

Metody Monte Carlo

Laboratorium 6

Zadanie 1

Kod (C++)

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <string>
#include <fmt/core.h>
#include "model_isinga.h"

using namespace std;

// https://youtu.be/K--1hLv9yv0

struct EXPERIMENT_CONFIGURATION
{
    int minimalEnergy;
    int maximalEnergy;
    int step;
    int size;
};

int main(int argc, char *argv[])
{
    EXPERIMENT_CONFIGURATION experiments[3] = {{-184, -24, 8, 10}, {-768, -32, 32, 20}, {-3072, -128,
128, 40}};

    for (int x = 0; x < size(experiments); x++)
    {
        EXPERIMENT_CONFIGURATION exp = experiments[x];

        // Open output file
        ofstream outdata;
        outdata.open(fmt::format("experiment{0}.txt", x));

        cout << fmt::format("\nRozpoczeto eksperyment nr {}", x + 1);

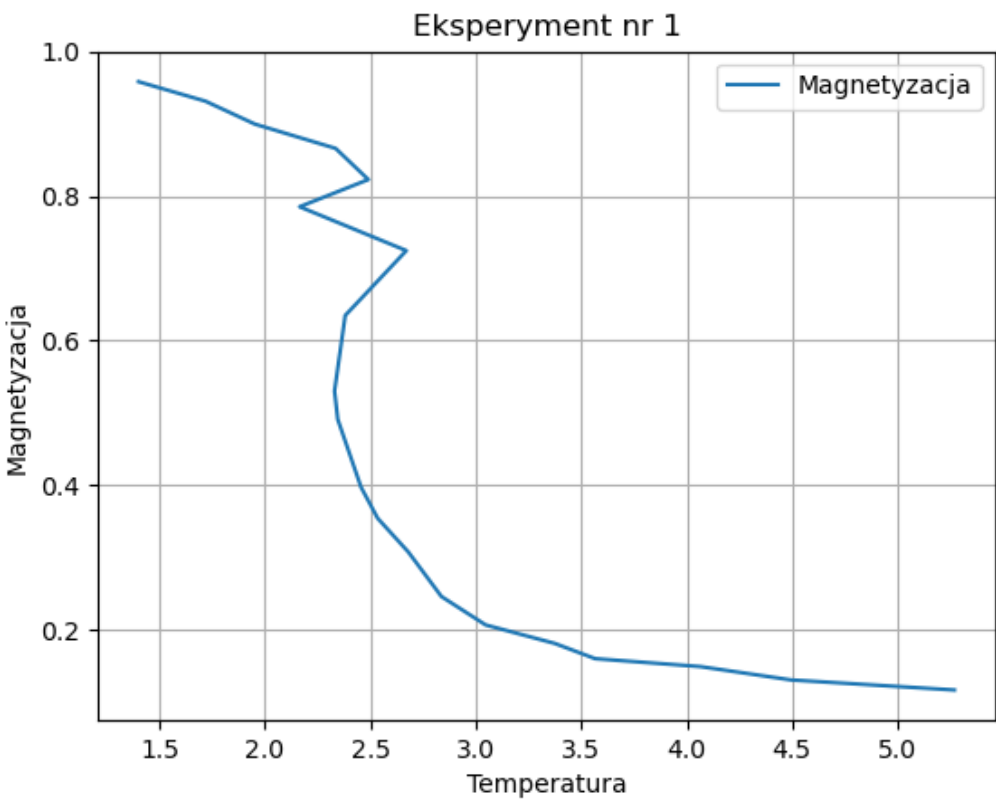
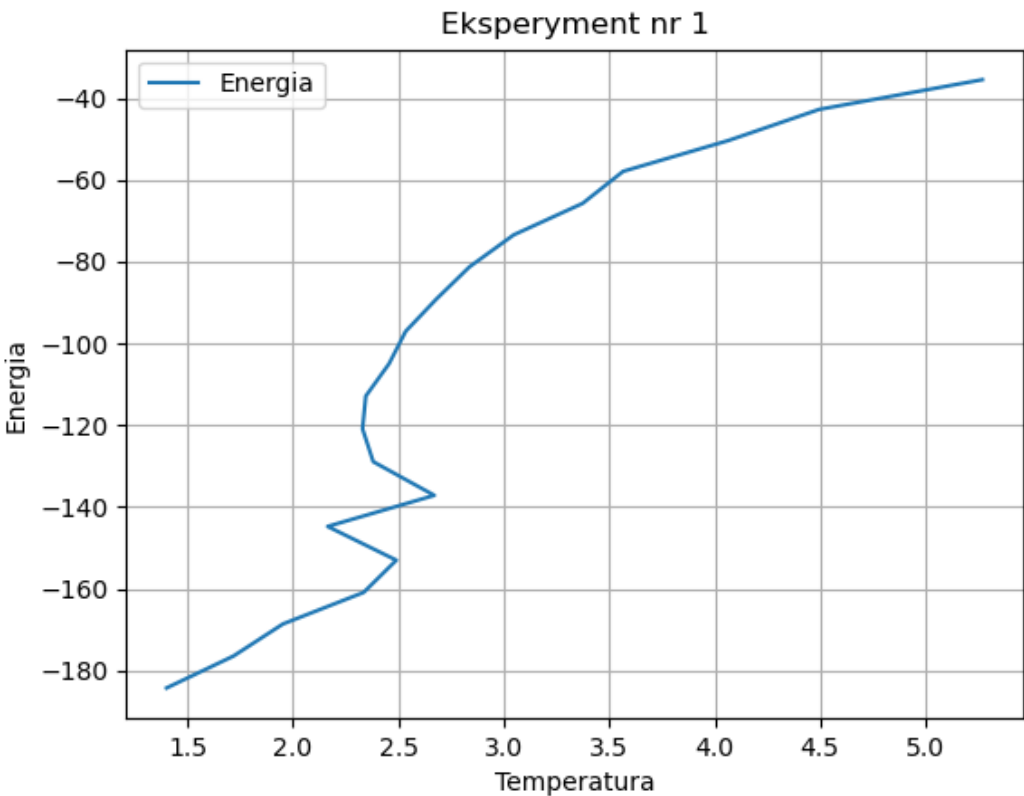
        for (int i = exp.minimalEnergy; i < exp.maximalEnergy; i += exp.step)
        {
            outdata << "Symulacja modelu Isinga w Zespole Mikrokanonicznym" << endl;
            model_isinga *p1 = new model_isinga(exp.size, i);
            p1->doprowadzenie_do_stanu_rownowagi(1000);
            p1->zliczanie_srednich(1000);
            outdata << "Srednia Energia Układu = " << p1->podaj_srednia_energie_ukladu() << endl;
            outdata << "Srednia Energia Duszka = " << p1->podaj_srednia_energie_duszka() << endl;
            outdata << "Srednia Magnetyzacja = " << p1->podaj_srednia_magnetyzacje() << endl;
            outdata << "T = " << p1->podaj_temperature() << endl;

