

České vysoké učení v Praze
Fakulta elektrotechnická



Diplomová práce

Monte Carlo simulace fázových přechodů

Michal Kaňok

Vedoucí práce: Prof. RNDr. Petr Kulhánek, CSc.

Studijní program: Elektrotechnika a informatika

Obor: Výpočetní technika

Květen 2008

Pod'akovanie

Chcel by som poďakovať týmto ľuďom: vedúcemu práce Prof. RNDr. Petrovi Kulhánkovi, CSc., mojim rodičom, bratovi, Michalovi Chrenovi, Bc. Jozefovi Krčmárikovi, Ing. Michalovi Marčišovskému, Miroslavovi Mikšíkovi, Bc. Radimovi Stránskemu, Ing. Tomášovi Škereňovi a Miroslavovi Urbánkovi.

Prehlásenie

Prehlasujem, že som svoju diplomovú prácu vypracoval samostatne a použil som iba podklady uvedené v priloženom zozname.

Nemám závažný dôvod proti použitiu tejto školskej práce v zmysle §60 Zákona č. 121/2000 Sb., o autorskom práve, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon).

Vyhradzujem si však autorské právo na zdrojové kódy programu pre ich neskoršie uvoľnenie pod jednou z BSD licencií.

V Prahe, dňa 23. 05. 2008

Abstrakt

Táto diplomová práca sa zaoberá využitím metód Monte Carlo v štatistickej fyzike. Obsahuje analýzu vybraných metód Monte Carlo. Ďalej vybrané partie zo štatistickej fyziky použité na riešenie zvoleného fyzikálneho problému - simulácie fázových prechodov feromagnetických látok so zmenou teploty pomocou Metropolisovho algoritmu. A na koniec popis a implementáciu programu, ktorý zvolený problém vizualizuje.

Abstract

This work applies Monte Carlo methods to the problem of simulation of phase transitions of ferromagnetic materials during the cooling using the Metropolis algorithm. It analyzes few chosen Monte Carlo methods and some physical models used for description of the subject. It also includes description and implementation of software which simulates the problem.

Obsah

Zoznam obrázkov	xi
Zoznam tabuliek	xi
1 Úvod	1
1.1 Konvencie.....	1
1.2 Zoznam skratiek	1
1.3 Postavenie metód Monte Carlo v štatistickej fyzike	2
1.4 Štatistická fyzika, fázový prechod	2
1.5 História metód Monte Carlo a ich využitie	3
2 Modely	4
2.1 Klasické mriežkové modely	4
2.2 Isingov model.....	5
2.3 Pottsov model.....	6
2.4 Z_Q model.....	7
2.5 Heisenbergov model.....	8
2.6 Trojrozmerné modely	9
2.7 Ďalšie modely.....	11
3 Metódy Monte Carlo	12
3.1 Monte Carlo a výpočet π	12
3.1.1 Monte Carlo integrácia	12
3.1.2 Buffonova ihla	13
3.2 Generátory náhodných čísiel.....	14
3.2.1 Lineárne kongruenčné generátory.....	15
3.2.1.1 IBM RANDU.....	16
3.2.2 Kombinované lineárne kongruenčné generátory	18
3.2.3 Spomalené Fibonacciho generátory	18
3.2.3.1 Generátory so všeobecným posuvným registrom s návratom	19
3.3 Monte Carlo metódy pre mriežkové modely.....	19

3.3.1 Metropolisov algoritmus	20
3.3.2 Heat-bath (teplotná kúpeľ)	22
3.3.3 Metóda s hyperbolickou tangentou	22
3.3.4 Simulované ochladzovanie (simulated annealing)	22
3.3.4.1 Problém obchodného cestujúceho	23
3.3.5 Iné metódy	24
4 Program Monte Carlo simulátor	24
4.1 Objekt implementácie	24
4.2 Popis implementácie	25
4.2.1 Popis tried a ich grafická reprezentácia	25
4.2.1.1 Trieda Cmodel	26
4.2.1.2 Triedy užívateľského rozhrania	28
4.2.1.3 Triedy vykresľovacie	28
4.2.1.4 Triedy CmodelGenerator a CmodelSolver	31
4.2.2 Vstupy programu	32
4.2.3 Výstupy programu	32
4.2.4 Scenár použitia	32
4.3 Snímky programu	33
4.4 Časová náročnosť jednotlivých modelov	40
5 Záver	40
5.1 Snímky skutočných feromagnetických látok	41
Literatúra	42
Prílohy	44
A Inštalácia programu	44
B Ovládanie programu	44
C Obsah priloženého CD	44

Zoznam obrázkov

2.1: Dvojrozmerný Isingov model s náhodnou orientáciou spinov.	5
2.2: Dvojrozmerný Pottsov model s doménami, $Q = 8$	7
2.3: Dvojrozmerný Z_Q model v soft fáze, $Q = 16$	8
2.4: Dvojrozmerný Heisenbergov model s vírmi.	9
2.5: Trojrozmerný Isingov model s doménami.	10
2.6: Trojrozmerný Heisenbergov model s vírmi.	11
3.1: Grafické znázornenie bodov v štvrt'kruku a mimo neho.	12
3.2: Buffonova ihla.	14
3.3: Distribúcia bodov podľa IBM RANDU v priestore [5].	17
4.1: Grafická reprezentácia triedy Cmodel a jej dedičných tried.	27
4.2: Reprezentácia tried Diploma, CmodelWidget, CgrafWidget a užívateľského rozhrania.	30
4.3: Reprezentácia tried CmodelGenerator a CmodelSolver.	31
4.4: Screenshot programu po spustení, Isingov model.	34
4.5: Screenshot programu, trojrozmerný Heisenbergov model.	35
4.6: Screenshot programu, trojrozmerný priblížený Isingov model po skončení simulácie. ...	36
4.7: Screenshot programu, dvojrozmerný Heisenbergov model s rozmermi 150×150, s vypnutými oknami grafov a oknom nastavení.	37
4.8: Screenshot programu z priebehu simulácie s rôzne posunutými oknami.	38
4.9: Screenshot programu zobrazujúci "falošné" celkom rozbalené menu a vyskakovacie okno "About Application".	39
5.1: Snímky magnetických materiálov [15].	41

Zoznam tabuliek

4.1: Časy výpočtov.	40
--------------------------	----

1 Úvod

1.1 Konvencie

V tejto práci používam slovenský preklad cudzojazyčných, hlavne anglických, technických a fyzikálnych termínov, pokiaľ v slovenčine rozumný ekvivalent existuje a je používaný. V opačnom prípade používam pôvodný termín v inom jazyku a v zátvorke uvádzam doslovný preklad. Ak je to možné, používam slovenské skloňovanie cudzích slov (napr. "Isingov model" namiesto "Ising model"). V texte používam anglickú transkripciu japonských mien do latinky. Čísla menšie alebo rovné dvadsať sú v texte písané slovom, ostatné čísla, rovnice, čísla kapitol a čísla obrázkov číslicami. Algoritmy sú napísané pseudokódom podobným jazyku C++. Som autorom všetkých použitých obrázkov, pokiaľ nie je uvedené inak.

1.2 Zoznam skratiek

Fyzikálne veličiny a konštanty:

C - merná tepelná kapacita
 E - energia
 H - hamiltonián
 k - Boltzmannova konštanta
 K - konfigurácia
 M - magnetizácia
 T - teplota
 χ - magnetická susceptibilita

Jazykové skratky:

apod. - a podobne
atď. - a tak ďalej
napr. - napríklad
resp. - respektíve
t.j. - to jest
tzv. - tak zvaný

1.3 Postavenie metód Monte Carlo v štatistickej fyzike

Metódy Monte Carlo tvoria najväčšiu a najdôležitejšiu časť numerických metód riešiacich úlohy štatistickej fyziky [1]. Metódami Monte Carlo označujeme množstvo algoritmov, ktorých zjednocujú tieto spoločné znaky:

- k výpočtu potrebujú postupnosť náhodných čísiel,
- čím je použitý generátor pseudonáhodných čísiel lepší, tým dáva algoritmus lepšie výsledky,
- čím viac krokov s ďalšími vygenerovanými pseudonáhodnými číslami spravíme, tým viac sa blížíme k správne riešeniu, alebo je riešenie úlohy presnejšie.

Numerické metódy možno vo fyzike použiť tam, kde analytické riešenie problému neexistuje, nie je známe alebo je pomalé. Pre množstvo čiastkových výpočtov, ktoré môžu pri uskutočnení experimentu, simulácie alebo globálneho výpočtu numerickými metódami vzniknúť, je často potrebné pri práci s nimi použiť výpočtovú techniku. Vďaka tomu, že Monte Carlo simulácie môžu mať na špecifických modeloch lineárnu časovú zložitosť, na simuláciu menších modelov, rádovo do 10^6 častíc, nám dnes stačí bežný počítač. Výpočty na mriežkových modeloch [kapitola 2.1] sa dajú paralelizovať.

1.4 Štatistická fyzika, fázový prechod

Štatistická fyzika tvorí dôležitú časť teoretickej fyziky. K popisu a výpočtu makroskopických vlastností látok a vzťahov medzi nimi využíva vlastnosti, deje a interakcie častíc látok mikroskopických rozmerov. Alebo naopak, z makroskopických vlastností sa snaží odvodiť vlastnosti mikroskopické. Je teda akýmsi pojítkom medzi svetom častíc, atómov, mikrosvetom a svetom, ktorý poznáme a pozorujeme dennodenne okolo nás, spojitým svetom, makrosvetom. K tomu všetkému používa štatistická fyzika aparát matematickej štatistiky a teórie pravdepodobnosti.

Dnes je stredobodom pozornosti štatistickej fyziky výpočet vlastností kondenzovaného stavu hmoty, t.j. hlavne kvapalín a pevných látok [1]. Skúmaná látka sa vždy nachádza v nejakom stave, ktorý sa nazýva fáza. Fáza je charakterizovaná mikroskopickým usporiadaním častíc systému v priestore a s tým súvisiacimi makroskopickými vlastnosťami. Typickým príkladom fáz sú skupenstvá látky (pevné, kvapalné a plynné), alebo paramagnetická a feromagnetická fáza látky, alebo napr. rozdielne fázy uhlíka (diamant a grafit), ktoré majú odlišné vlastnosti pri rovnakej teplote a ostatných vonkajších fyzikálnych veličinách. Pri zmene niektorej z týchto vonkajších veličín môže dôjsť a veľmi často aj dochádza k zmene fázy. Takýto jav nazývame fázový prechod. Pri fázovom prechode zmena štruktúry látky výrazne ovplyvní nejakú jej makroskopickú vlastnosť. V tejto práci sú skúmanými vlastnosťami vnútorná energia E , merná tepelná kapacita C , magnetizácia M

a magnetická susceptibilita χ feromagnetických látok v závislosti na zmene teploty T . Použitím Monte Carlo simulácie je možné pozorovať zmeny všetkých spomenutých veličín a z ich grafu a vizualizácie mriežky rozoznať fázové prechody.

Kľúčovým problémom v skúmaní látok z mikroskopického hľadiska je to, že látky sú zložené z veľkého množstva malých častíc, typicky z atómov alebo molekúl. V skúmanom systéme bývajú častice rovnaké, alebo je ich iba niekoľko málo druhov. Pokiaľ by sme v simulácii uvažovali interakciu medzi každou dvojicou častíc v systéme, problém by bol už pri relatívne malom počte častíc prakticky neriešiteľný. Preto boli vytvorené modely, ktoré popis interakcií a štruktúru látky zjednodušujú, avšak dosiahneme nimi dobré výsledky. Napr. uvažujeme iba interakciu najbližších častíc, vzdialené opomenieme. Náhodný výskyt častíc nahradíme pravidelným mriežkovým modelom apod.

1.5 História metód Monte Carlo a ich využitie

Pokiaľ považujeme za základ výpočtu v metódach Monte Carlo použitie nejakého náhodného javu, môžeme povedať, že základ metód Monte Carlo je známy zhruba už od osemnásteho storočia. Pojem Monte Carlo ako označenie skupiny algoritmov, bol spopularizovaný okruhom fyzikov a matematikov (Stanislaw Ulam, Enrico Fermi, John von Neumann, Nicolas Metropolis) pôsobiacich v Národnom Laboratóriu v Los Alamos v Spojených Štátoch. Údajne úplne prvý, kto pojem Monte Carlo za týmto účelom použil, bol v roku 1949 práve Nicolas Metropolis [1]. Predtým sa pre označenie týchto metód používal pojem štatistické vzorkovanie. Prvotné využitie však nebolo vo fyzike, ale v matematike. Štatistické vzorkovanie sa používalo na výpočet určitých integrálov, ktoré sa nedali dopočítať inak. Ostatne metódy Monte Carlo sa k rovnakému účelu používajú dodnes, hlavne u "škaredých" funkcií a funkcií viacerých premenných.

Peknými vzorovými príkladmi na aplikáciu Monte Carlo metód sú dva spôsoby výpočtu π , pokusom nazývaným Buffonova ihla, alebo priamo z funkcie kružnice v kartézskom súradnom systéme [kapitola 3.1]. V 30tych rokoch použil Enrico Fermi metódy Monte Carlo k popisu vlastností novo objavenej častice, neutrónu. Podľa pamätí jeho študenta a spolupracovníka, Emilia Segra, to boli postupy presne také (až na názov), aké dnes pod pojmom Monte Carlo chápeme. Enrico Fermi však tieto metódy nikdy nepublikoval, asi preto, že to nepovažoval za dôležité a k publikovaniu mu postačili získané výsledky [1]. K rozvoju metód došlo pri práci na Projekte Manhattan, pri výrobe prvej nukleárnej zbrane, atómovej bomby. Neskôr, v 50tych rokoch, k podobnému účelu, vodíkovej bomby. Boli to práve Monte Carlo metódy, ktoré spustili vývoj generátorov pseudonáhodných čísiel, ktoré sú oveľa rýchlejšie než dovtedy používané tabuľky náhodných čísiel. Zaujímavosťou je, že numerické metódy vyťažovali v tom čase prvý elektronický počítač ENIAC.

Okrem širokého záberu v matematike a fyzike sa metódy Monte Carlo používajú v množstve iných vedných odboroch, námatkovo napr. v ekonomike, v počítačovej grafike na raytracing (sledovanie lúča), v umelej inteligencii, v počítačových hrách, na simuláciu dopravy, v architektúre a dizajne, na generovanie filmových efektov atď.

2 Modely

2.1 Klasické mriežkové modely

Ako už bolo spomenuté, štruktúru a správanie látky je možné a niekedy nutné upraviť tak, aby bolo danú úlohu možné riešiť a vyriešiť. Akým spôsobom, to už záleží od konkrétnej úlohy. Často je možné aplikovať na jednu úlohu viacero rôznych aproximácií, modelov.

V niektorých simuláciách vychádzajú z toho, že elementy látky (napr. atómy, alebo magnetické spiny) spolu interagujú. Z mikroskopického hľadiska je konfigurácia (okamžitý stav systému) daná hodnotou premenných práve týchto elementov. Základnou fyzikálnou veličinou, ktorá popisuje systém v štatistickej fyzike je hamiltonián H . Hamiltonián je veličina popisujúca celkovú energiu systému, je daný súčtom kinetickej (z hybnosti) a potencionalnej energie častíc [3].

Množinu konfigurácií systému (t.j. všetky stavy, v ktorých sa môže systém nachádzať) so svojim pravdepodobnostným rozdelením nazývame štatistický súbor. Štatistické súbory je možné triediť podľa rôznych hľadísk:

- na klasické a kvantové (klasické sú výpočtovo jednoduchšie),
- či je systém v stave rovnováhy alebo nerovnováhy (rovnovážny systém je výpočtovo jednoduchší),
- na modely spojité a diskrétne (podľa oboru hodnôt súradníc elementov v priestore; napr. v kartézskych súradniciach, diskrétne je výpočtovo výrazne jednoduchší).

