друге методе, Монте карло на пример.

Ми ћемо поново користити Метрополис методу за симулацију овог система, користећемо

поново Болцманову расподелу само што ћемо енергију рачунати путем ове формуле:

$$E(\sigma) = - \sum_{i < j} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j$$

Где  $J_{ij}$  представља енергију интеракције два суседа, а  $h_j$  представља енергију интеракције спина са спољашњим магнетним пољем. И сад можемо заиста уочити снагу Метрополис алгоритма. За симулацију овог система користићемо исту формулу као и за симулацију чврстих дискова, само што свака тачка овде има вредност спина и положај. И још једна разлика у односу на симулацију чврстих дискова је што уместо померања диска, у овом алгоритму мењамо вредност спина тачке на графу.

Метрополис алгоритам за Изингов модел

**begin**

**Initialize:**  $R$  Генеришемо  $N \times N$  решетку са униформно распоређеним спиновима

**Initialize:**  $T$  Подешавамо температуру система

**for**  $i := 1$  **to**  $N$  **step** 1 **do**

$x_{ij} \in R$  Изабери једану тачку из систему

$y_{ij} = -x_{ij}$  Промени знак спина у тој тачки

$q = \min(1, \exp(\frac{E_{x_{i,j}} - E_{y_{i,j}}}{k_B T}))$  Израчунај вероватноћу да се прихвати промена

$a = \text{rand}(0, 1)$  Изабери из униформне расподеле број од 0 до 1

**if** ( $a < q$ ) **then** Провери да ли је корак прихваћен

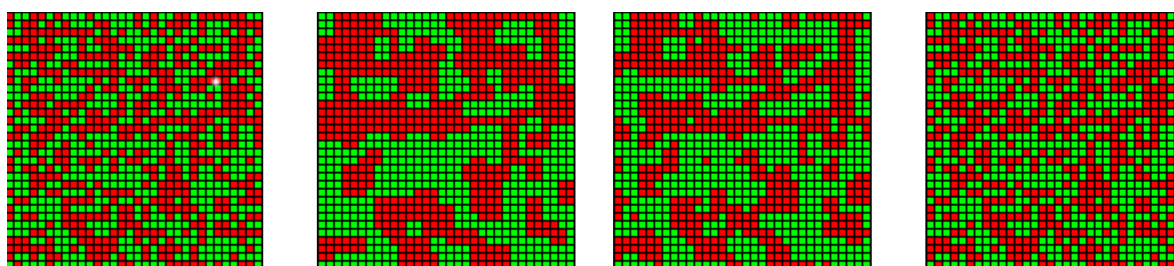
$x_{ij} = y_{ij}$  Мењамо спин изабране тачке

**fi**

**od**

**end**

**Пример 7.** Узећемо модел са  $32 \times 32$  тачака, спинове ћемо униформно расподелити и искористити овај алгоритам на различитим температурама.

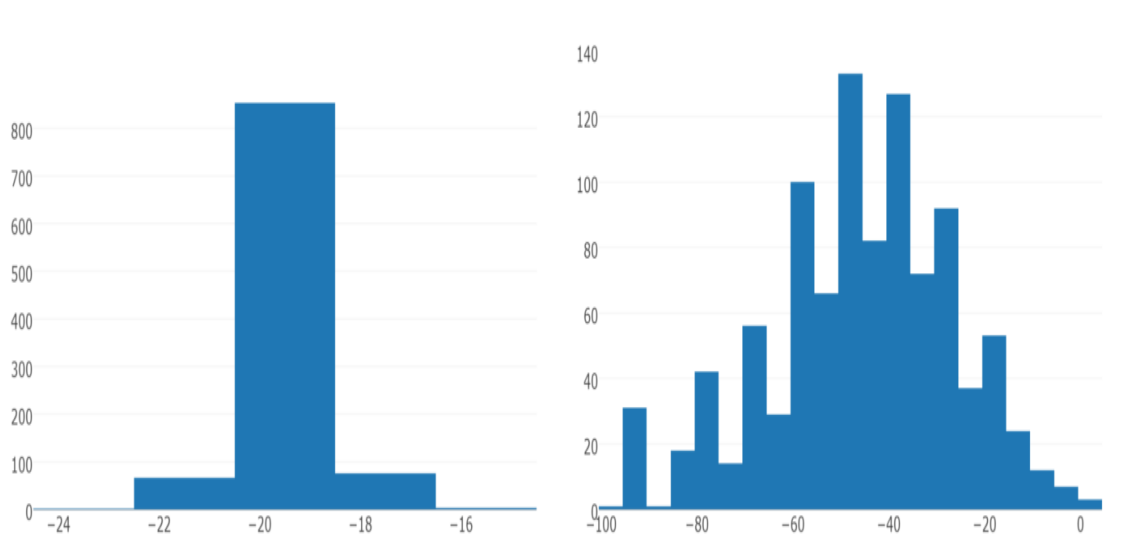


**Слика 6.3:** Изингов 2Д модел  $32 \times 32$ , прва слика оригинал, друга на  $T = 0.4$ , трећа на  $T = 160$  а четврта на  $T = 64000$

