Metody Obliczeniowe Fizyki i Techniki
Laboratorium 4
Zastosowanie schematu Metropolisa:model Isinga 2D

Krzysztof Tondera III rok

09.05.2021

## 1 Siatka $2^n \cdot 2^n$ zlokalizowanych spinów $\frac{1}{2}$

Na początku zaimplementowano klasę która tworzy siatkę na wzór tej z modelu Isinga, i zaimplementowano odpowiednie parametry: $\mu$B=0.1 meV, J=100 meV oraz badano zakres temperatur kT∈(0,1000) meV

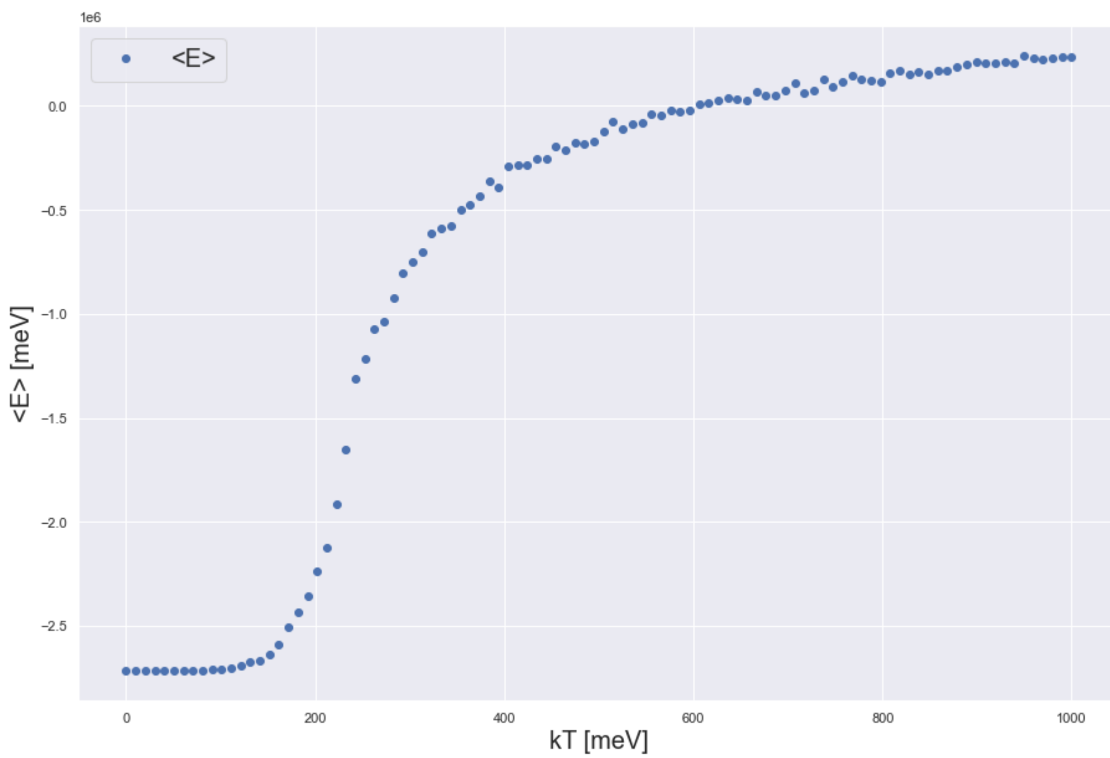
[31]:
```
class lattice:

    def __init__(self,n,mu_B,J):
        self.n=n
        self.mu_B=mu_B
        self.J=J

    dim=2**7

    state = np.ones(shape=(dim,dim))

    def show(self):
        fig, ax = plt.subplots(constrained_layout=True)
        cs = ax.contourf(self.state,cmap="bwr",levels=1)
        cbar = fig.colorbar(cs)
        plt.title("Ising Model")
        plt.show()

    def energy(self):
        E=0
        for i in range(self.dim):
            for j in range(self.dim):
                S=self.state[i,j]
                nb = self.state[(i+1)%self.dim, j] + self.state[i,(j+1)%self.dim] + self.
↪state[(i-1)%self.dim, j] + self.state[i,(j-1)%self.dim]
                E += -self.J*nb*S-self.mu_B*self.state[i,j]
        return E/2/(2**(2*self.n))

    def metropolis_algorithm(self,kT_max):

        kT_tab=np.linspace(0.01,kT_max,100)
        E=self.energy()
        M=0
```

```
        avg_E=[]
        M_tab=[]

        start = time.time()
        for kT in kT_tab:
            for i in range(30000):
                a=np.random.randint(0, self.dim)
                b=np.random.randint(0, self.dim)

                sn = self.state[(a+1)%self.dim, b] + self.state[a,(b+1)%self.dim] + self.
↪state[(a-1)%self.dim, b] + self.state[a,(b-1)%self.dim]
                dE=2*self.mu_B*self.state[a][b]+2*self.J*self.state[a][b]*sn

                if random.random() < np.exp(-dE/kT):
                    #print(E,dE)
                    E=E+dE
                    self.state[a,b]*=-1


            M=sum(sum(self.state))
            avg_E.append(E)
            M_tab.append(M)
        end = time.time()
        print("{} s".format(end - start))
        return avg_E,M_tab,kT_tab


    def metropolis_correlation(self,kT_max,N):

        kT_tab=np.linspace(0.01,kT_max,100)
        E=self.energy()
        M=0
        d=2

        avg_E=[]
        M_tab=[]
        sigm_tab=[]

        dim0 = int(self.dim/2)

        start = time.time()
        for kT in kT_tab:
            sigm1=0
            sigm2=0
            sigm= 0
            for i in range(N):
                a=np.random.randint(0, self.dim)
                b=np.random.randint(0, self.dim)

                sn = self.state[(a+1)%self.dim, b] + self.state[a,(b+1)%self.dim] + self.
↪state[(a-1)%self.dim, b] + self.state[a,(b-1)%self.dim]
```