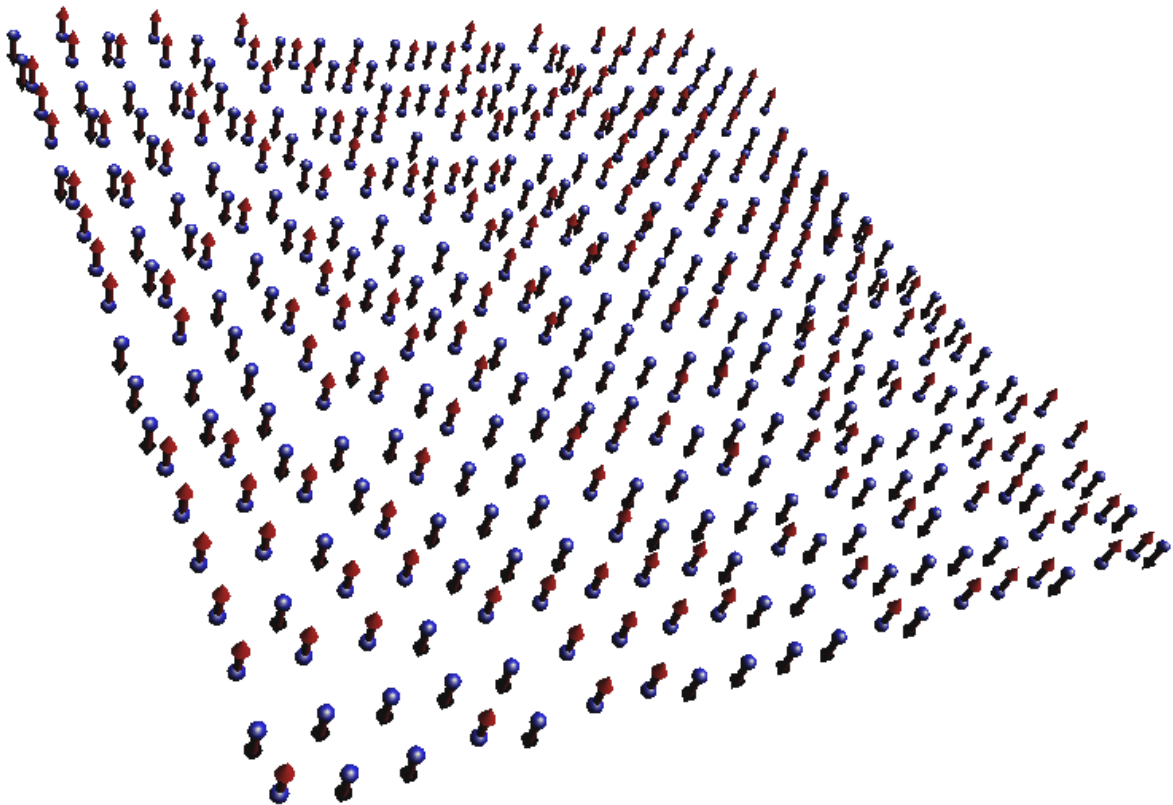
Špecifickým prípadom diskrétnych modelov sú mriežkové modely. V tejto práci je detailnejšie rozobratý a naimplementovaný klasický mriežkový model v rovnováhe, s pravouhlou mriežkou a interakciou iba najbližších elementov. Ďalšou vlastnosťou tohto modelu je použitie periodických okrajových podmienok. To znamená, že susedia na okraji mriežky majú ako susedov častice z opačného konca mriežky (napr. v dvojrozmernom modeli, kde x je rozmer v smere osi x , y je rozmer v smere osi y , častica so súradnicami $[0,0]$ bude mať ako susedov častice so súradnicami $[0,1]$, $[1,0]$, $[x-1,0]$, $[0, y-1]$)

2.2 Isingov model

Tento model vznikol špeciálne pre použitie pri štúdiu magnetizmu. Je to najjednoduchší z nasledujúcich modelov. Každé dve susedné častice na susedných vrchoch mriežky označíme $\langle a, b \rangle$. Rovnica popisujúca interakčnú energiu vyzerá pre Isingov model takto:

$$W_0 = -J \sum_{\langle a, b \rangle} m_a m_b ; \quad m_a \in \{-1, +1\} . \quad (1)$$

Všetky konštanty sú zahrnuté v J . Dva rovnako orientované magnetické spiny prispievajú k energii systému hodnotou $-J$, dva opačne orientované spiny hodnotou $+J$. Tento model má dve fázy, pri vysokých teplotách je to chaotická fáza a pri nízkych usporiadaná fáza. V chaotickej fáze sú magnetické spiny orientované náhodne. V usporiadanej fáze vznikajú oblasti s rovnakou orientáciou spinov, ktoré sa nazývajú Weissove domény. K fázovému prechodu medzi chaotickou a usporiadanou fázou dochádza pri Curieho teplote. Látky s $J > 0$ majú feromagnetické vlastnosti, látky s $J < 0$ antiferomagnetické vlastnosti [2].



Obrázok 2.1: Dvojrozmerný Isingov model s náhodnou orientáciou spinov.

2.3 Pottsov model

Pottsov model je podobný Isingovmu modelu, ale spiny môžu nadobúdať nielen dve diskkrétne hodnoty ale Q hodnôt, kde Q je prirodzené číslo. Hodnoty Q predstavujú smer spinov. Pre $Q = 2$ je Pottsov model, až na výpočet interakčnej energie, totožný s Isingovým. Podobne ako Isingov model, má Pottsov model usporiadanú vysokoteplotnú chaotickú fázu a nízokoteplotnú usporiadanú fázu. K fázovému prechodu dochádza, rovnako ako v Isingovom modeli, pri Curieho teplote. Dva rovnako orientované spiny prispievajú do energie hodnotou $-J$, dva rôzne orientované spiny do energie neprispievajú [2].

$$W_0 = -J \sum_{\langle a,b \rangle} \delta_{m_a m_b} ; \quad \alpha_a = \frac{2\pi}{Q} m_a ; \quad m_a \in \{1, \dots, Q\} , \quad (2)$$

kde α_a je otočenie spinu.



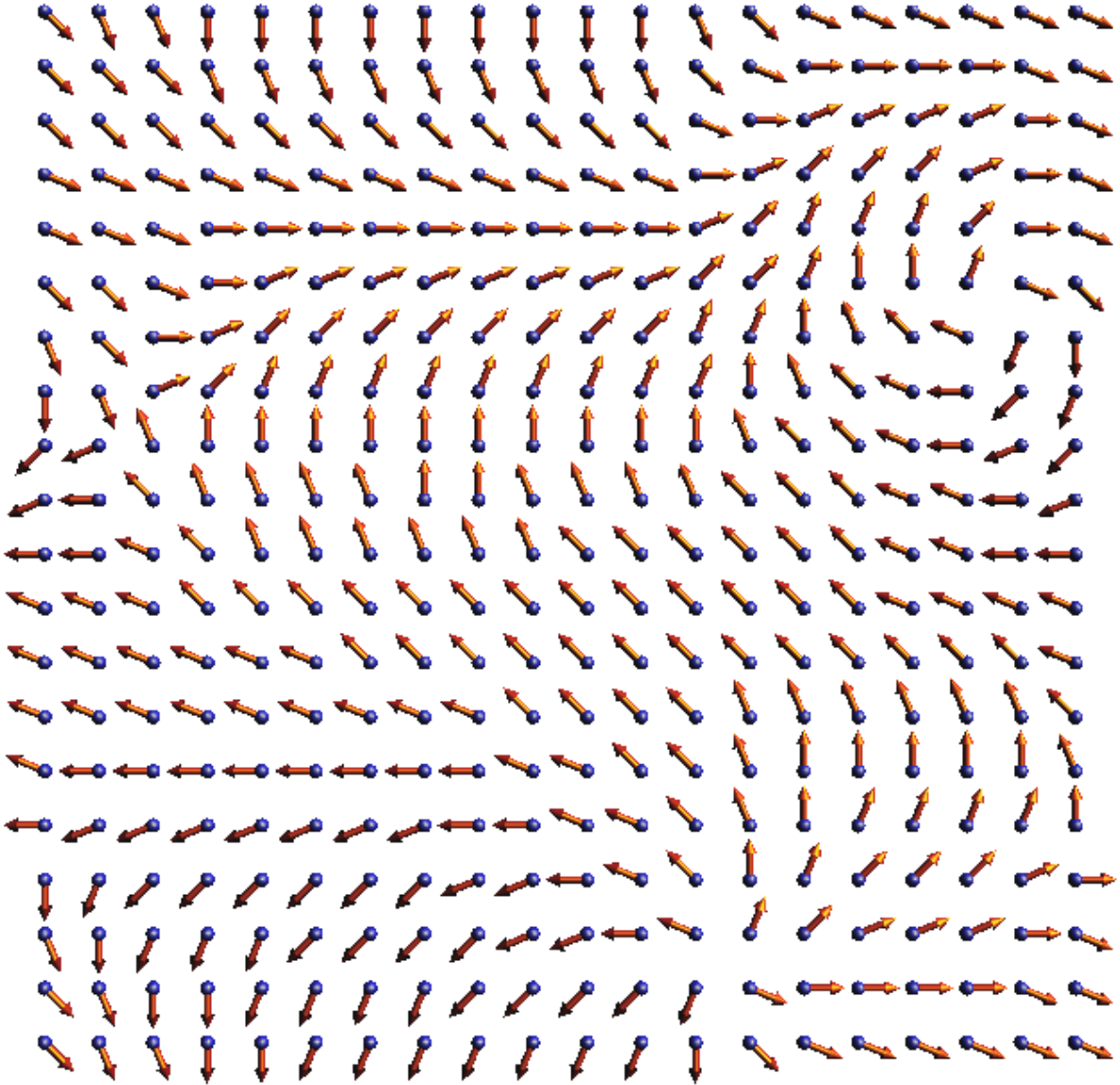
Obrázok 2.2: Dvojrzmerný Pottsov model s doménami, $Q = 8$.

2.4 Z_Q model

Z_Q model je podobný Pottsovmu modelu, taktiež je to Q -smerový model. Oproti Pottsovmu modelu sa líši v predpise interakčnej energie, ktorá je daná kosínom uhla dvoch susedných spinov. Z_Q model má tri fázy, pri vysokých teplotách, pri stredných teplotách a pri nízkych teplotách. Pri vysokých teplotách chaotickú fázu s náhodnou orientáciou spinov, nízkoteplotnú usporiadanú fázu s doménami a fázu pri stredných teplotách nazývanú soft fáza. Fázový prechod medzi nízkoteplotnou a soft fázou je Curieho prechod, prechod medzi soft fázou a vysokoteplotnou fázou sa nazýva Kosterlitz-Thoulessov prechod [2].

$$W_0 = -J \sum_{\langle a,b \rangle} \cos(\alpha_a - \alpha_b); \quad \alpha_a = \frac{2\pi}{Q} m_a; \quad m_a \in \{1, \dots, Q\}, \quad (3)$$

kde α_a je otočenie spinu.



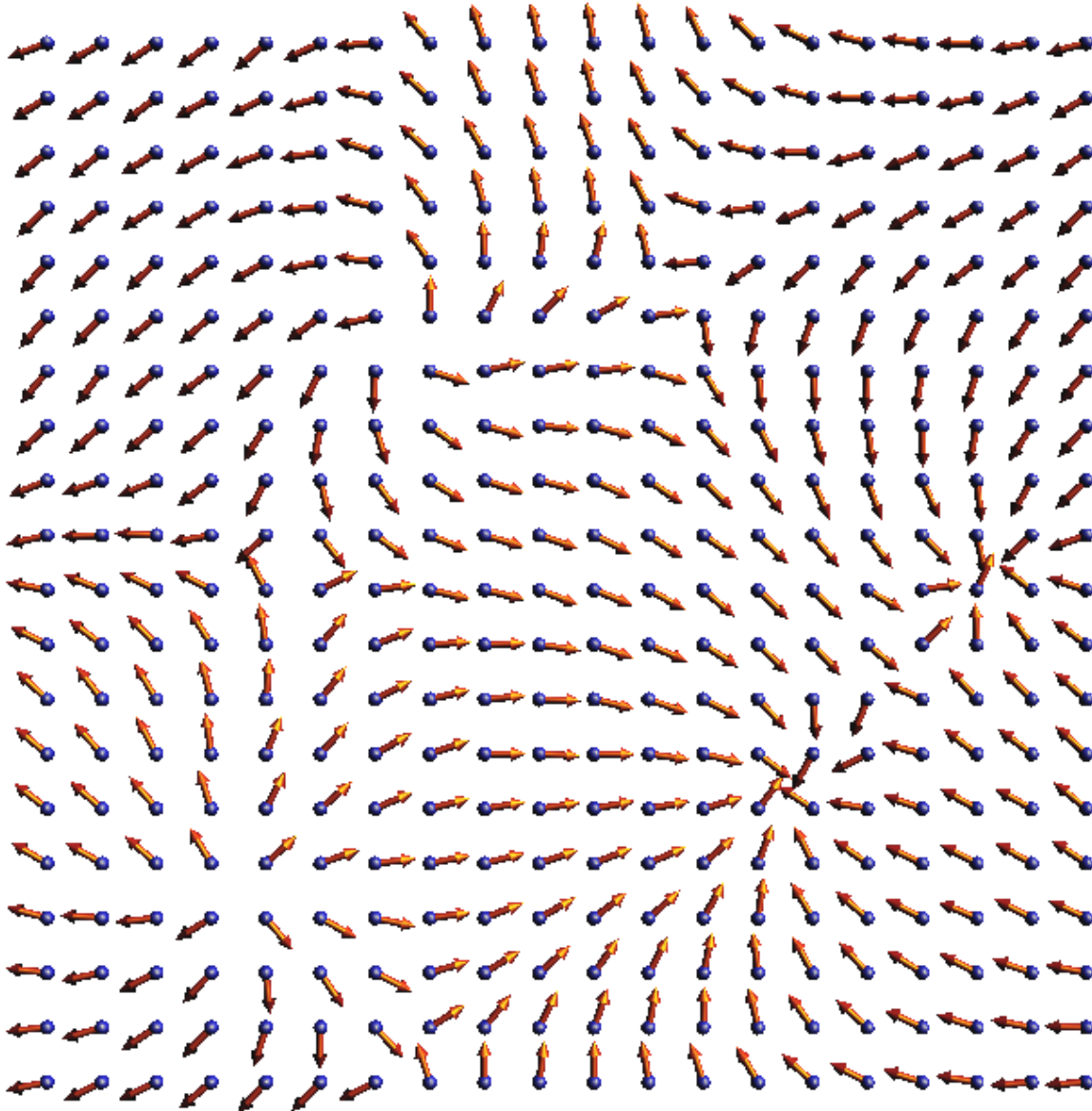
Obrázok 2.3: Dvojrozmerný Z_Q model v soft fáze, $Q = 16$.

2.5 Heisenbergov model

Heisenbergov model mení rozdelenie spinov Z_Q modelu z diskrétného na spojité, čiže je to Z_Q model s $Q \rightarrow \infty$. V našom prípade do rozsahu typu premennej použitej k reprezentácii spinu (double). Spiny teda môžu nadobúdať akýkoľvek smer. Interakčná energia je daná kosínom uhla. Model má dve fázy. Vysokoteplotnú neusporiadanú fázu a nízkoteplotnú soft

fázu, pri ktorej vznikajú víry a spinové vlny. Fázový prechod medzi nimi je Kosterlitz-Thoulessov prechod [2].