На слици зеленом бојом су обојене тачке са  $-$  спином а црвеном са  $+$  спином. Гледајући ову слику примећујемо да се на малим температурама полако групишу позитивни и негативни спинови и један почиње да преовлађује. Са порастом температуре они се униформно распоређују. Изоставља се да на веома ниским температурама почиње да преовладавају или позитивни или негативни спинови. Магнетни момент постаје

све већи и приближава се једном од екстемума. Због симетрије проблема добија се да је једнако вероватно да буду сви негативни или сви позитивни на крају. Ова конвергенција приказана је на следећем примеру.

**Пример 8.** Почетни модел ће нам бити систем са  $10 \times 10$  тачака, и његов момент је  $-20$ . Нека су температуре  $T = 0.0001$  и  $T = 10000000$  и нека је број тестова  $10000$ .



**Слика 6.4:** Изингов 2Д модел  $10 \times 10$  , прва слика  $T = 10000000$ , друга  $T = 0.0001$

Хистограм је направљен тако да је  $y$  оса представља број тестова са магнетним моментом  $x$ . Видимо да оба хистограма имају облик звона, резултат који нам ц.г.т предвиђа. Удубљена која нарушавају тај облик су последица чињеница да магнетни момент мора бити дељив са 2 у  $10 \times 10$  моделу. Тај резултат је последица да промена једног спина мења резултат за  $+2$  или  $-2$ , а знамо да је један од могућих резултата 0 (исти број  $+$  и  $-$  спинова). Због тога је број свих непарних магнетних момента 0, па се јављају 'рупе' у хистограму.

## 6.2 Симулација тродимензионих система

Метрополис алгоритам показује своју праву моћ када се користи на вишедимензионим системима. За њега прелажење из  $N$  у  $N + 1$  димензиони систем не представља ништа више него убацивање још једног случајног параметра и ако је потребно мале измене рачуна расподеле вероватноће. Тражење аналитичког решења наших проблема се невероватно закомпликује у вишедимензионом систему, самим тим је овде метрополис алгоритам још кориснији него за једнодимензиони и дводимензиони систем.

### 6.2.1 Расподела електрона на сфери

За следећу симулацију посматраћемо како се  $n$  електрона распоређују на сфери. Тражићемо да наш систем електрона има најмању могућу потенцијалну енергију, њихово најстабилније стање. Зато у овом алгоритму користими веома малу температуру као промелјиву. За потенцијалну енергију електрона користимо израз:  $E_p = \sum_{i < j} \frac{e^2}{4\pi\epsilon_0\epsilon_1 r_{i,j}}$ . За расподелу следећих стања разматрамо да се наш електрон може померати само по површини сфере и да два електрона не могу бити у истој тачки:  $g(R' \| R) = 0$  ако у  $R'$  електрон налази у један од два горе описана забрањена стања у противном  $g(R' \| R) = const$ . Битно је навести формулу за тражење произвљне тачке на сфери, који се користи овде за генерисање почетног стања и за померање електрона, пошто ћемо сваким кораком електрон преместити на случајну тачку на сфери, наравно узимајући у обзир вероватноћу преласка из једно стање у друго.