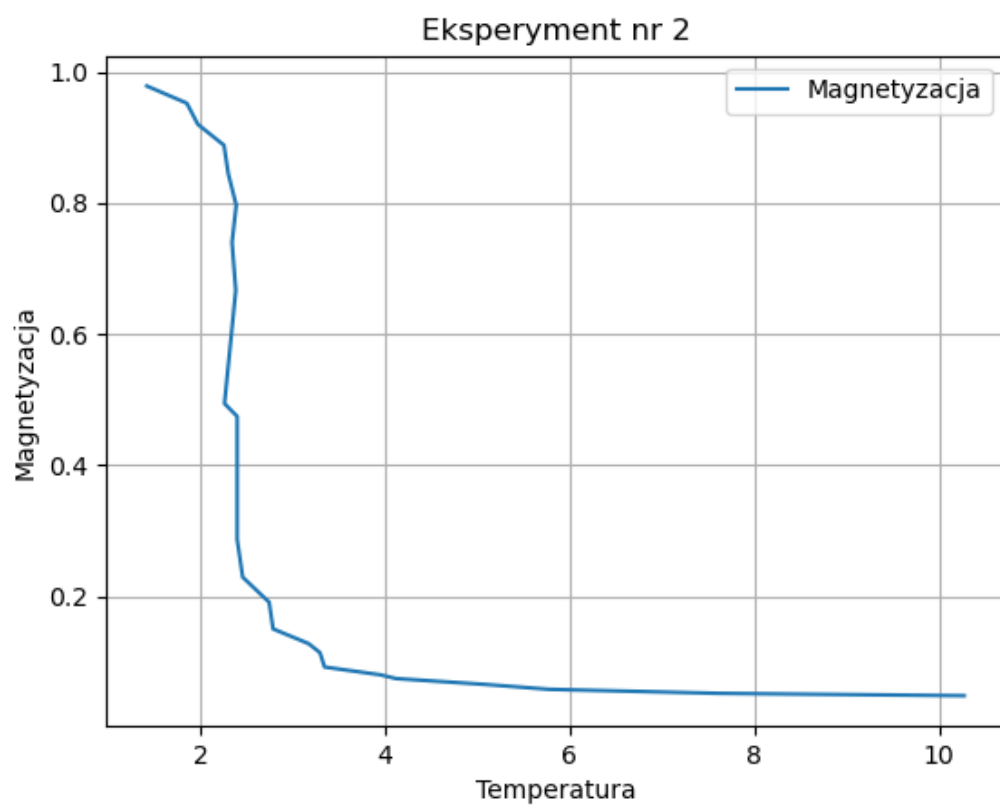
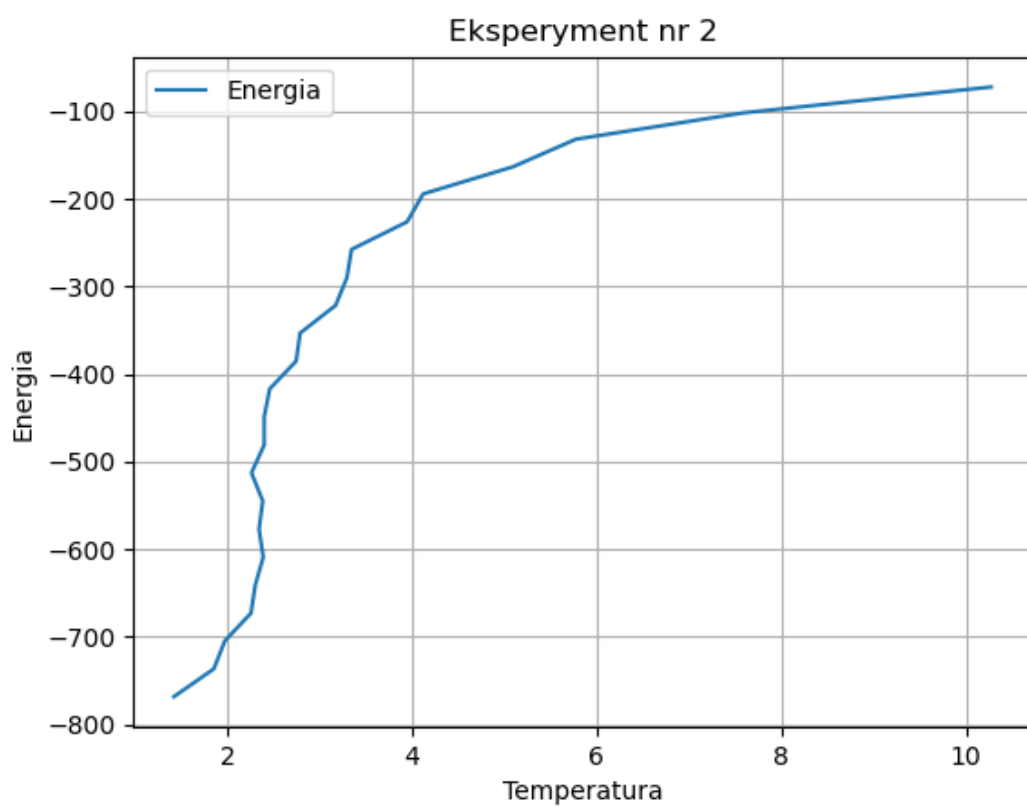
            delete p1;
        }