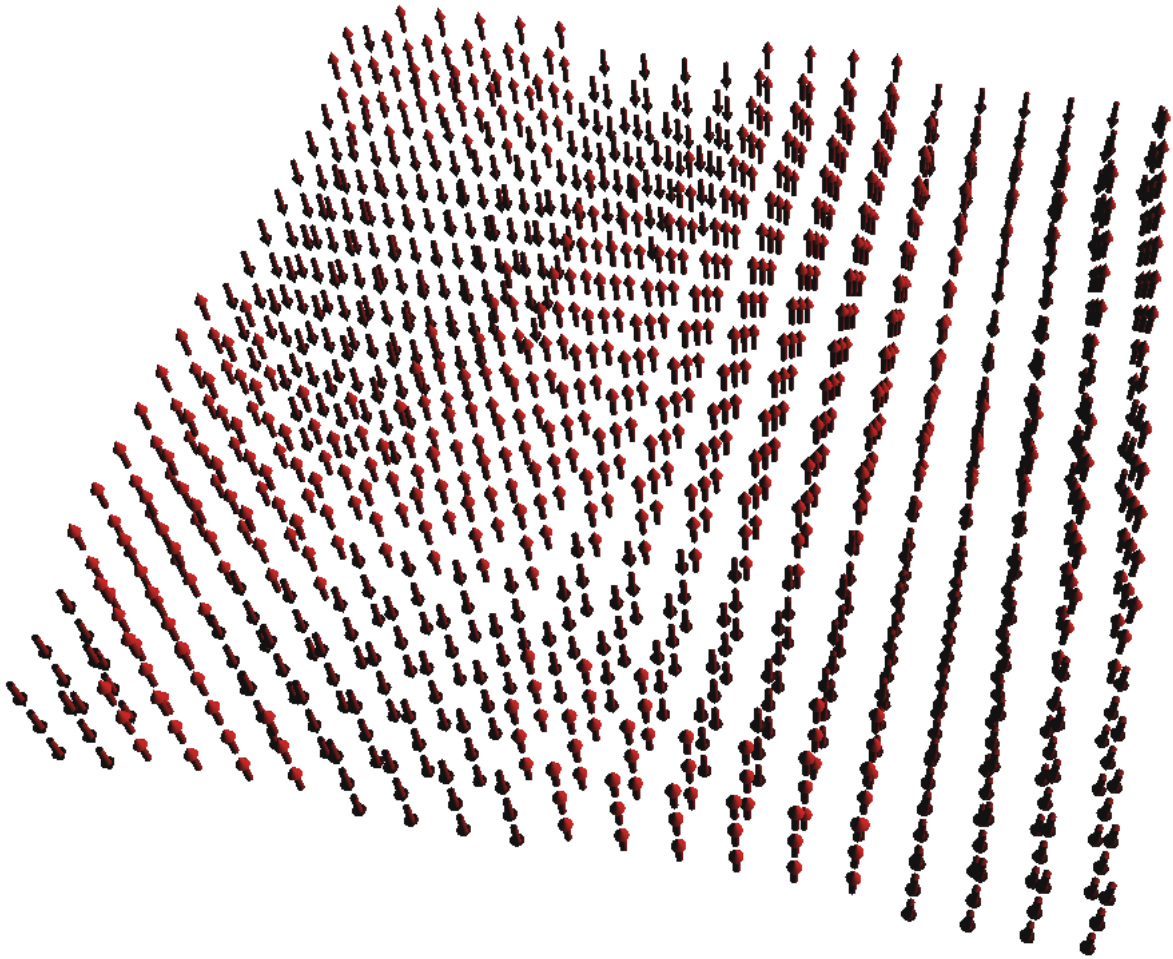
$$W_0 = -J \sum_{\langle a,b \rangle} \cos(\alpha_a - \alpha_b); \quad \alpha_a \in [0, 2\pi). \quad (4)$$



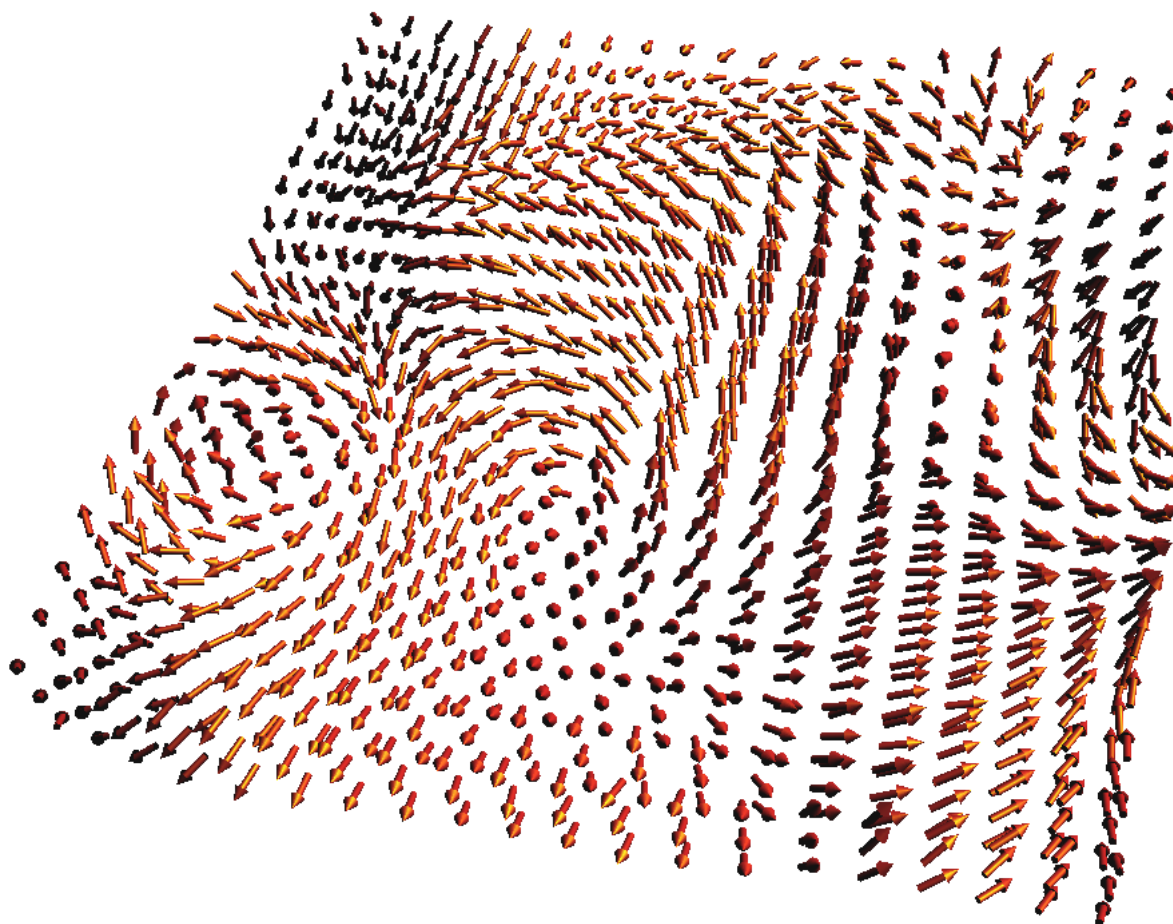
Obrázok 2.4: Dvozmerný Heisenbergov model s vírmi.

2.6 Trojzmerné modely

V trojzmernom priestore na trojzmernej mriežke sa situácia mierne výpočtovo komplikuje. Každý atóm má šesť namiesto štyroch susedov. V Pottsovom a Z_Q modeli treba riešiť problém rovnomernej distribúcie spinov na trojzmernej guli.



Obrázok 2.5: Trojrozmerný Isingov model s doménami.



Obrázok 2.6: Trojrozmerný Heisenbergov model s vírmi.

2.7 Ďalšie modely

Tieto modely nie sú v mojej práci podrobnejšie popísané, ani naimplementované v programe, uvádzam ich iba ako zaujímavosť.

- Spinové sklá

Konštanta J v predpise interakčnej energie sa líši s každou dvojicou spinov a a b . Môže mať náhodný charakter:

$$W_0 = - \sum_{\langle a,b \rangle} J_{ab} f(m_a, m_b). \quad (5)$$

- XY model

V anglickej literatúre sa takto označuje model, ktorý v tejto práci volám dvojrozmerný Heisenbergov model.

- N-vektorový model

N-vektorový model je všeobecné označenie zahŕňajúce Isingov model, Heisenbergov dvojrozmerný (XY model) a Heisenbergov trojrozmerný model.

Okrem spomenutých modelov existujú aj ďalšie modely a ich modifikácie. Za všetky spomeniem COP Isingov model, ktorý modeluje reálny plyn mapovaním častíc na vrcholy mriežky.

3 Metódy Monte Carlo

3.1 Monte Carlo a výpočet π

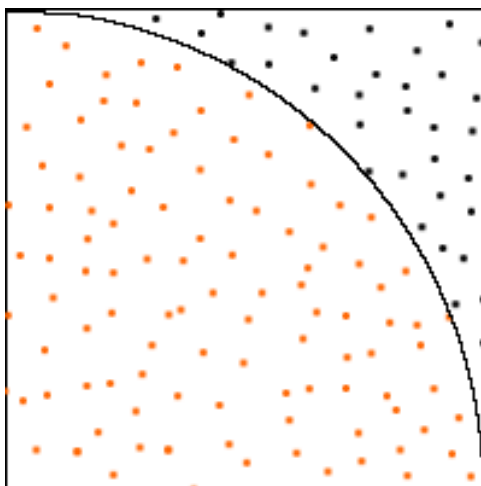
Veľmi peknými príkladmi na demonštráciu metód Monte Carlo sú dva spôsoby výpočtu π , Monte Carlo integrácia a Buffonova ihla.

3.1.1 Monte Carlo integrácia

Predstavme si štvrtkruh vpísaný do štvorca, posunutý do počiatku súradnej sústavy [obrázok 3.1]. Do štvorca budeme náhodne triafať body a sčítať tie, ktoré padli do vnútra kruhu. Keď uskutočníme týchto krokov dostatočne veľa, môžeme s určitou presnosťou vypočítať π .

$$\pi = 4 \frac{\text{body}_v_{\text{štvrtkruhu}}}{\text{všetky_body}}.$$

Či je bod vnútri štvrtkruhu alebo nie, sa sá jednoducho spočítať z rovnice kruhu $x^2 + y^2 \leq r^2$, ak položíme $r = 1$, úloha sa ešte zjednoduší.



Obrázok 3.1: Grafické znázornenie bodov v štvrtkruhu a mimo neho.

Algoritmus:

```
int n = 1000;           //počet všetkých bodov
int body_v_kruhu = 0;  //počet bodov padnuvších do vnútra
                        //štvrtkruhu
double x;              //x-ová súradnica bodu
double y;              //y-ová súradnica bodu
double gsl_rng_uniform(); //funkcia vracajúca číslo od 0 do 1
for (int i = 0; i < n; i++) {
    x = gsl_rng_uniform();
    y = gsl_rng_uniform();
    if ((x*x+y*y) < 1) {
        body_v_kruhu++;
    }
}
double pi = 4 * body_v_kruhu / n;
```

Je zrejmé, že čím väčšie je číslo n , tým presnejšie bude naše π . Chyba experimentu je nepriamo úmerná n^2 . Samozrejme predpokladáme, že máme dobrý generátor pseudonáhodných čísiel [kapitola 3.2].

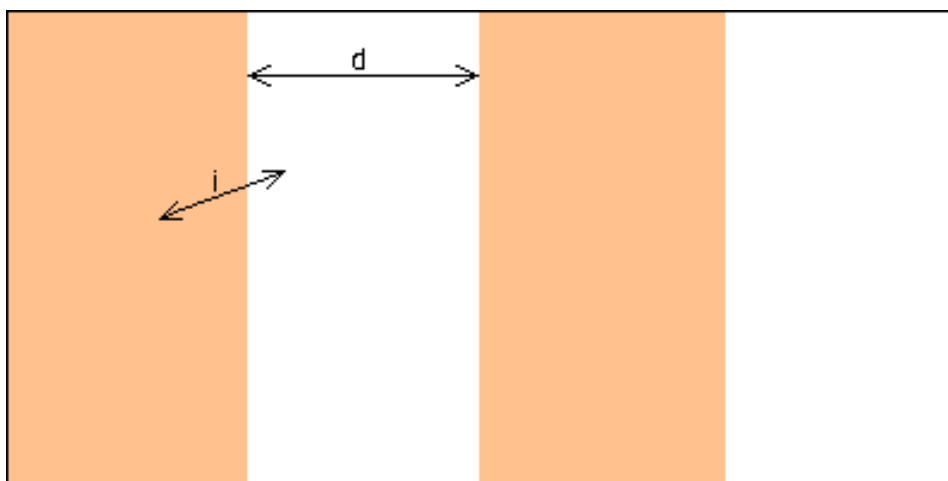
3.1.2 Buffonova ihla

Tento experiment je pomenovaný podľa Georges-Louisa Leclerca, Comte de Buffon. Ten ako prvý, v roku 1777, prišiel na to, že ak máme tenkú ihlu dĺžky i a mapu pruhov, medzi ktorými sú rozostupy dĺžky $d \geq i$, tak pravdepodobnosť toho, že dopadnuvšia ihla pretne dva pruhy je

$$p = \frac{2i}{\pi d}. \quad (6)$$

V roku 1820, Pierre-Simon, marquis de Laplace z tejto rovnice odvodil vzťah pre π a počet náhodných hodov. Ak budeme ihlu n krát náhodne hádzať na mapu a počet prípadov, v ktorých ihla pretne dva pruhy označíme h , potom

$$\pi = \frac{2in}{dh}. \quad (7)$$



Obrázok 3.2: Buffonova ihla.

V roku 1901 publikoval taliansky matematik Mario Lazzarini, že vďaka tomuto experimentu spočítal veľkosť π pre $n = 3408$ ako $355/113$. Predpokladá sa, že si výsledky upravil, alebo dokonca vymyslel, ale každopádne hodnota $355/113$ je so skutočným π identická na prvých šesť desatinných miest [1].

3.2 Generátory náhodných čísiel

Pri práci s Monte Carlo metódami sa používajú výhradne softvérové generátory náhodných čísiel. Softvérové generátory sú rôzne počítačové algoritmy. Okrem softvérových generátorov existujú aj hardvérové generátory. Hardvérové generátory dodávajú náhodné čísla podľa zmien niektorých mikroskopických fyzikálnych, teoreticky nepredvídateľných javov. Tieto javy, tvoriace ich základ, sú napr. elektrický šum, fotoelektrický efekt alebo iné kvantové javy. Hardvérové generátory môžeme označiť ako skutočne náhodné, nedeterministické. Softvérovým generátorom hovoríme pseudonáhodné, pretože sú deterministické. To znamená, že z rovnakej počítačovej premennej, nazývanej seed (slovensky semienko), algoritmus generuje vždy tú istú postupnosť pseudonáhodných čísiel. To môže byť v experimentoch výhoda, pretože Monte Carlo simuláciu s rovnakým generátorom pseudonáhodných čísiel spolu s rovnakým seedom na začiatku môžeme zopakovať a dostaneme rovnakú postupnosť čísiel a tým pádom aj rovnaký výsledok simulácie. Je relatívne málo známych generátorov a ešte menej "skutočne dobrých", ktoré je možné použiť v metódach Monte Carlo.

Generátor pseudonáhodných čísiel používa na vytvorenie nových čísiel nejakú funkciu predošlých, už vygenerovaných čísiel uložených v pamäti počítača,

$$n_i = f(n_{i-1}, n_{i-2}, \dots, n_{i-j}),$$

kde n_i je nové generované číslo, n_{i-m} je už v minulosti vygenerované číslo uložené v pamäti. Pri spustení algoritmu je číslo n_{i-m} nahradené seedom. Pokiaľ je čísiel j viac ako jedno, $j > 1$, znamená to, že aj generátor musí byť inicializovaný viacerými seedmi. Typický počet uložených čísiel j býva rádovo nula až 1000.

Požiadavky na generátor pseudonáhodných čísiel sú:

- čo najdlhšia perióda T , t.j. počet čísiel, po ktorých celý rad začne sám seba opakovať:

$$T : n_i = n_{i+T},$$

- rovnomerné rozloženie čísiel n_i ,
- medzi členmi súvislej postupnosti vybranej z radu $(n_{i-j}, n_{i-j-1}, \dots, n_{i-j-m})$, ani medzi $(f(n_{i-j}), f(n_{i-j-1}), \dots, f(n_{i-j-m}))$, kde f je nejaká funkcia, nie je žiadna zrejma súvisť, korelácia,
- výpočet je dostatočne rýchly.

Horšie alebo lepšie "skrytá" korelácia medzi členmi súvislej postupnosti vybranej z radu (tretí požiadavok) samozrejme je, z definície generátoru pseudonáhodných čísiel. To, či generátor spĺňa požiadavku o korelácií sa dá overiť testami o štatistickej náhodnosti, napr. Diehard testom [9] alebo TestU01 testom [10].