Алгоритам за генерисање  $N$  тачака на сфери

**begin**

**for**  $i := 1$  **to**  $N$  **step** 1 **do**

$r_1 = rand(0, 1)$

Генерише први случајан број од 0 до 1

$r_2 = rand(0, 1)$

Генерише други случајан број од 0 до 1

$\phi = r_1 * 2 * \pi$

Израчунај први сферни угао

$\omega = arccos(2 * r_2 - 1)$

Израчунај други сферни угао

$x_i = sin(\omega)sin(\phi)$

Израчунај x координату

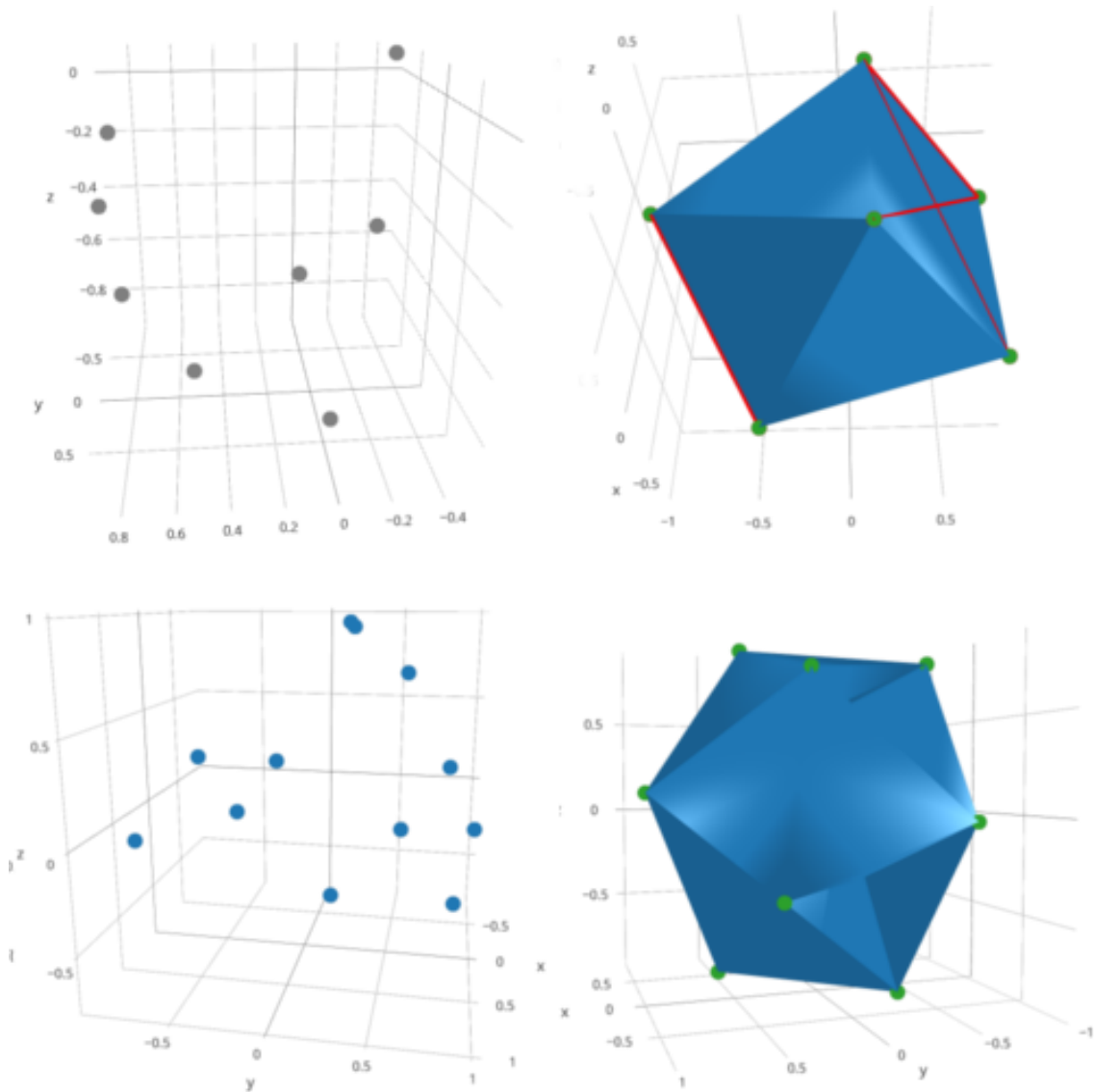
$y_i = sin(\omega)cos(\phi)$

Израчунај y координату

$z_i = cos(\omega)$

Израчунај z координату

**end**



**Слика 6.5:** Горe лево 8 наелектрисања случајно, горe десно после алгоритма, доле лево 12 наелектрисања случајно распоређено, доле десно 12 наелектрисања после алгоритма

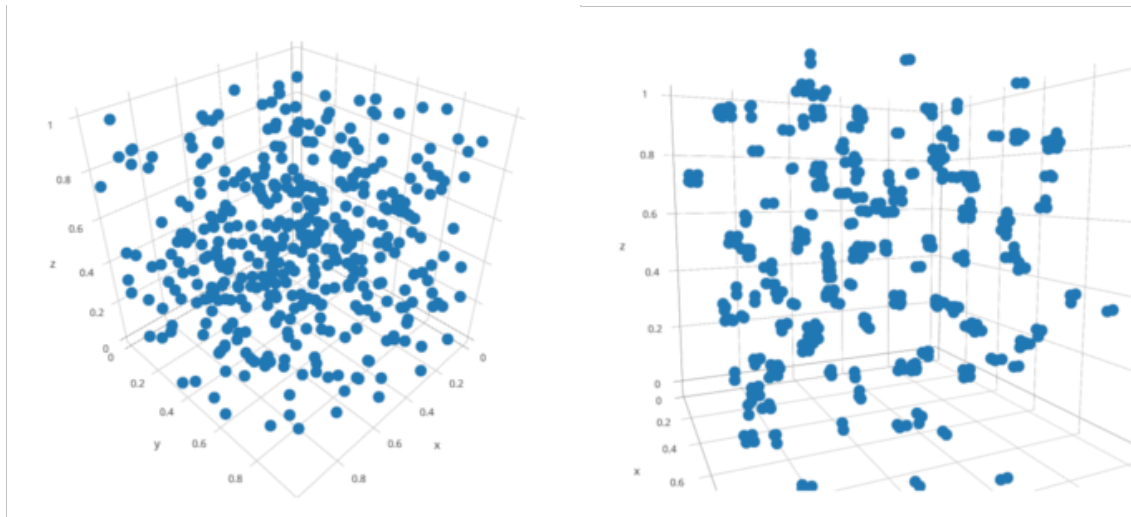
Резултат који приказује слика 6.5 поклапа се са чињеницом да се електрони распоређују по сфери у распоред у којем су најудаљењији један у другог, тако да је просечно растојање максимално. У примеру са 8 електрона добија се извртута коцка чија је горњи квадрат ротиран под углом од 45 степени у односу на доњи. За 12 електрона добијамо правилан икосаедар. За веће бројеве електрона добијају се врло комплексне конфигурације и све више личе на униформну расподелу.