        cout << fmt::format("\nZakonczo eksperyment nr {}", x + 1);
        outdata.close();
    }

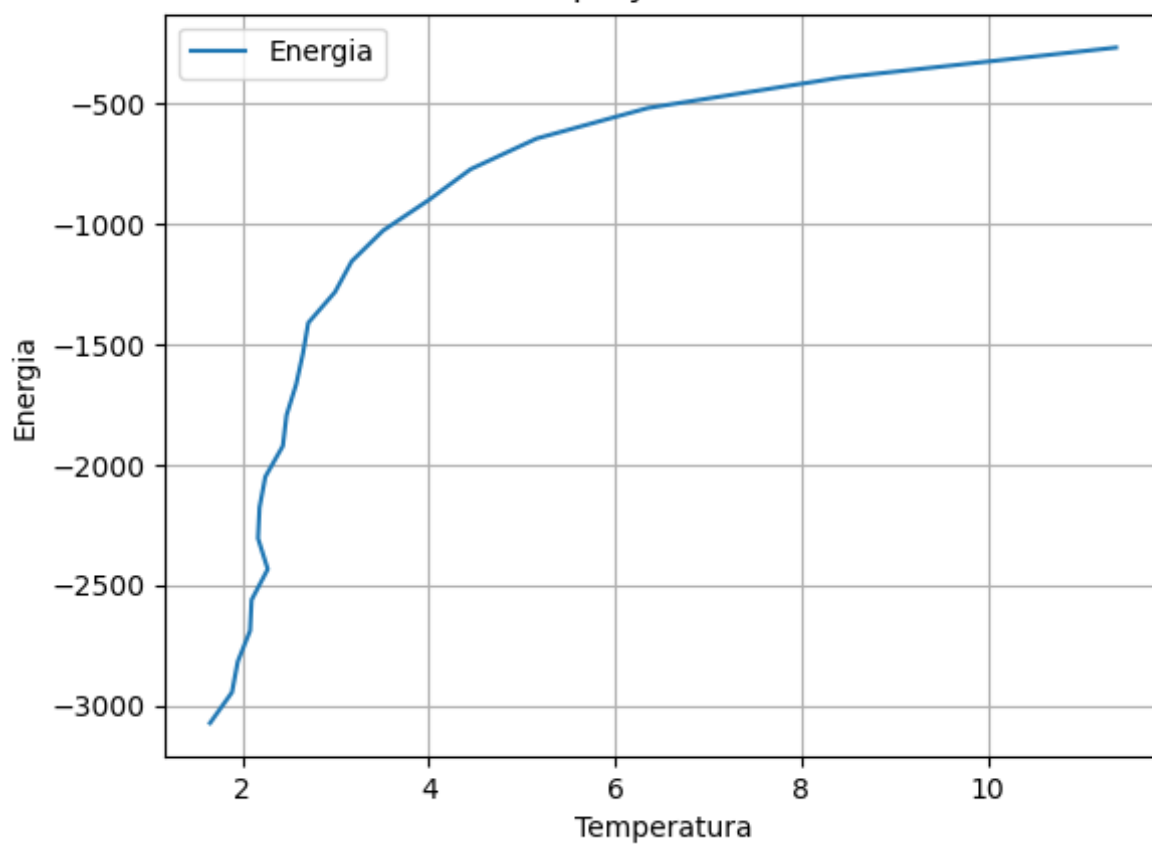
    return 0;
}
```

Wyniki

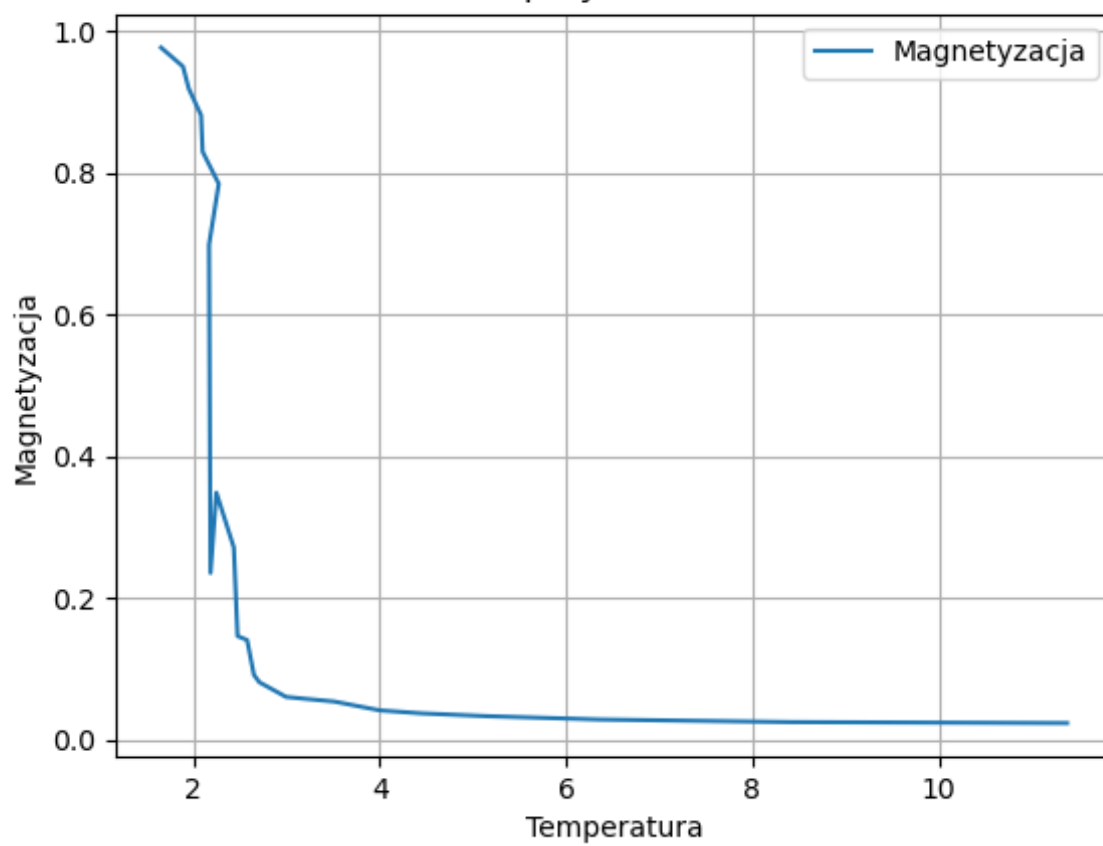




Eksperyment nr 3



Eksperyment nr 3



Im większa siatka tym dokładniejsza symulacja. Im wyższa temperatura systemu, tym magnetyzacja jest mniejsza, a energia rośnie. Pokrywa się to z termodynamicznym parametrem entropii, która mówi, że entropia układu zwiększa się (czyli wyrównuje do stanu równowagi). Tutaj mamy to samo zjawisko.