Nutno dodať, že rôzne generátory nájdu uplatnenie v rôznych odvetviach informatiky. Generátory vhodné k použitiu v simuláciách Monte Carlo nemusia byť vhodné v kryptografii a naopak. Napr. umelej inteligencii použitej v počítačových hrách, ktorá počíta aj s náhodným elementom, postačuje takmer akýkoľvek generátor.

V generátoroch pseudonáhodných čísiel sa využívajú hlavne kombinácie týchto bitových, logických a matematických operácií:

- mod - zvyšok po celočíselnom delení,
- xor - logický exkluzívny or,
- << - bitový posun doľava,
- >> - bitový posun doprava,
- & - bitový and,
- temperovanie - jemná úprava medzivýsledkov,
- skipping - vynechávanie čísiel, ktoré spôsobujú koreláciu.

3.2.1 Lineárne kongruenčné generátory

Princíp lineárnych kongruenčných generátorov (anglicky linear congruential generators, skratka LCG) navrhol v roku 1948 Derrick Henry Lehmer [5].

Nové členy postupnosti sa dopočítajú podľa rovnice

$$n_i = (an_{i-1} + b) \bmod m \Rightarrow n_i = f(n_{i-1}), \quad (8)$$

kde a , b , m sú zvolené prirodzené konštanty, seed označíme n_0 . Lineárne kongruenčné generátory nie sú k použitiu v metódach Monte Carlo vhodné, uvádzam ich preto, lebo sú značne rozšírené a tvoria základ lepších, k použitiu v metódach Monte Carlo vhodných generátorov.

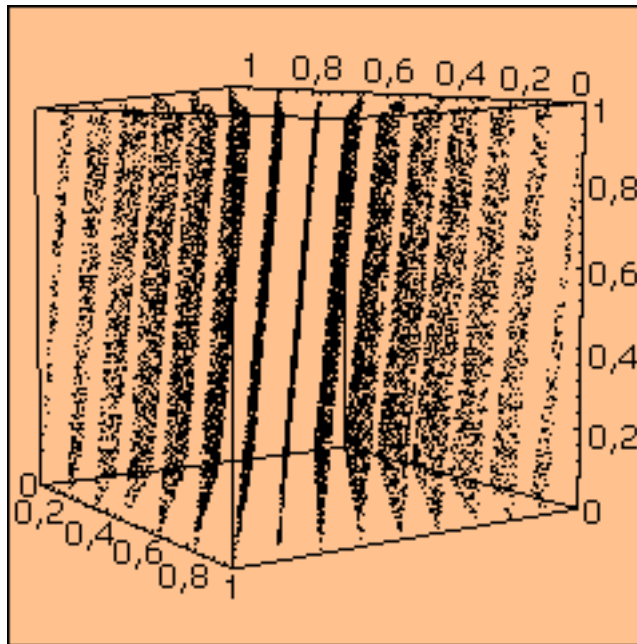
Do skupiny LCG patria napr. tieto generátory:

- ANSI C, alebo aj BSD random generátor (`gsl_rng_rand` (deklarácia v knižnici GSL [6])), kde $a = 1103515245$, $b = 12345$, $m = 2^{31}$, $n_0 = 12345$;
používal sa v unixových systémoch, v prekladačoch ANSI C, ako `rand()` funkcia, neskôr nahradený DRAND48.
- DRAND48, alebo aj Unix `rand48` generátor (`gsl_rng_rand48`), kde
 $a = 25214903917$, $b = 11$, $m = 2^{48}$, $n_0 = 0$;
unixový generátor, funkcia ANSI `drand48()`.
- CRAY (`gsl_rng_ranf`), kde
 $a = 44485709377909$, $b = 0$, $m = 2^{48}$, $n_0 = 1$;
používal sa v kompilátoroch PASCALu na strojoch CRAY.
- množstvo ďalších, čitateľa odkážem na literatúru [5] a [6].

Jednotlivé LCG sú equidistribuované maximálne v piatich rozmeroch. Čo to znamená, vidno z obrázku 3.3. LCG je možné vylepšiť na kombinované LCG.

3.2.1.1 IBM RANDU

Keď umiestňujeme atómy do trojrozmerného priestoru podľa pravdepodobností, ktoré sú generované LCG s $a = 2^{16} + 3$, $b = 0$, $m = 2^{31}$, $n_0 = 1$, stane sa to, že trojice n_i , n_{i-1} , n_{i-2} sú korelované podľa vzorca $n_i = 6n_{i-1} - 9n_{i-2}$. Keď potom umiestňujeme body s vygenerovanými súradnicami do trojrozmernej kocky s dĺžkou hrany jedna, body ležia v pätnástich rovinách [3], [5].



Obrázok 3.3: Distribúcia bodov podľa IBM RANDU v priestore [5].

Jedná sa o algoritmus IBM RANDU (`gsl_rng_randu`) vymyslený v 60tych rokoch, konštanty boli vybrané pre ich jednoduchú binárnu reprezentáciu. Rozšírenie tohto generátora spôsobilo, že niektoré výsledky simulácií zo 70tych rokov, v ktorých bol IBM RANDU použitý, boli chybné. IBM RANDU sa stal učebnicovým príkladom zlého generátora, s veľmi krátkou periódou (2^{29}) a výraznou koreláciou. Informatik Donald Ervin Knuth ho v roku 1981 nazval "naozaj strašný".

"...stačí spomenúť iba meno RANDU a žalúdok a oči programátora sa zhrozia!" Donald E. Knuth, 1998 [3], [4], [5].

Je dôležité si uvedomiť, že nemusíme vedieť a často ani nemáme možnosť dozvedieť sa čokoľvek bližšieho o generátoroch pseudonáhodných čísiel štandardne naimplementovaných v používaných prekladačoch. V žiadnom prípade by sme sa nemali na zabudovaný generátor spoliehať, pokiaľ o ňom nič nevieme. V horšom prípade to totiž môže byť a často aj býva LCG. LCG sa v bežných prekladačoch často inicializuje seedom podľa aktuálneho systémového času v počítači. Tento prístup nerieši problém s equidistribúciou. Navyše takto, podľa systémového času inicializovaného generátora, dostaneme rôzne postupnosti čísiel a experiment nemusíme zopakovať s rovnakým výsledkom. Preto je vhodné si zistiť, aký generátor používa kompilátor, ktorým chceme kód preložiť. V prípade nevyhovujúceho alebo neznámeho zabudovaného generátora je nutné použiť iný, napr. z knižnice GSL.

3.2.2 Kombinované lineárne kongruenčné generátory

Kombinovaný LCG je zložený z dvoch alebo viacerých LCG. Pre kombinovaný LCG, zložený z dvoch LCG, sú rovnice takéto:

$$\begin{aligned}n_i &= (x_i - y_i) \bmod(\max(m_x, m_y)), \\x_i &= (a_x x_{i-1} + b_x) \bmod m_x, \\y_i &= (a_y y_{i-1} + b_y) \bmod m_y.\end{aligned}\tag{9}$$

Príkladom môžu byť L'Ecuyerove LCG [príklad v kapitole 3.2.3].

3.2.3 Spomalené Fibonacciho generátory

Rozdiel Spomalených Fibonacciho generátorov (anglicky Lagged Fibonacci generator, skratka LFG) oproti LCG je v tom, že LFG k výpočtu ďalšieho pseudonáhodného čísla n_i nevyužívajú iba jedno predošlé číslo n_{i-1} , ale aj ďalšie predošlé, už vygenerované čísla n_{i-j} .

$n_i = f(n_{i-1}, n_{i-2}, \dots, n_{i-j})$; kde f je niektorá binárna operácia, sčítanie (anglicky Additive LFG, skratka ALFG), odčítanie, násobenie (anglicky Multiplicative LFG, skratka MLFG) alebo xor (Všeobecný posuvný register s návratom; anglicky Generalised Feedback Shift Register, skratka GFSR [kapitola 3.2.3.1]).

- L'Ecuyer, Blouin a Coutreov LCG (gsl_rng_mrg)

$$n_i = (a_1 n_{i-1} + a_5 n_{i-5}) \bmod m,\tag{10}$$

kde $a_1 = 107374182$, $a_5 = 104480$, $m = 2^{31} - 1$.

- Kombinovaný multiplikatívny L'Ecuyerov LCG (gsl_rng_cmrg)

$$\begin{aligned}n_i &= (x_i - y_i) \bmod m_x, \\x_i &= (a_{1x} x_{i-1} + a_{2x} x_{i-2} + a_{3x} x_{i-3}) \bmod m_x, \\y_i &= (a_{1y} y_{i-1} + a_{2y} y_{i-2} + a_{3y} y_{i-3}) \bmod m_y,\end{aligned}\tag{11}$$

kde

$$a_{1x} = 0, a_{2x} = 63308, a_{3x} = -183326, m_x = 2^{31} - 1,$$

$$a_{1y} = 86098, a_{2y} = 0, a_{3y} = -539608, m_y = 2145483479.$$

- RANLUX (gsl_rng_ranlx, gsl_rng_ranlux)

tento algoritmus vychádza z LCG RCARRY, ktorý dopĺňa o Fibonacciho členy a vynecháva čísla, ktoré spôsobujú koreláciu.

Všetky tri generátory spomenuté v tejto kapitole sú vhodné na použitie v metódach Monte Carlo.

3.2.3.1 Generátory so všeobecným posuvným registrom s návratom

- Mersenne Twister algoritmus (gsl_rng_mt19937)

Tento algoritmus bol navrhnutý s ohľadom na jeho budúce využitie v metódach Monte Carlo, prvá verzia je z roku 1997, odvtedy vzniklo množstvo ďalších vylepšených verzií a verzií orientovaných na rýchlosť, paralelné systémy apod. Autormi sú Makoto Matsumoto a Takuji Nishimura. Je to rýchly generátor s periódou $T = 2^{19937} - 1$. Tento generátor je v 32-bitovej verzii equidistribuovaný v 623 rozmeroch.

Pseudokód môže čitateľ nájsť v [11].

Zdrojový kód rôznych verzií v rôznych programovacích jazykoch zase na domovskej stránke projektu [7].

- Gfsr4 (gsl_rng_gfsr4)

$$n_i = n_{i-A} \text{ XOR } n_{i-B} \text{ XOR } n_{i-C} \text{ XOR } n_{i-D}, \quad (12)$$

kde $A = 471, B = 1586, C = 6988, D = 9689$.

- Kombinovaný Tauswortheov LCG od L'Ecuyra (gsl_rng_taus)

$$\begin{aligned} n_i &= x_i \text{ XOR } y_i \text{ XOR } z_i, \\ x_i &= ((x_{i-1} \& 4294967294) \ll 12) \text{ XOR } (((x_{i-1} \ll 13) \text{ XOR } (x_{i-1})) \gg 19), \\ y_i &= ((y_{i-1} \& 4294967288) \ll 4) \text{ XOR } (((y_{i-1} \ll 2) \text{ XOR } (y_{i-1})) \gg 25), \\ z_i &= ((z_{i-1} \& 4294967280) \ll 17) \text{ XOR } (((z_{i-1} \ll 3) \text{ XOR } (z_{i-1})) \gg 11). \end{aligned} \quad (13)$$

Kombinovaný Tauswortheov LCG je použitý aj v mojom programe. Má kratšiu periódu ako Mersenne Twister, ale je rýchlejší [6].

3.3 Monte Carlo metódy pre mriežkové modely

Nasadenie metód Monte Carlo v mriežkových modeloch v tejto práci slúži k nájdeniu rovnovážneho stavu pri danej teplote a spočítaniu voľnej energie z nich. Na príklade Isingovho modelu mriežky feromagnetika to vyzerá takto. Analýza všetkých konfigurácií mriežky (stavov, v ktorých sa môže mriežka nachádzať) je zložitou exponenciálny problém. Ak môže spin nadobúdať iba dve hodnoty (v prípade Isingovho modelu), pri počte n atómov je počet konfigurácií rovný 2^n . V prípade trojrozmerného modelu v tvare kocky s hranou desať atómov je počet konfigurácií 2^{1000} a to už je časovo neriešiteľný problém. Časovú zložitou algoritmu nezmení ani náhodný výber konfigurácií.

Spôsob, akým nájdeme rovnováhu je nasledovný. Voľná energia systému sa dá spočítať z partičnej sumy

$$Z = \sum_i e^{-\beta H_N(K_i)}, \quad (14)$$

kde H_N je hamiltonián pre N spinov na mriežke, K_i je konfigurácia systému a $\beta = \frac{1}{kT}$,

kde $k = 1,38 \times 10^{-23}$ je Boltzmannova konštanta a T je termodynamická teplota [2].

K správnej konfigurácii sa doberieme postupnosťou iných konfigurácií tak, aby postupnosť pravdepodobností P_i konfigurácií K_i konvergovala ku kanonickému rovnovážnemu rozdeleniu: $P_i \rightarrow P_i^{eq}$. Toto sa dá zariadiť pomocou Markovovho procesu (reťazca), ktorý je popísaný pravdepodobnosťou prechodu $w_{i \rightarrow j}$ z konfigurácie K_i do konfigurácie K_j . Aby postupnosť konvergovala, potrebujeme, aby bola splnená tzv. balančná podmienka, ktorá hovorí, že

$$P_i^{eq} w_{i \rightarrow j} = P_j^{eq} w_{j \rightarrow i}. \quad (15)$$

To znamená, že pravdepodobnosť zmeny z konfigurácie i na konfiguráciu j sa musí rovnať pravdepodobnosti zmeny z konfigurácie j na konfiguráciu i .

Z kanonického rozdelenia sa dá balančná podmienka prepísať do tvaru [2]

$$\frac{w_{i \rightarrow j}}{w_{j \rightarrow i}} = e^{-\beta(H_N(K_j) - H_N(K_i))}. \quad (16)$$

Ďalšou požiadavkou je splnenie ergodickej podmienky, ktorá hovorí, že k akejkoľvek konfigurácii K_i sa môžeme prepracovať reťazcom konfigurácií z akejkoľvek konfigurácie K_j , t.j. prechod medzi výberom akýkoľvek dvoch konfigurácií z množiny konfigurácií je možný, aj keď pravdepodobnosť toho, že sa to stane, môže byť veľmi malá.

Markovov reťazec je potom postupnosť konfigurácií, kde začíname konfiguráciou K_i , z konfigurácie stavu K_i generujeme novú konfiguráciu K_{i+1} , z K_{i+1} zase K_{i+2} atď. tak, že postupnosť konfigurácií má Boltzmannove rozdelenie. Reťazec smeruje do rovnováhy.

3.3.1 Metropolisov algoritmus

Metropolisov algoritmus je jeden z najrozšírenejších algoritmov patriacich do metód Monte Carlo. Použil ho v roku 1953 Nicolas Metropolis na simuláciu modelu plynu [1]. Jeho zovšeobecnenie z Boltzmannovho rozdelenia spravil v roku 1970 W. K. Hastings. Preto sa môžeme stretnúť aj s názvom Metropolis-Hastingsov algoritmus.

Postup Metropolisovho algoritmu pre Isingov model (predpis pre jeho interakčnú energiu je v kapitole 2.2) je rovnaký ako v kapitole 3.3. Algoritmus transformuje aktuálnu konfiguráciu K_i na konfiguráciu K_j . Ak konfigurácia K_j spĺňa balančnú podmienku, algoritmus prejde do stavu K_j . Ak K_j balančnú podmienku nespĺňa, algoritmus zachová pôvodnú konfiguráciu K_i . Tento proces sa opakuje n -krát, alebo sa prestane opakovať pokiaľ je počet prijatých konfigurácií zo všetkých generovaných konfigurácií menší ako nejaké určené číslo. Generovanie nových stavov môže prebiehať tak, že nová konfigurácia je od starej odlišná iba v jednom spine, ktorý je vyberaný buď náhodne alebo systematicky. Alebo môžeme otáčať naraz celou skupinou spinov. V prípade jedného spinu je maximálny možný

rozdiel interakčnej energie $8J$. V prípade Pottsovoho a ďalších modelov môžeme generovať otočenie spinu náhodne na celom intervale možných hodnôt, alebo iba na určitom okolí okolo pôvodnej hodnoty. Balančná podmienka je v prípade Metropolisovho algoritmu splnená týmto predpisom pravdepodobnosti prechodov:

$$\begin{aligned} w_{i \rightarrow j} &= e^{-\beta(H_N(K_j) - H_N(K_i))}; & (H_N(K_j) - H_N(K_i)) > 0, \\ w_{i \rightarrow j} &= 1; & (H_N(K_j) - H_N(K_i)) \leq 0. \end{aligned} \quad (17)$$

Tento predpis hovorí, že konfigurácia s nižšou vnútornou energiou je prijatá vždy, konfigurácia s vyššou energiou je prijatá vtedy ak je $\gamma \leq e^{-\beta(H_N(K_j) - H_N(K_i))}$, kde $\gamma \in (0,1)$ je nejaké náhodné vygenerované číslo z intervalu $(0,1)$. V opačnom prípade zachováme pôvodnú konfiguráciu. Pravdepodobnosť prijatia konfigurácie s vyššou energiou exponenciálne klesá s klesajúcou teplotou.