### 6.2.2 Чврсте сфере у кутији

Како само име налаже овде ће бити речи о проширењу модела за чврсте дискове на 3 димензије док основна идеје, параметри и формуле остају исте. Овакава модел

још боље описује интеракцију молекула пошто и у природи се та интеракција дешава у тродимензионалном простору.

**Пример 9.** Нека имамо 400 чврстих сфера чије је полупречник  $r = 0.01$  и униформно их расподелимо у коцку чија је страница 1. Сви остали параметри су исти као за пример 6. Њихова расподела пре и после симулације је :



**Слика 6.6:** Лево униформно распоређени, десно распоред после симулације са температуром  $T = 0.000001$

Ова симулација нам показује како се компактно постављају честице при ниским температурама. Видимо такође да се групишу у мање стабилније подјединице, нешто што личи на молекуле. Зато се овај модел користио за симулацију интеракција молекула узимајући у разматрање Ван дер Валсове силе. Данас се такође користе Монте Карло методе за овај проблем, само што се користе много компликованије и прецизније формуле за енергије интеракције.

# 7

## Закључак

Током израде овог матурског рада имао сам прилике да се боље упознам са статистичком физиком и са симулирањем система. Утицај тог искуства и знања приметио сам још када сам почео да пишем други алгоритам за овај матурски рад, алгоритам за Изингов модел феромагнетизма. Време које ми је било потребно да га напишем, тестирам и скицирам резултат драстично се смањило у односу на први алгоритам, Чврсти дискови у кутији. Научио сам да користим теорему 5.4.1 за израчунавање очекиваних вредности система, као што је магнетни момент добијен Изинговим моделом и да скицирам његов хистограм. Тек после вишеструког коришћења Монте Карло метода сазнао сам разлог зашто се ове методе данас користе у симулацији разних стохастичких и вишедимензионих процеса. Пре него што сам почео истраживати ову област једине физичке симулације које сам вршио биле су оне најједноставније, слободан пад, хитац, колизија кругова и многе друге елементарне проблеме. Сада имам довољно алата да симулирам много комплексније процесе и притом знам како да добијем резултат који желим, како га сачувам и прикажем. Мислим да би свако ко воли физику и програмирање требало да зна и да проба да користи неку Монте Карло методу.



# Литература

[Shm] R. Shonkwiler and F. Mendivil, *Explorations in Monte Carlo methods*, Springer Verlag, New York, SAD 2009.

[Gre] M. Kalos and P. Whitlock, *Monte Carlo Methods, Volume 1: Basics*, A Wiley-Interscience publication, SAD 1986.

[Wol] <http://mathworld.wolfram.com/SpherePointPicking.html>

[Fsu] [https://people.sc.fsu.edu/jburkardt/c\\_src/ising\\_2d\\_simulation/ising\\_2d\\_simulation.html](https://people.sc.fsu.edu/jburkardt/c_src/ising_2d_simulation/ising_2d_simulation.html)

[Wiki] [https://en.wikipedia.org/wiki/Metropolis-Hastings\\_algorithm](https://en.wikipedia.org/wiki/Metropolis-Hastings_algorithm)

[Wiki] [https://en.wikipedia.org/wiki/Ising\\_mode](https://en.wikipedia.org/wiki/Ising_mode)

[Wiki] [https://en.wikipedia.org/wiki/Middle-square\\_method](https://en.wikipedia.org/wiki/Middle-square_method)

[Wiki] [https://en.wikipedia.org/wiki/Uniform\\_distribution\\_\(continuous\)](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous))

[Mat] <http://www.mathpages.com/home/kmath005/kmath005.htm>

[Link] <http://skyeome.net/BenderdeMoll01/thesisImages/markov.JPG>