Co także jest warte uwagi to powtarzające się wartości na osi X, szczególnie na wykresach pierwszego eksperymentu. Wysoce prawdopodobne, że wynika to z niedokładności (niskiego obszaru symulacji).

Zadanie 2

Kod (C++)

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <string>
#include <vector>
#include <fmt/core.h>
#include "model_isinga.h"

using namespace std;

int main(int argc, char *argv[])
{
    int experiments[3] = {10, 20, 40};

    for (int x = 0; x < size(experiments); x++)
    {
        int size = experiments[x];
        // Open output file
        ofstream outdata;
        outdata.open(fmt::format("experiment{0}_zad2.txt", x));

        cout << fmt::format("\nRozpoczeto eksperyment nr {}", x + 1);

        for (float t = 1.0; t <= 10.0; t += 0.2)
        {
            model_isinga *p1 = new model_isinga(size, t);
            p1->doprowadzenie_do_stanu_rownowagi2(1000);
            p1->zliczanie_srednich2(1000);
            outdata << "-----" << endl;
            outdata << "T = " << t << endl;
            outdata << "Srednia Energia Układu = " << p1->podaj_srednia_energie_ukladu() << endl;
            outdata << "Srednia Magnetyzacja = " << p1->podaj_srednia_magnetyzacje() << endl;

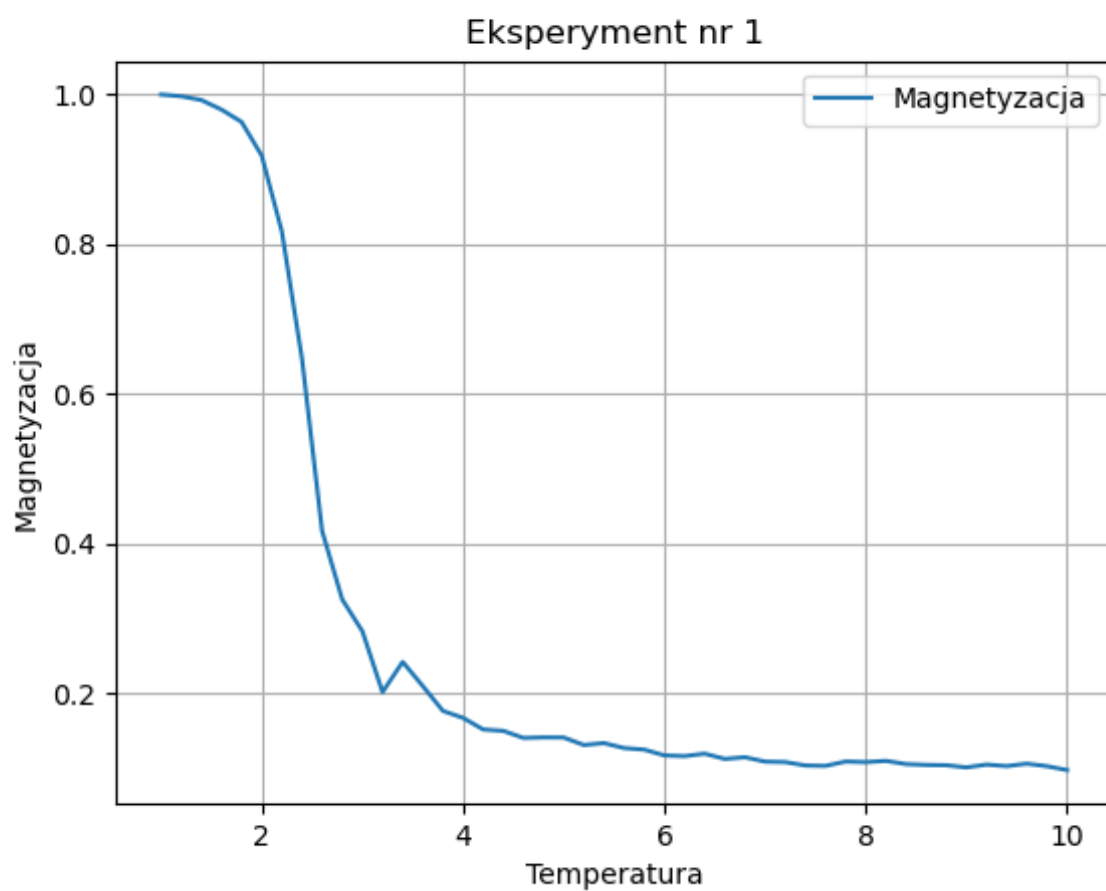
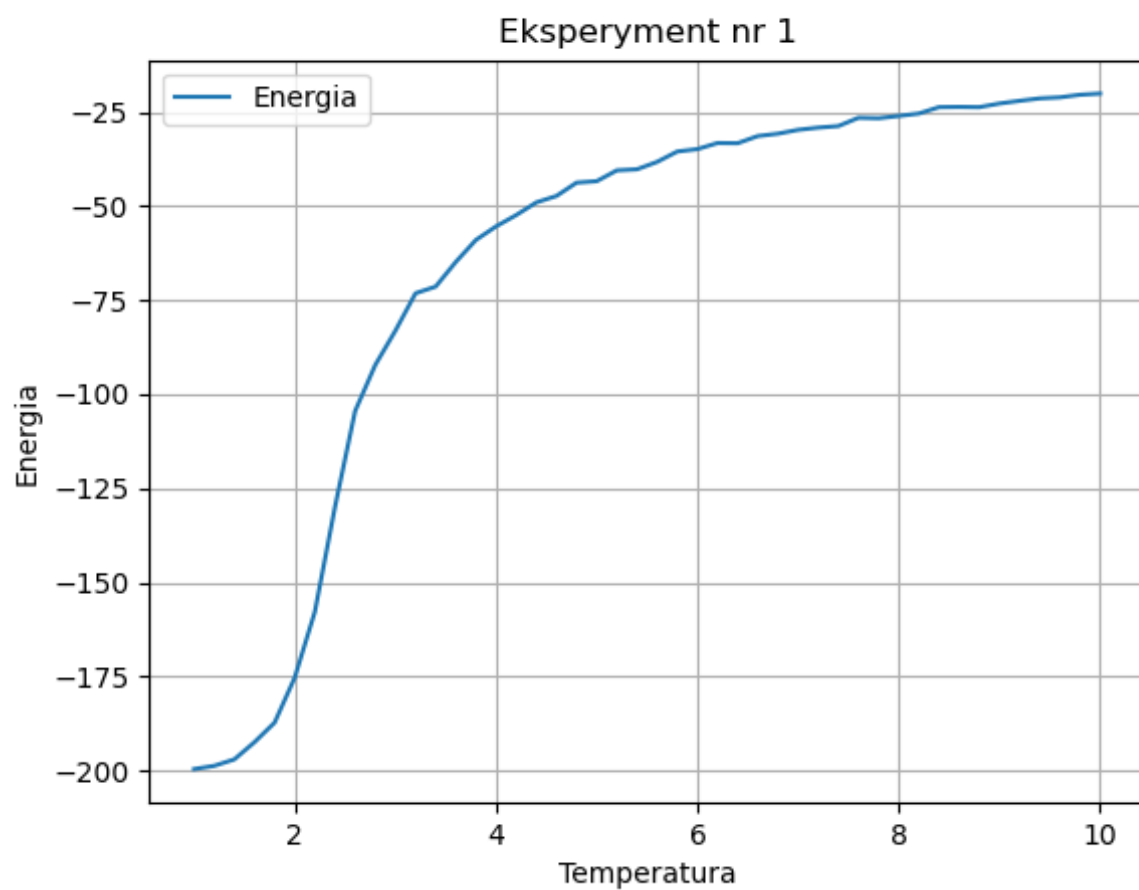
            delete p1;
        }

        cout << fmt::format("\nZakonczono eksperyment nr {}", x + 1);
        outdata.close();
    }

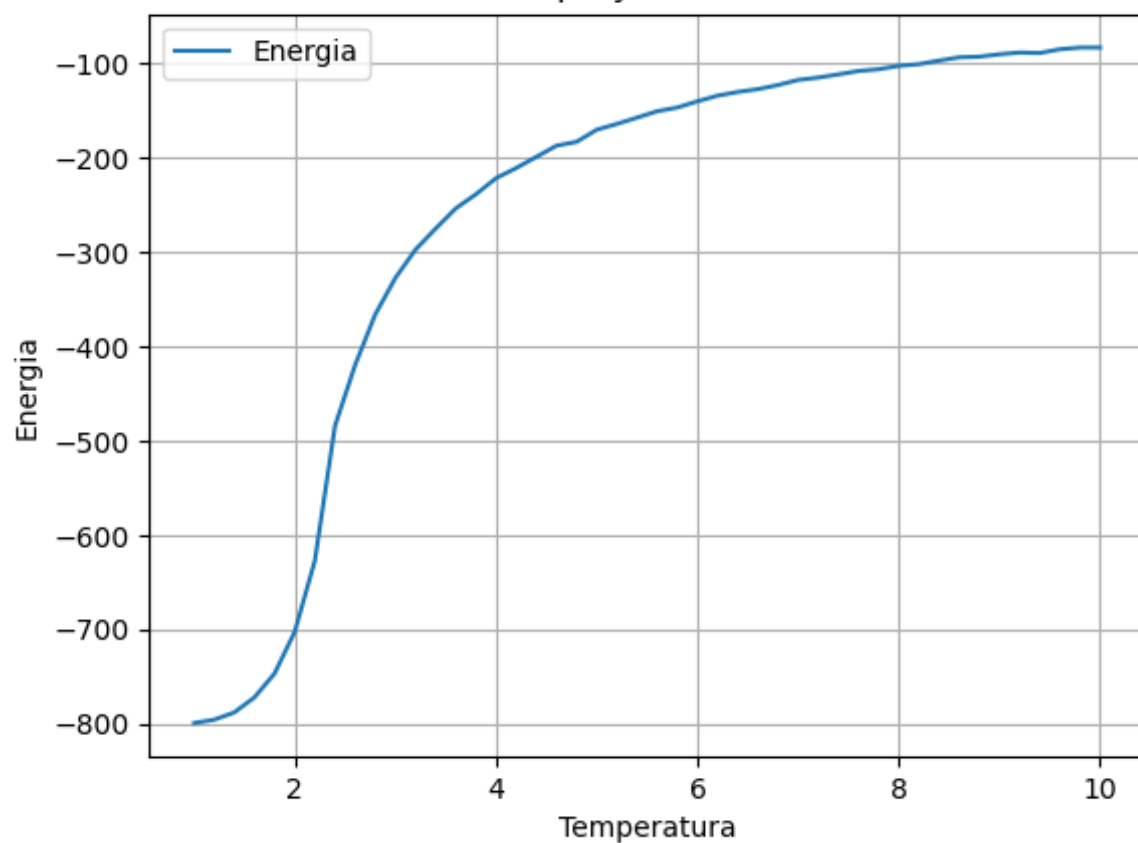
    return 0;
}
```