Pri implementácii algoritmu na počítači väčšinou uvažujeme periodické okrajové podmienky, čím aspoň čiastočne zmenšíme závislosť na veľkosti mriežky [kapitola 2.1]. Výber počiatočnej konfigurácie by nemal mať vplyv na výsledok algoritmu, počiatočnú konfiguráciu môžeme zvoliť náhodne. Ďalším parametrom algoritmu bude počiatočná teplota. Pri vysokých teplotách sa bude člen $e^{-\beta(H_N(K_j) - H_N(K_i))}$ blížiť k jednotke, preto bude takmer každá konfigurácia, bez ohľadu na veľkosť vnútornej energie, prijatá. To znamená, že spiny sa budú neustále náhodne otáčať. Naopak, pri nízkych teplotách, budú všetky konfigurácie s vyššou vnútornou energiou odmietnuté. V prípade Isingovho modelu by mal algoritmus skončiť v jednej z dvoch konfigurácií s najnižšou energiou. Buď so všetkými spinmi orientovanými nahor, alebo všetkými spinmi orientovanými dole. Niekedy je vhodné teplotu v priebehu výpočtu meniť [kapitola 3.3.4].

V ustálenom stave môžeme na modeli zmerať (spočítať) fyzikálne veličiny. Energii E , magnetizáciu M , mernú tepelnú kapacitu C , magnetickú susceptibilitu χ . Energia sa spočíta ako súčet interakčnej energie všetkých dvojíc susedov, magnetizácia ako vektorový súčet všetkých magnetických spinov mriežky. Merná tepelná kapacita a magnetická susceptibilita ako derivácie energie a magnetizácie podľa teploty. Na mriežkových modeloch je však k výpočtu C a χ výhodnejšie namiesto derivácie počítat fluktuácie veličín [2].

Fluktuácia veličiny X , $\text{var } X$ sa rovná

$$\text{var } X = \langle X^2 \rangle - \langle X \rangle^2. \quad (18)$$

Medzi E , M , C a χ platia tieto vzťahy [2]:

$$\text{var } E = kT^2 C, \quad (19)$$

$$\text{var } M = \frac{kT}{\mu_0 V} \chi. \quad (20)$$

3.3.2 Heat-bath (teplotná kúpeľ)

Heat-bath je algoritmus veľmi podobný Metropolisovmu algoritmu. Líši sa v predpise pravdepodobnosti prechodov, kde

$$w_{i \rightarrow j} = \frac{e^{-\beta H_N(K_j)}}{\sum_k e^{-\beta H_N(K_k)}}. \quad (21)$$

V tejto metóde uvažujeme zmenu vždy iba jedného spinu a ignorujeme jeho pôvodnú hodnotu. Suma v menovateli je súčtom všetkých konfigurácií, ktoré môžu vzniknúť s novým spinom. Na príklade Isingovho modelu sú pravdepodobnosti toho, že vybraný menený spin bude smerovať hore (hodnota 1), alebo dole (hodnota -1):

$$P_1 = \frac{e^{-\beta E_1}}{e^{-\beta E_1} + e^{-\beta E_{-1}}}, \quad (22)$$

$$P_{-1} = \frac{e^{-\beta E_{-1}}}{e^{-\beta E_1} + e^{-\beta E_{-1}}}, \quad (23)$$

kde E_1 je energia stavu ak by spin smeroval po novom nahor a E_{-1} ak by smeroval nadol. Tento algoritmus je oproti Metropolisovmu algoritmu výhodné použiť pri vysokých hodnotách Q v Pottsovom modeli. Výpočet jedného kroku síce zaberie viac času, ale k rovnováhe sa blíži oveľa rýchlejšie [1], [kapitola 2.3].

3.3.3 Metóda s hyperbolickou tangentou

Metóda s hyperbolickou tangentou je opäť variáciou na predošlé algoritmy s iným predpisom pravdepodobnosti prechodov, ktorým sa balančná podmienka splní [2]:

$$w_{i \rightarrow j} = \frac{1}{2} \left[1 - \text{th} \frac{-\beta(H_N(K_j) - H_N(K_i))}{2} \right]. \quad (24)$$

3.3.4 Simulované ochladzovanie (simulated annealing)

Simulované ochladzovanie je algoritmus, v ktorom dochádza k postupnému znižovaniu teploty o nejaký krok, keď v každej teplote je zavolaný jeden zo spomínaných algoritmov (Metropolisov algoritmus, heat-bath alebo metóda s hyperbolickou tangentou). Novú teplotu môžeme dopočítavať napr. odčítavaním kroku od starej teploty, vtedy teplota klesá lineárne, alebo starú teplotu násobiť krokom menším ako jedna, vtedy klesá teplota exponenciálne.

Ak predpokladáme, že $T_k = \frac{T_0}{\log k}$, kde T_k je k -tá teplota a T_0 je dostatočne vysoká

počiatočná teplota, je možné nájsť optimálne riešenie úlohy. Problémom je, že nikdy nebudeme vedieť, či sme už optimálne riešenie našli, alebo sme zatiaľ iba v lokálnom extrém. Logaritmicke znižovanie teploty je navyše pomalé. Obvykle chceme pri každej

teplote zistiť hodnotu meraných veličín, v prípade feromagnetík to sú už spomínané veličiny E , M , C a χ . Simulované ochladzovanie ponúka elegantný spôsob na riešenie množstva iných fyzikálnych aj nefyzikálnych problémov, ktoré zdanlivo s ochladzovaním vo fyzikálnom zmysle slova nemajú nič spoločné.

Medzi problémy riešiteľné simulovaným ochladzovaním patria optimalizačné problémy, napr.:

- problém obchodného cestujúceho [kapitola 3.4.1],
- nájdenie základného stavu systému (napr. konfigurácie mriežkového modelu blízko absolútnej nuly, stavu s minimálnou energiou),
- nájdenie extrému komplikovanej funkcie,
- narhnutie do plochy čo najmenšieho plošného spoja zo známych veľkostí súčiastok, ich vývodov a prepojení a obmedzení v zapojení,
- problém navrhnutia rozvrhu, efektívne využitie miestností a času prednášajúcich a študentov [2].

3.3.4.1 Problém obchodného cestujúceho

Problém obchodného cestujúceho znie: Obchodný cestujúci má navštíviť všetky dané mestá tak, aby:

- cesta, ktorú prešiel, bola čo najkratšia,
- každé mesto navštívil práve raz,
- na konci sa vrátil do východzieho mesta.

Problém obchodného cestujúceho je NP-úplný grafový problém s faktoriálnou časovou zložitou. Jedným zo spôsobov, ako ho riešiť, je práve simulované ochladzovanie. Pseudokód riešenia vyzerá takto:

```
Algoritmus:

int pocet_opakovani = 1000;    //počet opakovaní generovania zmenenej cesty
                                //v jednej teplote
double teplota1;
double teplota2;
double zmena_teploty;
cesta stara_cesta;            //premenná s poradím miest
cesta nova_cesta;             //premenná s poradím miest
double gsl_rng_uniform();     //funkcia vracajúca číslo od 0 do 1
cesta generuj_nahodnu_cestu(); //funkcia, ktorá vygeneruje náhodnú
                                //kružnicu cez všetky body grafu
cesta generuj_zmenenu_cestu(stara_cesta);
                                //funkcia, ktorá nejakým spôsobom,
                                //napríklad zamenou poradia dvoch bodov v
                                //starej ceste vygeneruje novú cestu
double dlzka(cesta);          //spocita dlzku cesty
```

```

stara_cesta = generuj_nahodnu_cestu;
while (teplota2 < teplota1) {
    for(int i=0; i<pocet_opakovani; i++) {
        nova_cesta = generuj_zmenenu_cestu(stara_cesta);
        if (gsl_rng_uniform() < exp((dlzka(stara_cesta) -
            dlzka(nova_cesta)) / teplota1)) {
            stara_cesta = nova_cesta;
        }
    }
    teplota1 = teplota1 - zmena_teploty;
}

```

Tým, že akceptujeme aj stavy s vyššou energiou (dlhšou cestou), zabezpečíme to, aby algoritmus nezostal "zaseknutý" v lokálnom minime. Po skončení algoritmu bude najlepšia nájdená cesta uložená (nemusí byť najlepšou možnou, ale bude "dostatočne dobrá") v premennej `nova_cesta`.

3.3.5 Iné metódy

Existuje množstvo iných, alebo zľahka modifikovaných už spomenutých algoritmov, ktoré sa používajú v štatistickej fyzike. Napríklad Wolfov algoritmus, ktorý pri modifikovaní spinov v Isingovom modeli otáča spiny celej domény. Alebo Kawasakiho model, ktorý zachováva celkovú magnetizáciu, pri svojom behu v jednom kroku vymení dva opačne orientované spiny, apod.

4 Program Monte Carlo simulátor

4.1 Objekt implementácie

Program používa metódy Monte Carlo k simulácii fázových prechodov feromagnetických látok. Štruktúru systému aproximuje použitím mriežkového modelu s periodickými okrajovými podmienkami. Simulovanými fázovými prechodmi sú prechody medzi neusporiadanými vysokoteplotnými fázami látok a nízkoteplotnými usporiadanými fázami. Pri fázovom prechode z teplej do studenej fázy sa výrazne znižuje hodnota celkovej energie E systému a naopak narastá magnetizácia H . Merná tepelná kapacita C a magnetická susceptibilita χ sa správajú ako derivácie energie a magnetizácie. Priebeh týchto veličín program pri simulácii počíta a vynáša do grafu. Program implementuje šesť základných druhov modelov spomenutých v kapitolách 2.2 až 2.6:

Dvojrozmerné modely:

- Isingov model,
- Pottsov model,
- Z_Q model,

- Heisenbergov model.
- Trojrozmerné modely:
- Isingov model,
- Heisenbergov model.

Maximálna veľkosť hrany mriežky modelu je 10^3 , preto je maximálny počet častíc 10^6 pre dvojrozmerný model a 10^9 pre trojrozmerný. Vybranými naimplementovanými metódami sú Metropolisov algoritmus [kapitola 3.3.1] a simulované ochladzovanie [kapitola 3.3.4].

Atómy su zobrazené ako guľičky. Na zobrazenie smerov spinov som použil šípky (inde sa môžeme stretnúť aj s reprezentáciou pomocou farieb, čísiel, znakov + alebo – apod.)

4.2 Popis implementácie

Ako vývojové prostredie som zvolil Microsoft Visual Studio 2005, programovacím jazykom bol jazyk C++. Na návrh a preklad užívateľského rozhrania som použil Trolltech Qt Designer a open source knižnice Qt vo verzii 4.3.4 [12]. Na generovanie náhodných čísel som použil GNU Scientific Library (GSL) vo verzii 1.11 [8], [kapitola 3.2], v ktorej už je naimplementovaná široká škála generátorov pseudonáhodných čísel vhodných na použitie v simuláciách Monte Carlo. Grafický výstup programu, t.j. samotný model a grafy sú vykresľované knižnicou OpenGL a jej nadstavbou GLU. Program bol skompilovaný a inštalátor vytvorený pomocou nástrojov Visual Studia, ale mal by sa dať bez zmien, prípadne iba s malými zmenami, ktorých si nie som vedomý, skompilovať aj iným C++ prekladačom a v inom, napr. unixovom, prostredí. Pri písaní kódu boli použité manuály a tutoriály z internetového magazínu root [13] a z obsiahlej nápovedy knižnice Qt s názvom Qt Assistant [12]. Použil som štyri voľne šíriteľné ikony z prostredia GNOME [14]. V programe nie sú takmer žiadne komentáre, myslím si, že všetky funkcie, premenné a konštanty majú zrozumiteľný názov, z ktorého je ich použitie jasné. Aplikácia obsahuje aj jednoduché menu a toolbar (panel nástrojov), z ktorého možno schovať jednotlivé časti rozhrania, ukončiť program a zobrazit' okno s informáciami o programe.

4.2.1 Popis tried a ich grafická reprezentácia.

Všetky hlavné triedy zdrojového kódu majú vlastný hlavičkový súbor (.h) a súbor so zdrojovým kódom (.cpp).

Reprezentáciu mriežkových modelov a výpočty nad nimi implementuje trieda:

- Cmodel (model.h)

Vygenerované užívateľské rozhranie nastavuje trieda:

- Diplomka (diplomka.h)

Vykresľovanie modelu a grafov zabezpečujú triedy:

- CmodelWidget (GLkreslitko.h)
- CgrafWidget (graf.h)
- Komunikáciu medzi užívateľským rozhraním a modelom zabezpečujú triedy:
- CmodelGenerator (CmodelGenerator.h)
- CmodelSolver (CmodelSolver.h)

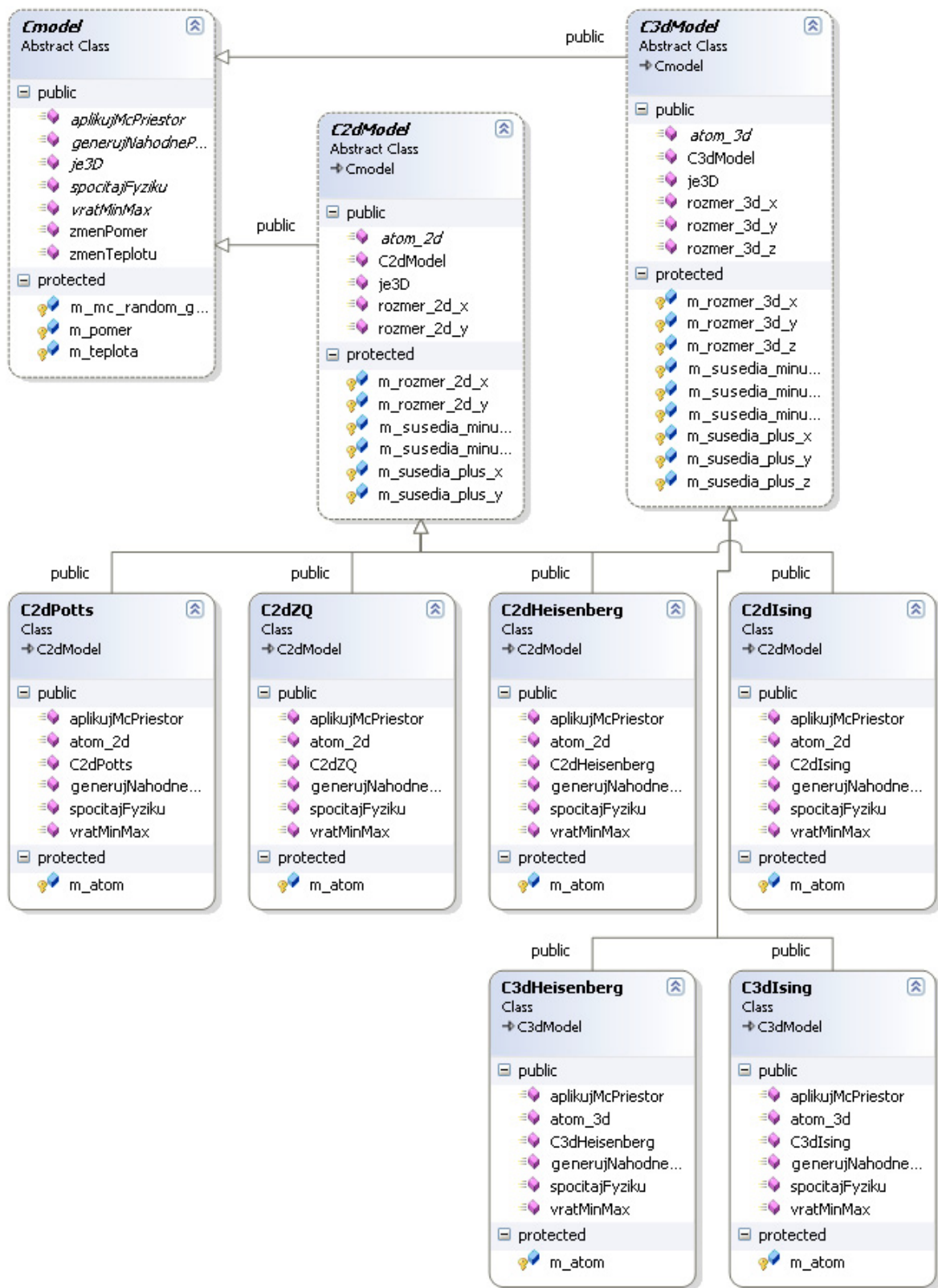
4.2.1.1 Trieda Cmodel

Je ústrednou triedou celého programu. Táto trieda je rodičovská pre triedy C2dModel a C3dModel, ktoré znázorňujú dvojrozmerný a trojrozmerný model. Z nich sú odvodené triedy jednotlivých modelov:

- C2dIsing,
- C2dPotts,
- C2dZQ,
- C2dHeisenberg,
- C3dIsing,
- C3dHeisenberg.