Wyniki

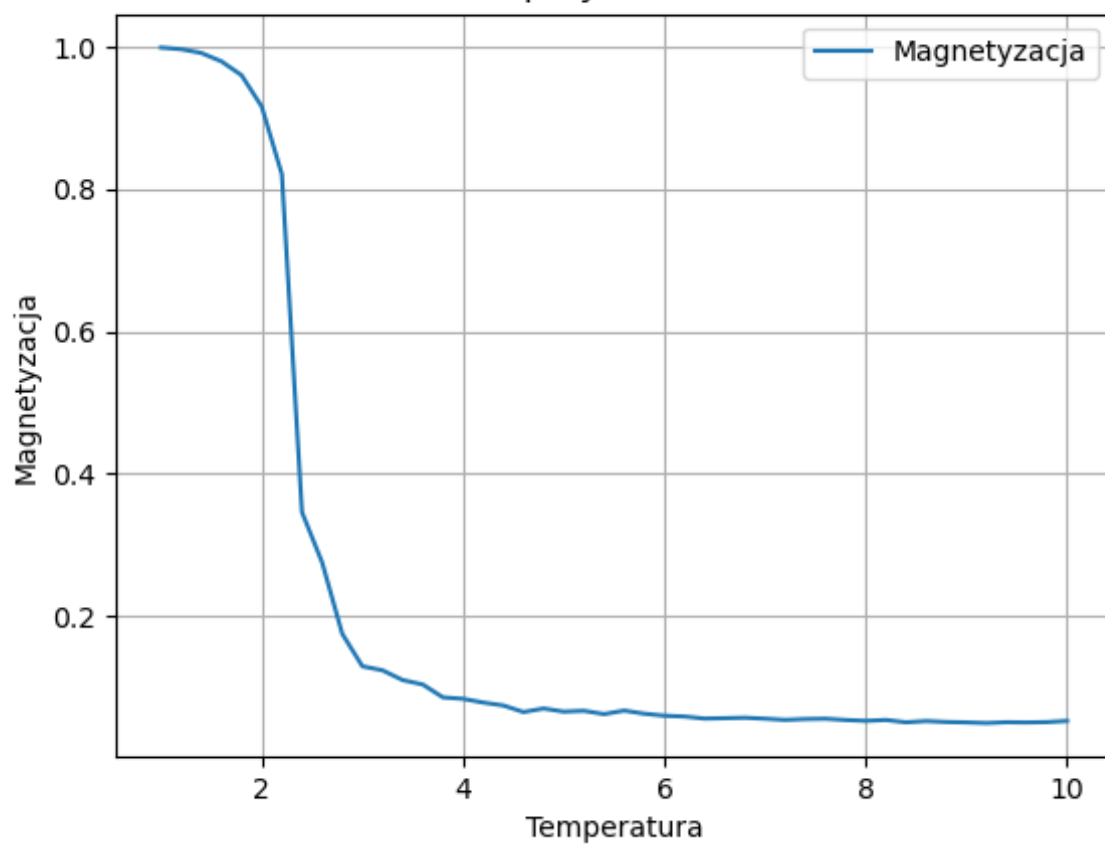
Kod generacyjny model_isinga.cpp i model_isinga.h został zmodyfikowany aby umożliwić symulację w oparciu o parametr temperatury i algorytm Metropolis'a.