Jednotlivé triedy obsahujú alebo dedia okrem iného tieto premenné a funkcie:

- rozmery modelu, maximálny rozmer v jednom smere je 10^3 ,
- dvojrozmerné alebo trojrozmerné pole na uloženie mriežkového modelu (m_atom). Pole je statické, pri pokuse použiť dynamické pole klesla rýchlosť výpočtov na polovicu,
- jednorozmerné pole susedov, ktoré je použité pri výpočte periodických okrajových podmienok a súradníc susedov jednotlivých atómov. Je to jednoduchá a názorná reprezentácia. Pri pokuse dopočítavať súradnice susedných atómov pri výpočte sa rýchlosť programu nezmenila,
- vlastný random generátor z knižnice GSL, konkrétne kombinovaný Tauswortheov lineárny kongruenčný generátor od L'Ecuyra (gsl_rng_taus),
- počet rôznych orientácií spinov v Pottsovom a Z_Q modeli,
- aktuálnu teplotu prostredia,
- funkciu na vygenerovanie náhodných orientácií všetkých spinov,
- funkciu na spočítanie hodnôt fyzikálnych veličín, energie, magnetizácie, mernej tepelnej kapacity a magnetickej susceptability,
- funkciu vracajúcu maximálne a minimálne dosiahnuteľné hodnoty týchto veličín,
- funkciu aplikujúcu jeden krok Metropolisovho algoritmu na každý atóm,
- funkcie na zmenu súkromných premenných triedy, t.j. na zmenu teploty a zmenu počtu spinových orientácií.



Obrázok 4.1: Grafická reprezentácia triedy Cmodel a jej dedičných tried.

4.2.1.2 Triedy užívateľského rozhrania

Práca na rozhraní prebiehala v programe Trolltech Qt Designer. Ten uloží vytvorené užívateľské rozhranie do súboru vo formáte XML, ktoré je zásuvným modulom vo Visual Studiu preložené do C++ kódu vo forme hlavičkového súboru (ui_diplomka.h). Všetky elementy knižnice Qt majú predponu "Q". Sú to akési šablóny prvkov nielen užívateľského rozhrania, ktoré možno prispôbovať konkrétnym požiadavkom programátora. Trieda diplomka rozširuje užívateľské rozhranie o metódy, ktoré preposielajú vstupy [kapitola 4.2.2] zadané užívateľom k spracovaniu triede CmodelGenerator. Okrem toho definuje východzie premenné (vstupy) programu po jeho spustení, implementuje akcie pri ukončení programu (zastavenie výpočtu) a okno so stručnými informáciami o programe, tzv. "About Application".

4.2.1.3 Triedy vykresľovacie

Program vykresľuje tri okná s grafickým výstupom:

- okno s mriežkovým modelom,
- okno s grafom energie E a jej deriváciou, mernou tepelnou kapacitou C ,
- okno s grafom magnetizácie M a jej deriváciou, magnetickou susceptibilitou χ .

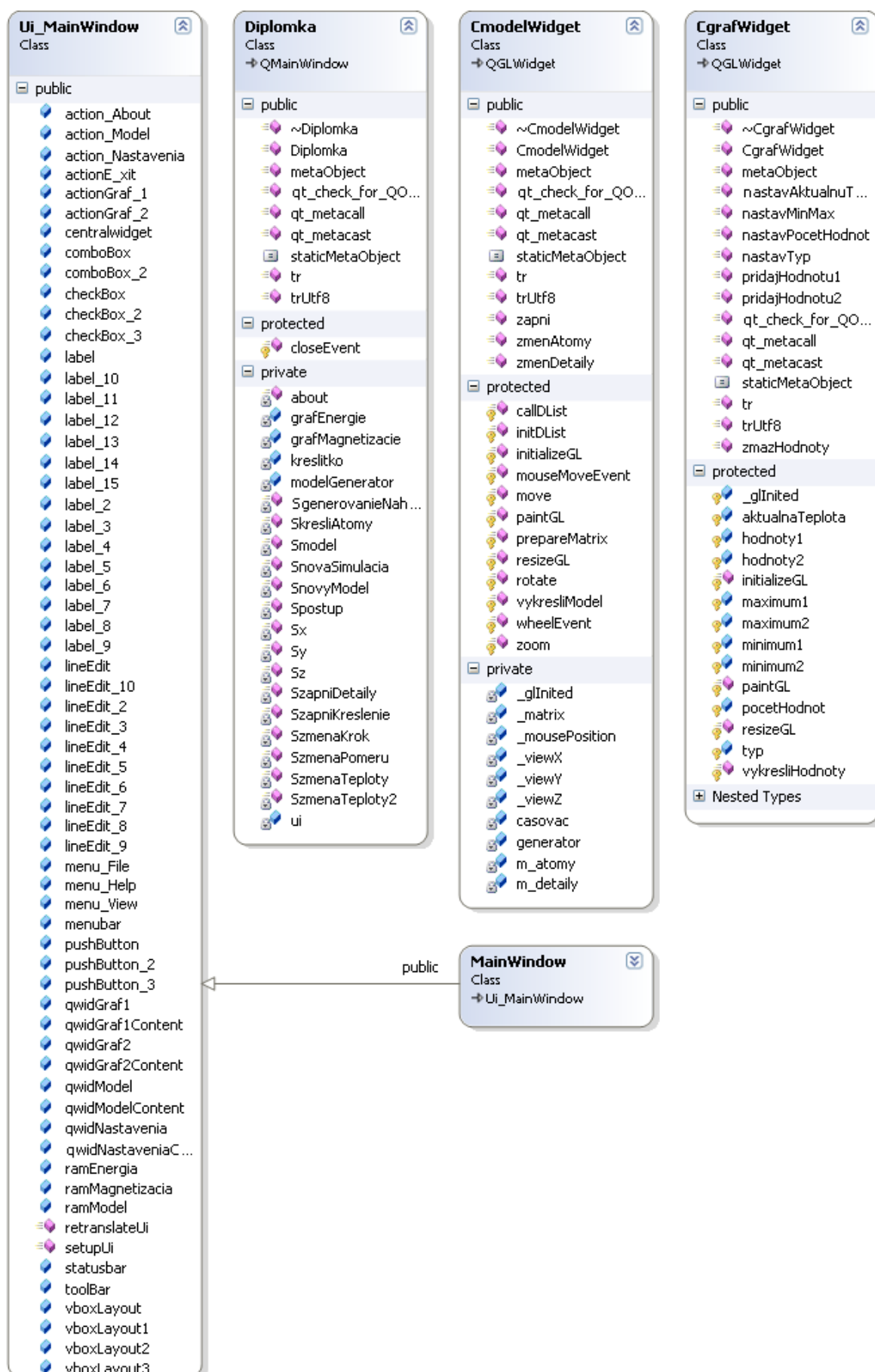
O ich vykresľovanie sa starajú triedy CmodelWidget a CgrafWidget, ktoré dedia vlastnosti z objektu Qt knižnice s názvom QGLWidget. V okne nastavení programu je ponuka troch možností spôsobu vykresľovania mriežkového modelu:

- zapnúť alebo vypnúť samotné vykresľovanie. V prípade rozmerovo veľkých modelov vypnuté vykresľovanie značne zrýchli výpočty a tým aj simuláciu. Pri zapnutom vykresľovaní je model prekreslený každú jednu desatinu sekundy podľa časovača (QTimer),
- zapnúť alebo vypnúť detaily vykresľovania. Pri zapnutých detailoch sú k znázorneniu atómov a spinov použité guľičky a šípky zložené zo štvornásobne viac bodov než pri vypnutých, nižších detailoch. Vypnutie detailov zrýchli výpočet a simuláciu,
- zapnúť alebo vypnúť vykresľovanie atómov. Pri vypnutom zostanú vykresľované iba spiny.

S modelom je možné otáčať. Je možné ho približovať, oddalovať a posúvať.

V grafoch sú znázornené energia a magnetizácia modrou farbou, merná tepelná kapacita a magnetická susceptibilita červenou farbou. Grafy sú prekresľované iba pri simulácii s každou zmenou teploty. Veličiny sa vykresľujú v smere klesajúcej teploty T , t.j. v smere zprava doľava. Grafy nemajú absolútne hodnoty veličín, ale iba pomerové. Je to preto, že vo výpočtoch neuvažujem Boltzmannovu konštantu k a konkrétnu hodnotu konštanty J z predpisu modelov. V smere zvislej osi sú energia a magnetizácia preškáľované v rozmedzí

ich najväčšich a najmenšich možných hodnôt pre jednotlivé modely. Merná tepelná kapacita a magnetická susceptibilita sú preškálované od nuly po ich najväčšiu, v priebehu simulácie nameranú hodnotu. V grafoch je pre prehľad o aktuálnom stave simulácie zobrazená okamžitá hodnota teploty T .



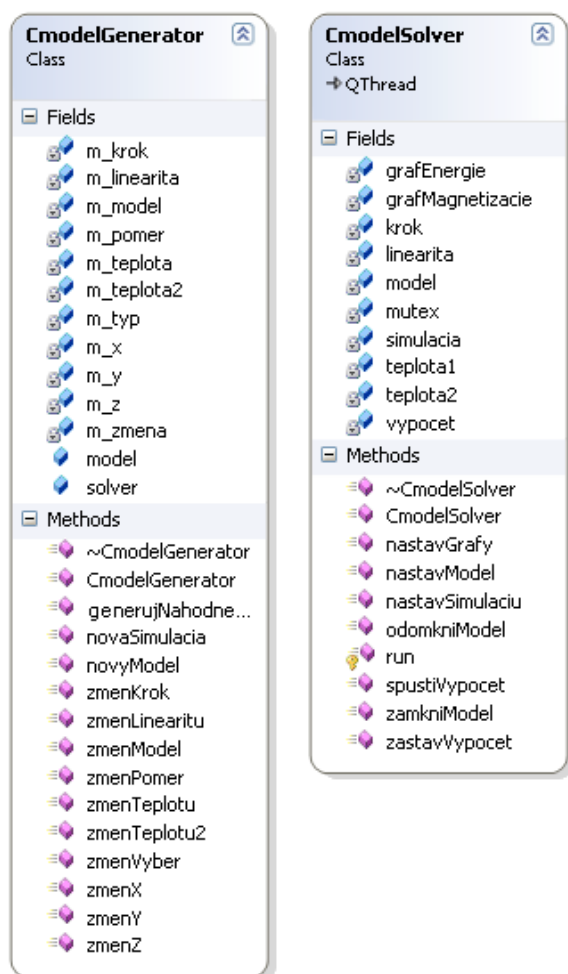
Obrázok 4.2: Reprezentácia tried Diplomka, CmodelWidget, CgrafWidget a užívateľského rozhrania.

4.2.1.4 Triedy CmodelGenerator a CmodelSolver

V prípade zmeny niektorej hodnoty alebo kliknutí na tlačítko (dve tlačítka OK a tlačítko Generuj) v okne nastavení sa aktivuje signál v triede Diplomka a zavolá sa funkcia triedy CmodelGenerator. Táto trieda obsahuje aktuálne hodnoty všetkých vstupov a rozhoduje, ako sa zachovať v prípade kliknutia na jedno z tlačítiek. Pri stlačení tlačítka sa buď:

- zmení niektorá z vlastností aktuálneho modelu,
- zmení celý model,
- zmení celý model a spustí sa simulácia.

Simuláciu obsluhuje trieda CmodelSolver. Simuláciu spustí na základe vstupných premenných a v cykle násobne aplikuje na model Metropolisov algoritmus s klesajúcou teplotou (simulované ochladzovanie). Výpočet v triede CmodelSolver prebieha v novom vlákne (mutex). Je to výhodné v tom, že výpočet môžeme kedykoľvek pozastaviť, zmeniť jeho parametre, alebo vykresliť aktuálny stav modelu. Z toho vyplýva, že metódy na zastavenie výpočtu, uzamknutie modelu a naopak odomknutie modelu a spustenie výpočtu sú volané pomerne často.



Obrázok 4.3: Reprezentácia tried CmodelGenerator a CmodelSolver.

4.2.2 Vstupy programu

Vstupmi programu a tým zároveň parametrami mriežkového modelu a MonteCarlo simulácie sú:

- typ modelu (šesť už niekoľkokrát spomínaných základných modelov),
- rozmer v smere osi x (prirodzené číslo od jedna do 1000),
- rozmer v smere osi y (prirodzené číslo od jedna do 1000),
- rozmer v smere osi z (prirodzené číslo od jedna do 1000),
- počet orientácií spinov v Pottsovom a Z_Q modeli, t.j. uhol otočenia spinu je násobok pomeru 2π ku zadanému počtu,
- teplota T_1 v tvare $a \times 10^b$, kde $a > 0$ a b je reálne číslo v rozsahu typu C++ double
- spustenie Metropolisovho algoritmu,
- generovanie náhodného poľa modelu,
- teplota T_2 v tvare $a \times 10^b$, kde $a > 0$ a b je reálne číslo v rozsahu typu C++ double, pre použitie v simulovanom ochladzovaní,
- krok zmeny teplota T_2 v tvare $a \times 10^b$, kde $a > 0$ a b je reálne číslo v rozsahu typu C++ double, pre použitie v simulovanom ochladzovaní,
- výber spôsobu poklesu teploty v jednom kroku, lineárne alebo exponenciálne,
- spustenie simulovaného ochladzovania,
- zapnutie vykresľovania,
- zapnutie vykresľovania detailov,
- zapnutie vykresľovania atómov.

4.2.3 Výstupy programu

Výstupmi programu sú:

- trojrozmerný grafický model skúmaného systému,
- dvojrozmerný graf závislosti energie E a mernej tepelnej kapacity C na teplote,
- dvojrozmerný graf magnetizácie M a magnetickej susceptibility χ na teplote.

4.2.4 Scenár použitia

Použitie programu prebieha podľa nasledujúceho scenára.

Po spustení programu je predvoleným systémom dvojrozmerný Isingov model s rozmermi 32×32 . Teplota je 0,01 K.

Bez zmeny parametrov môže užívateľ stlačiť horné tlačítko *OK*, vtedy sa na predvolený Isingov model aplikuje Metropolisov algoritmus.

Bez zmeny parametrov môže užívateľ stlačiť tlačítko *Generuj*, vtedy sa na modeli vygeneruje nová náhodná orientácia spinov.

Bez zmeny parametrov môže užívateľ stlačiť dolné tlačítko *OK*, vtedy sa na modeli spustí simulované ochladzovanie. Program porovná teplotu T_1 a teplotu T_2 a simuláciu začne od väčšej z nich (ak je $T_1 < T_2$, tak ich jednoducho vymení). Metropolisov algoritmus je cyklicky aplikovaný na model. Teplota klesá od T_1 exponenciálne alebo lineárne, kým nie je nižšia ako T_2 . Na konci simulácie je Metropolisov algoritmus aplikovaná aj na T_2 . Pri simulovanom ochladzovaní je spustené vykresľovanie grafov.

So zmenou všetkých jednotlivých parametrov okrem teploty sa po kliknutí na horné tlačítko *OK* vygeneruje vždy nový model.

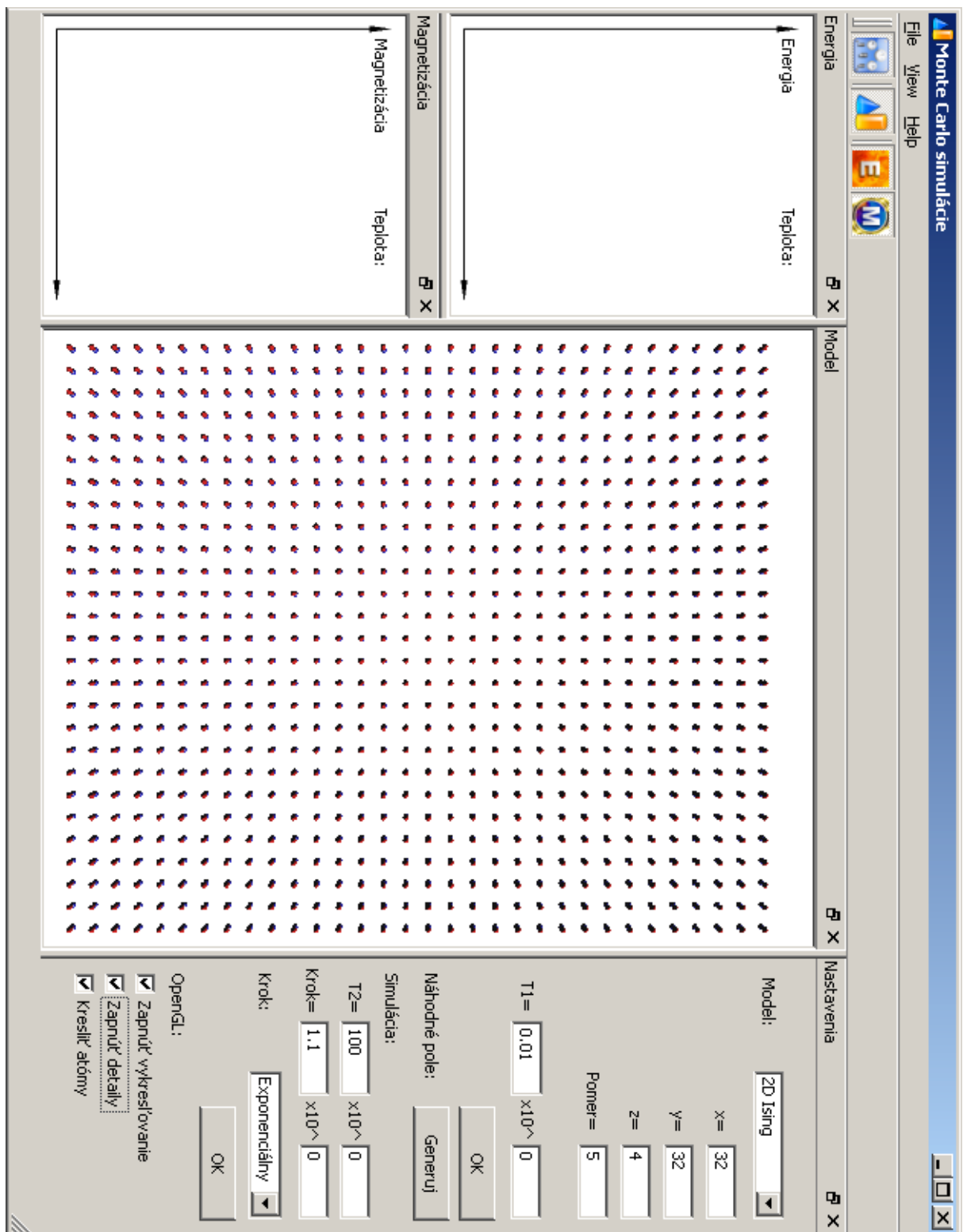
So zmenou teploty pôvodný model ostáva, zmení sa iba teplota systému.

Po kliknutí na dolné tlačítko *OK* sa nový model pred spustením simulovaného ochladzovania vygeneruje vždy.

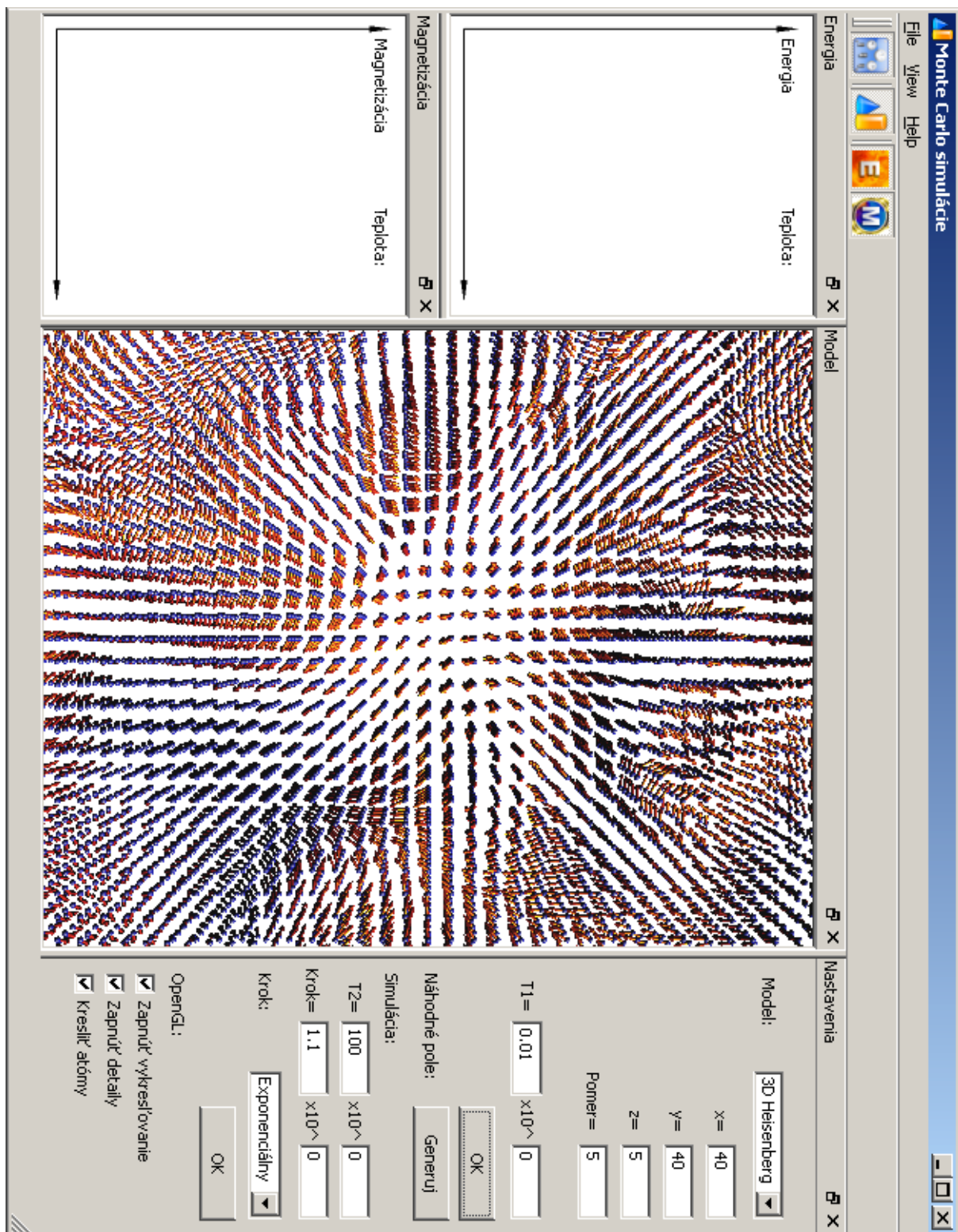
4.3 Snímky programu

Pre prehľadnosť som obrázky otočil o 90° doprava.

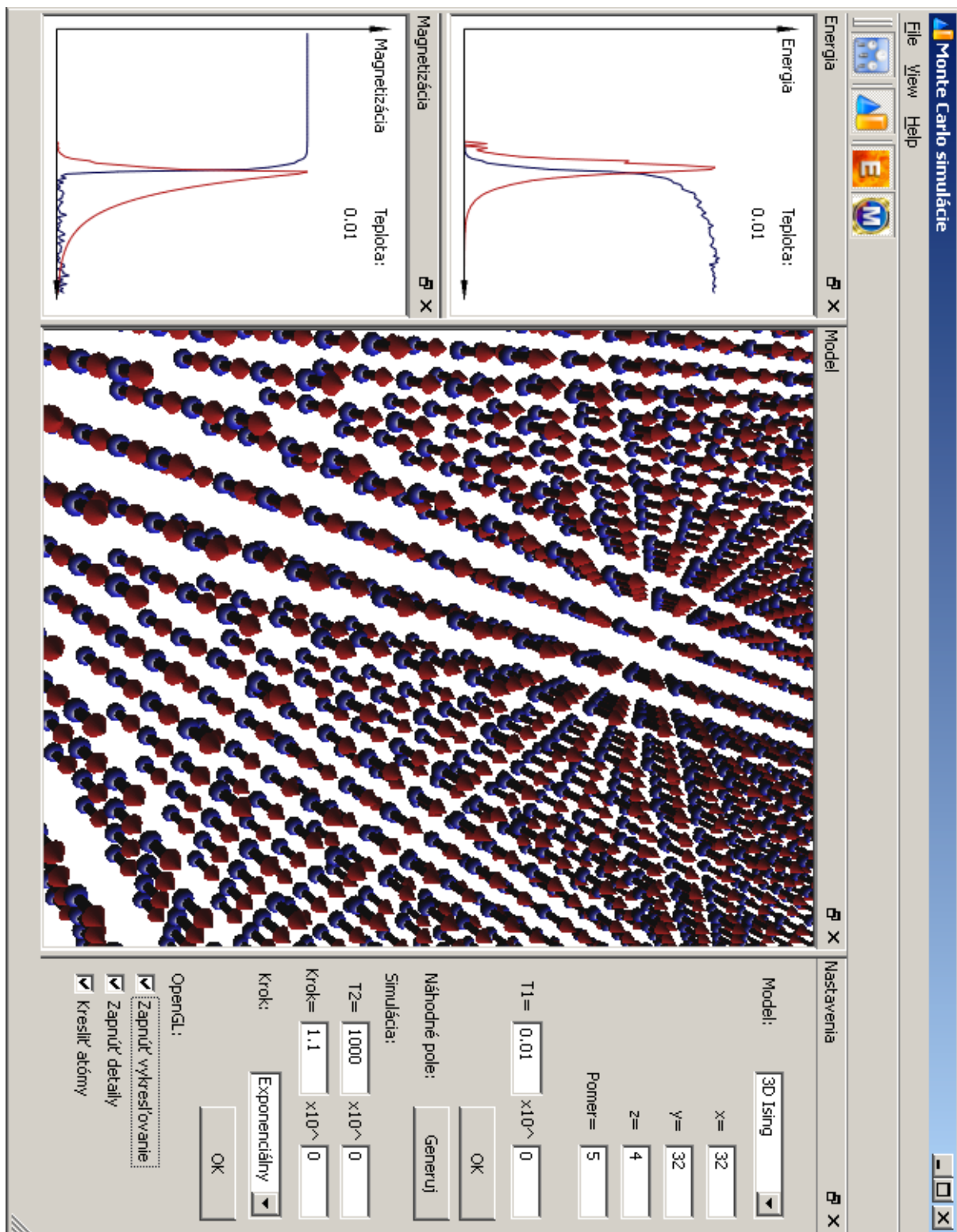
Ďalšie obrázky vygenerované programom sú použité v kapitole 2 u jednotlivých modelov.



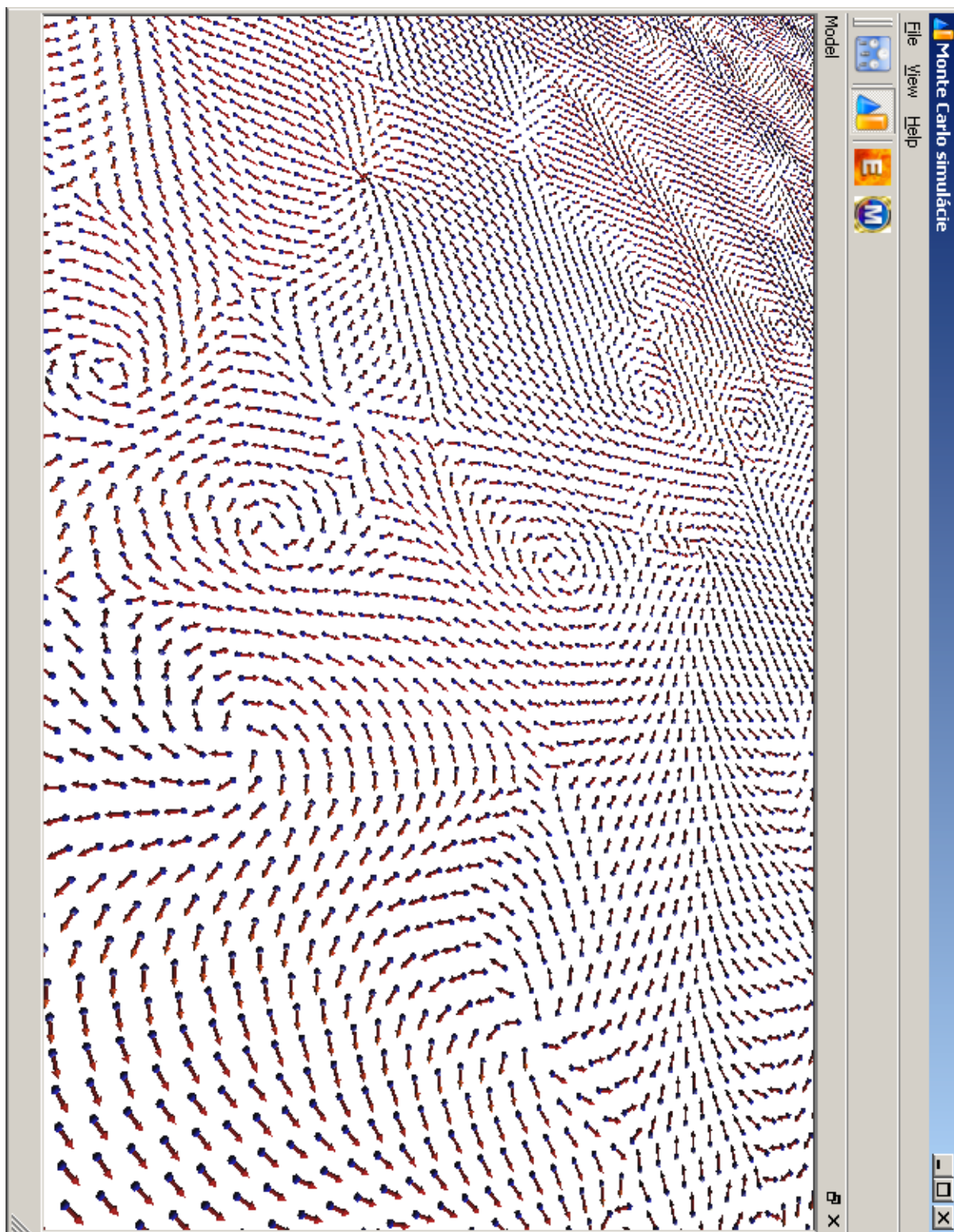
Obrázok 4.4: Screenshot programu po spustení, Isingov model.