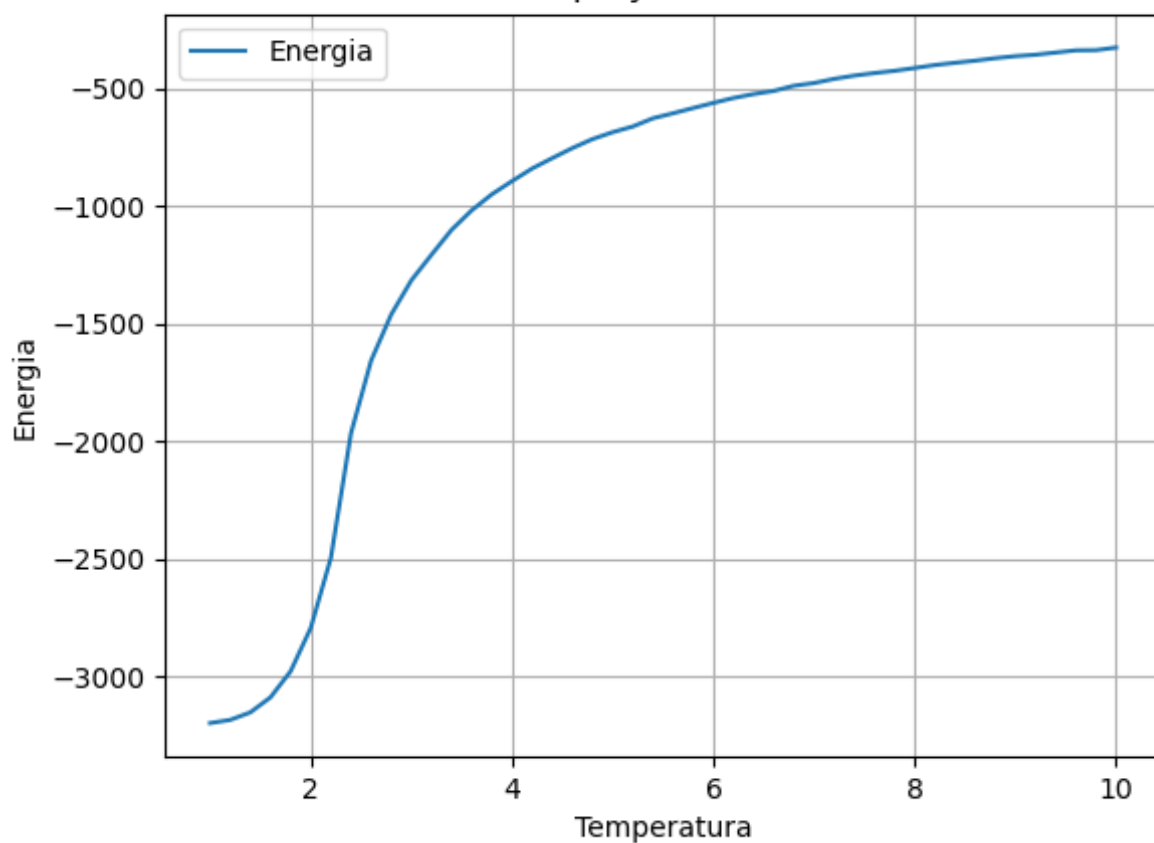
Eksperyment nr 2



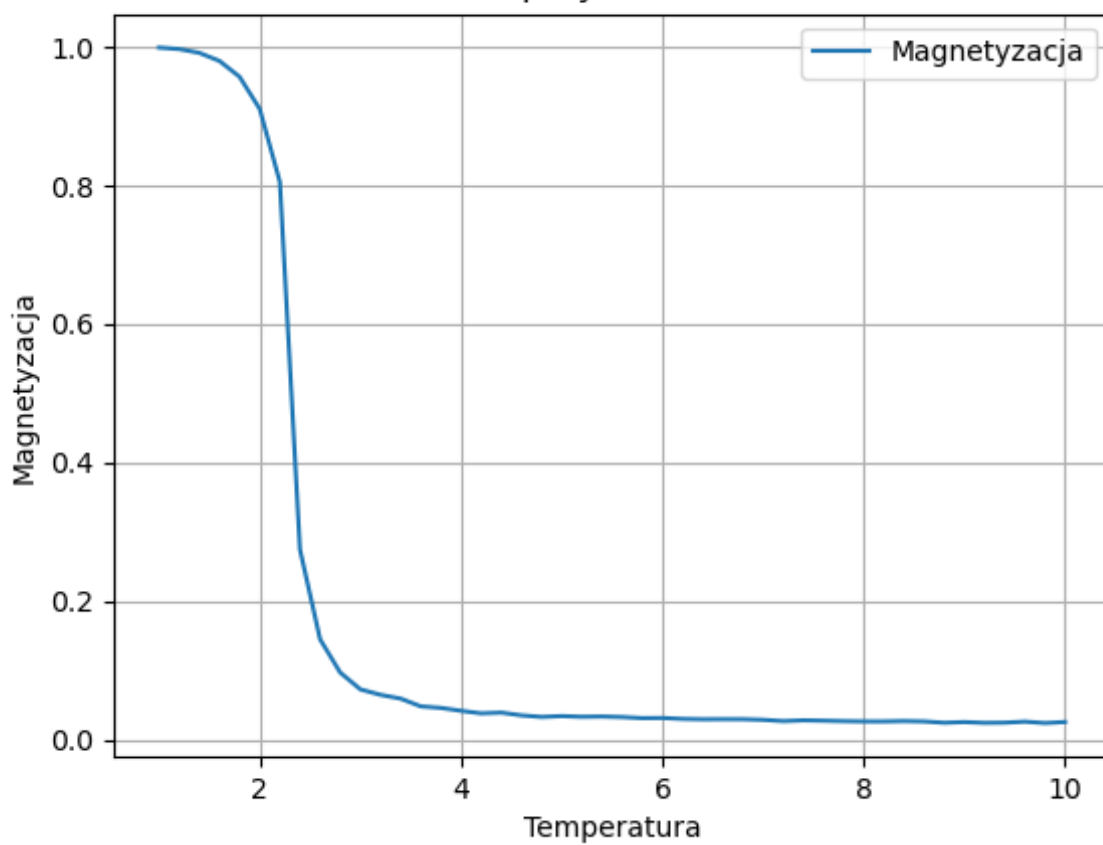
Eksperyment nr 2



Eksperyment nr 3



Eksperyment nr 3



Symulacja w układzie kanonicznym dostarcza dokładniejsze dane, ale zmiana energii od temperatury oraz magnetyzacji od temperatury pozostaje na tym samym poziomie jak w układzie mikrokanonicznym.