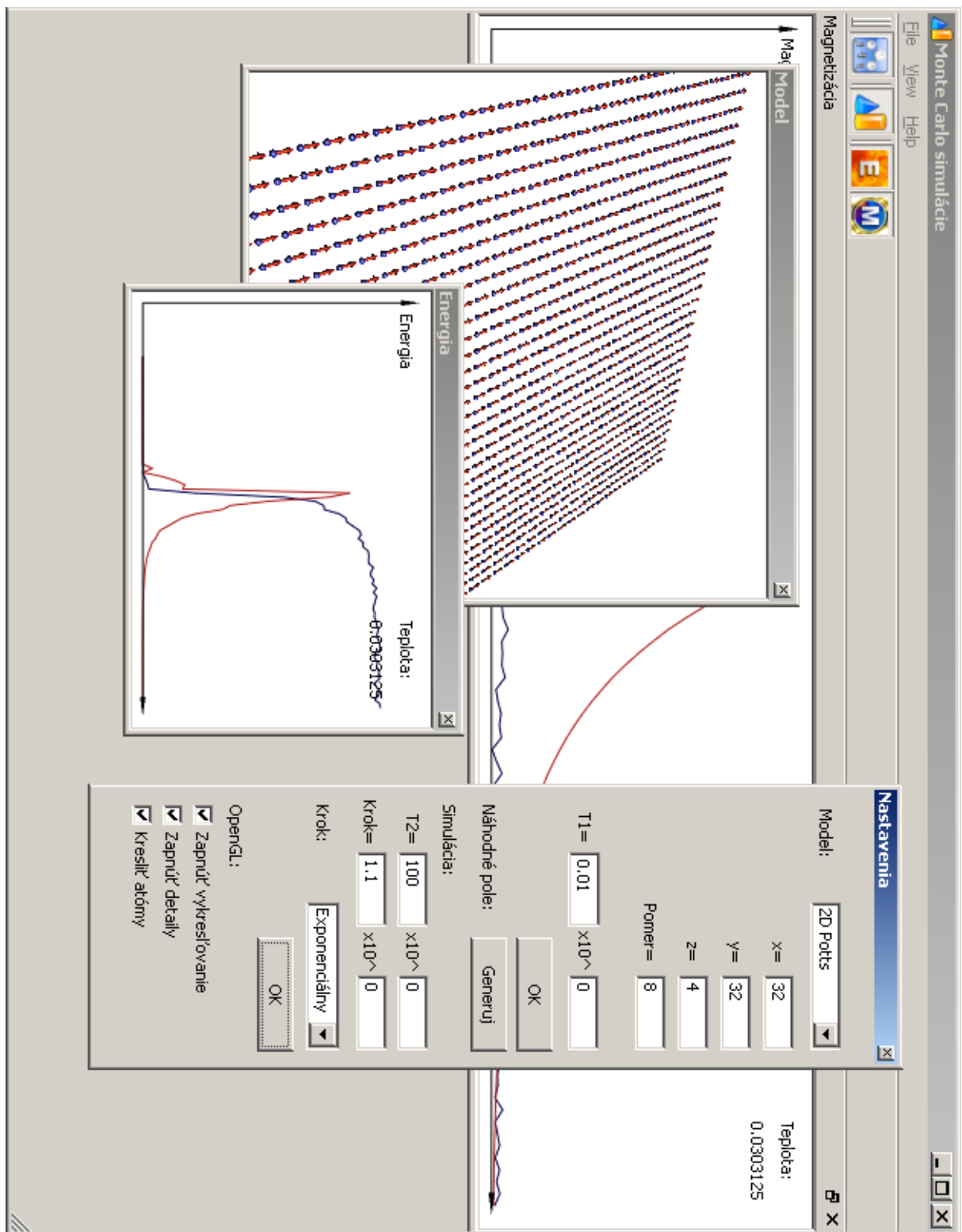
Obrázok 4.5: Screenshot programu, trojrozmerný Heisenbergov model.



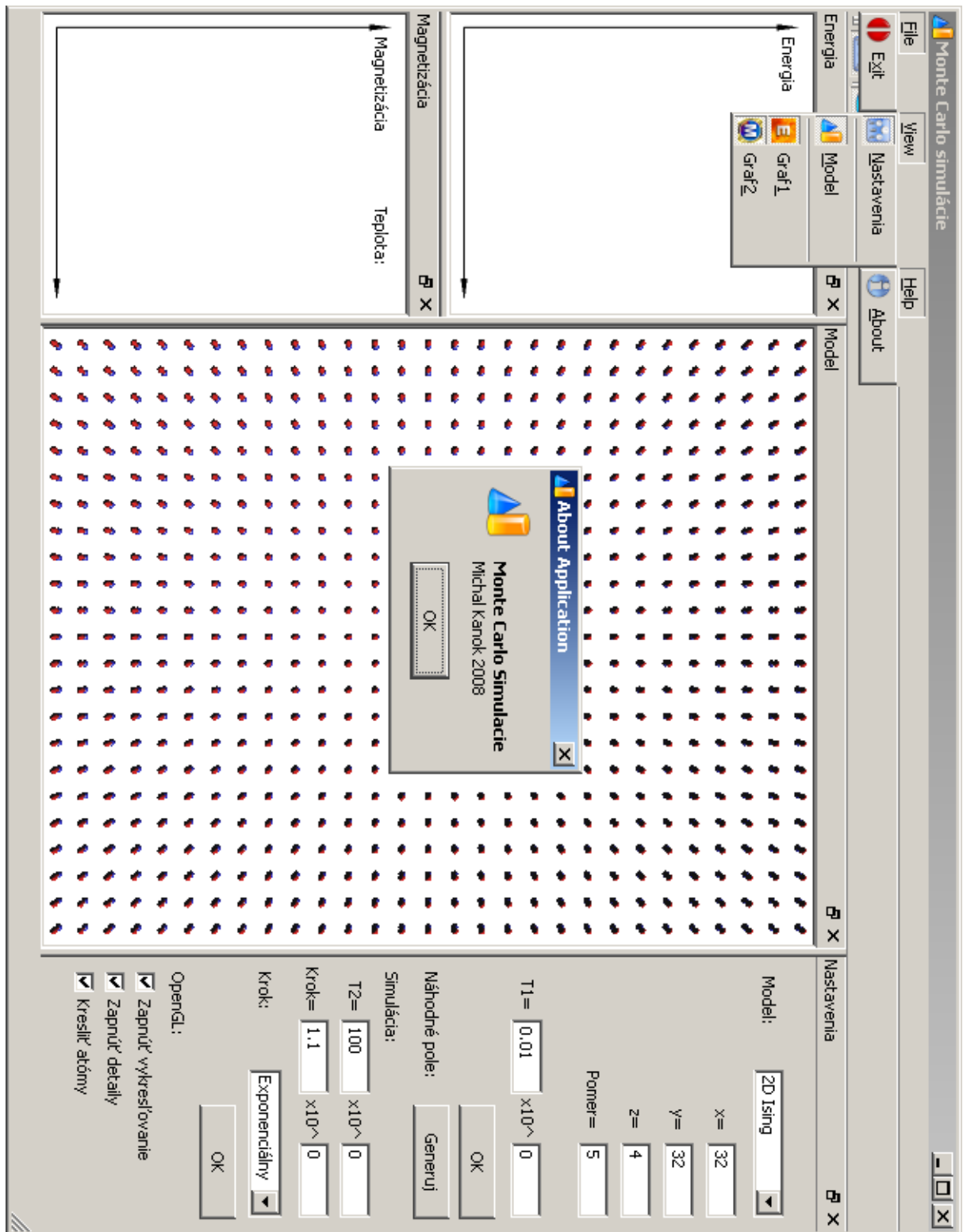
Obrázok 4.6: Screenshot programu, trojrozmerný približený Isingov model po skončení simulácie.



Obrázok 4.7: Screenshot programu, dvojrozmerný Heisenbergov model s rozmermi 150×150 , s vypnutými oknami grafov a oknom nastavení.



Obrázok 4.8: Screenshot programu z priebehu simulácie s rôzne posunutými oknami.



Obrázok 4.9: Screenshot programu zobrazujúci "falošné" celkom rozbalené menu a vyskakovacie okno "About Application".

4.4 Časová náročnosť jednotlivých modelov

Doba výpočtu pri simulovanom ochladzovaní od teploty 100 K do 0,01 K, s exponenciálnym znižovaním teploty, pre rôzne modely a ich veľkosti pri vypnutom vykresľovaní modelu:

	Rozmer	Čas výpočtu [s]
2D Ising	32×32	17
2D Ising	96×96	122
2D Potts	32×32	26
2D ZQ	32×32	58
2D Heisenberg	32×32	59
2D Heisenberg	96×96	501
3D Ising	32×32×9	140
3D Heisenberg	32×32×9	1591

Tabuľka 4.1: Časy výpočtov.

5 Záver

Zadaním a cieľom diplomovej práce bolo zoznámiť sa s problematikou metód Monte Carlo a ich implementácia na vybranom fyzikálnom systéme s fázovými prechodmi.

Pri práci som si musel naštudovať časti rôznych vedných odborov, konkrétne:

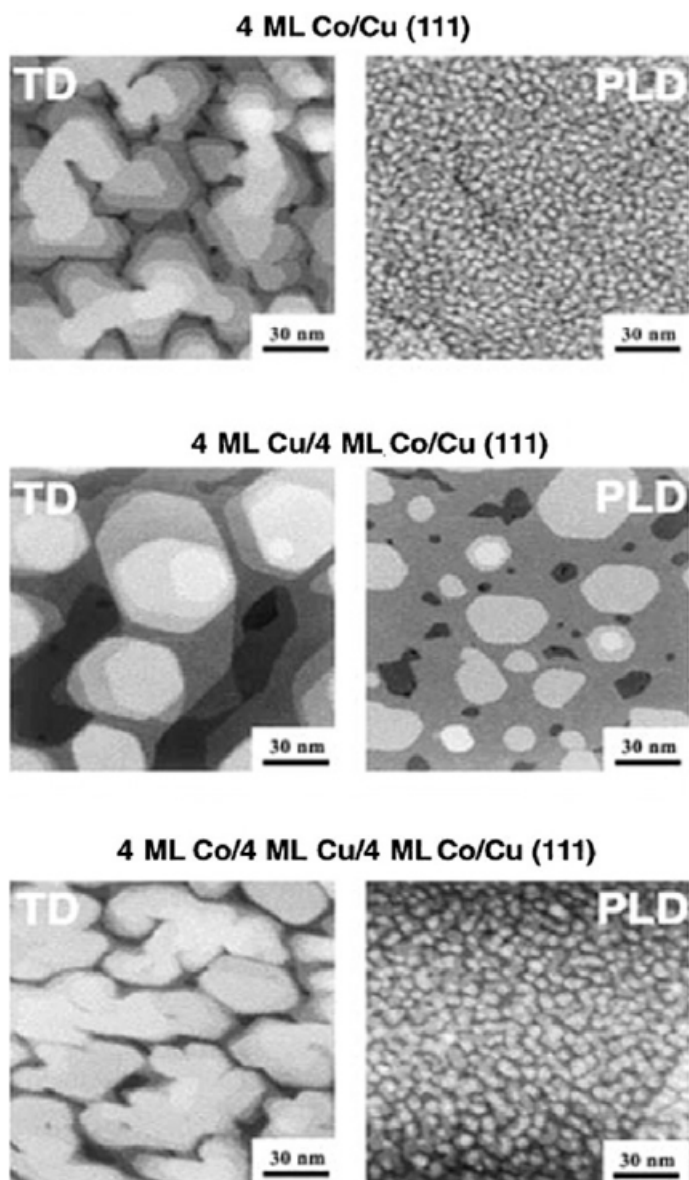
- samotné Monte Carlo metódy,
- vybrané partie zo štatistickej fyziky,
- problematiku generovania náhodných čísel,
- zväčšiť úroveň znalostí z objektového programovania, OpenGL a tvorby užívateľského rozhrania.

Ako fyzikálny systém k implementácii som zvolil ochladzovanie feromagnetických látok.

Program slúžiaci na výpočet a vizualizáciu nad fyzikálnym systémom bol naimplementovaný a funguje.

Rozhodol som sa vytvorené zdrojové kódy uvoľniť k ďalšiemu použitiu. Pravdepodobne venujem programu vlastnú internetovú stránku. Mohol by nájsť uplatnenie pri výuke Monte Carlo metód alebo feromagnetik.

5.1 Snímky skutočných feromagnetických látok



Obrázok 5.1: Snímky magnetických materiálov [15].

Na obrázku 5.1 sú snímky zachytávajúce odrazené polarizované svetlo podľa smeru magnetizácie z tenkej vrstvy feromagnetického materiálu (magneto-optický Kerrov efekt) [15].

Literatúra

[1] Newman M. E. J., Barkema G. T.: Monte Carlo Methods in Statistical Physics, Oxford University Press 2002, ISBN 0-19-851797-1

[2] Kulhánek P.: Statistická fyzika, studijní text pro doktorské studium, FEL ČVUT 2002, [online], [2008-05-05], <http://www.aldebaran.cz/studium/statistika.pdf>

[3] Nezbeda I., Kolafa J., Kotrla M.: Úvod do počítačových simulací, metody Monte Carlo a molekulární dynamiky, Univerzita Karlova v Praze, Nakladatelství Karolinum Praha 2003, ISBN 80-246-0649

[4] Článek na Wikipédii o IBM RANDU, [online], [2008-05-05], <http://en.wikipedia.org/wiki/RANDU>

[5] Stránky Katedry matematiky, University of Salzburg, [online], [2008-05-05], <http://crypto.mat.sbg.ac.at/results/karl/server/node4.html>

[6] Manuál GNU Scientific Library, [online], [2008-05-05], <http://www.gnu.org/software/gsl/manual/gsl-ref.html>

[7] Domovská stránka algoritmu Mersenne Twister, [online], [2008-05-05], <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>

[8] Domovská stránka GNU Scientific Library, [online], [2008-05-05], <http://www.gnu.org/software/gsl/>

[9] Domovská stránka Diehard testu, [online], [2008-05-05], <http://www.stat.fsu.edu/pub/diehard/>

[10] Domovská stránka TestU01 testu, [online], [2008-05-05], <http://www.iro.umontreal.ca/~simardr/testu01/tu01.html>

[11] Článek na Wikipédii o Mersenne Twister, [online], [2008-05-05], http://en.wikipedia.org/wiki/Mersenne_twister

[12] Domovská stránka frameworku Qt, [online], [2008-05-05], <http://trolltech.com/products/qt>

[13] Tišnovský P.: Seriál grafická knihovna OpenGL, [online], [2008-05-05],
<http://www.root.cz/serialy/graficka-knihovna-opengl/>

[14] Stránka ikon prostredia GNOME, [online], [2008-05-05],
<http://art.gnome.org/themes/icon/>

[15] Yu Shiratsuchi, Masahiko Yamamoto, Bader S.D.: Magnetism and surface structure of atomically controlled ultrathin metal films, Elsevier Ltd., 2006

Prílohy

A Inštalácia programu

Inštalácia programu sa spustí z CD zo súboru:

D:\monte_carlo_simulator\instalator\setup.exe ; kde D je jednotka mechaniky

Program sa nainštaluje so všetkými potrebnými knižnicami do adresára:

C:\Program Files\Monte Carlo simulator\

Samotný program spustíme súborom:

C:\Program Files\Monte Carlo simulator\Diplomka.exe

B Ovládanie programu

Ovládanie nastavení:

- prepínanie medzi položkami nastavení sa robí pomocou klávesy TAB alebo ľavým tlačítkom myši,
- nové hodnoty parametrov sa zadávajú klávesnicou.

Ovládanie menu:

- výber položiek menu sa robí pomocou myši alebo pomocou klávesovej skratky CTRL + podčiarknuté písmeno v položkách menu.

Ovládanie modelu:

- priblíženie a vzdialenie modelu docielime otáčaním kolieska myši,
- otočenie modelu okolo stredu tiahnutím myši so stlačeným ľavým tlačítkom,
- posun modelu tiahnutím myši so stlačeným pravým tlačítkom.

C Obsah priloženého CD

- zdrojové kódy programu,
- projekt Visual Studia 2005,
- inštalátor programu,
- text diplomovej práce vo formátoch .doc a .pdf.