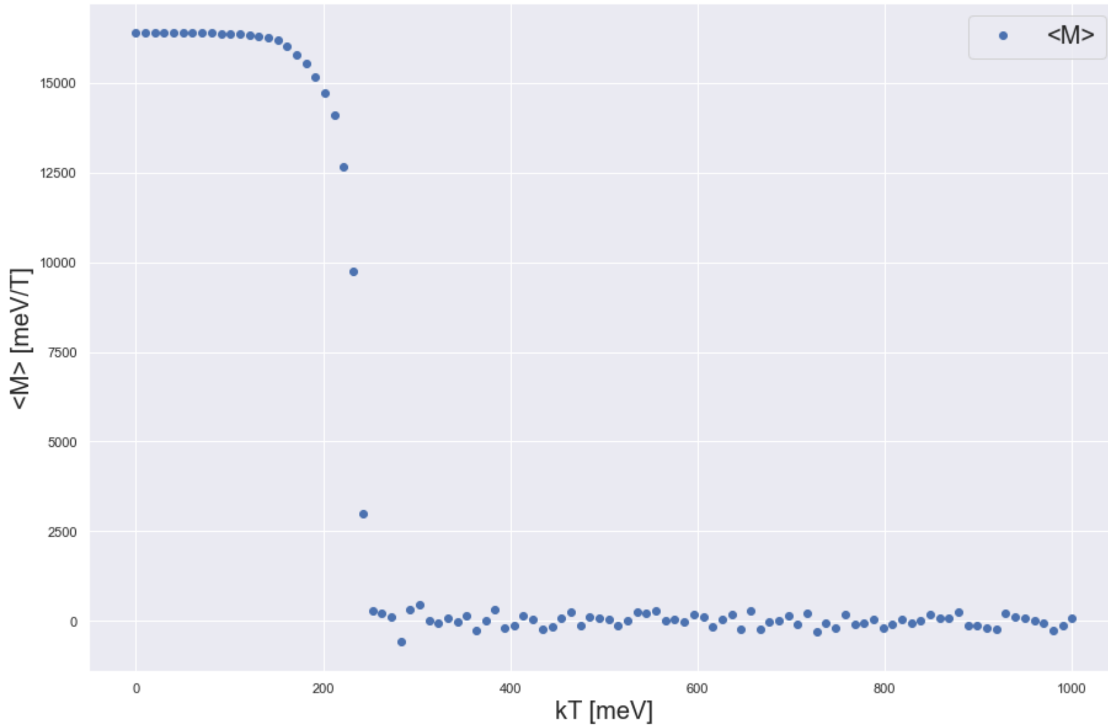
```
                dE=2*self.mu_B*self.state[a][b]+2*self.J*self.state[a][b]*sn


                sigm1 += self.state[dim0,dim0]
                sigm2 += self.state[dim0-d,dim0-d]
                sigm += self.state[dim0,dim0]*self.state[dim0-d,dim0-d]

                if random.random() < np.exp(-dE/kT):
                    #print(E,dE)
                    E=E+dE
                    self.state[a,b]*=-1


            M=sum(sum(self.state))
            avg_E.append(E)
            M_tab.append(M)
            sigm_tab.append(sigm/N-sigm1/N*sigm2/N)
        end = time.time()
        print("{} s".format(end - start))
        return avg_E,M_tab,kT_tab,sigm_tab
```

## 1.1 Zadania

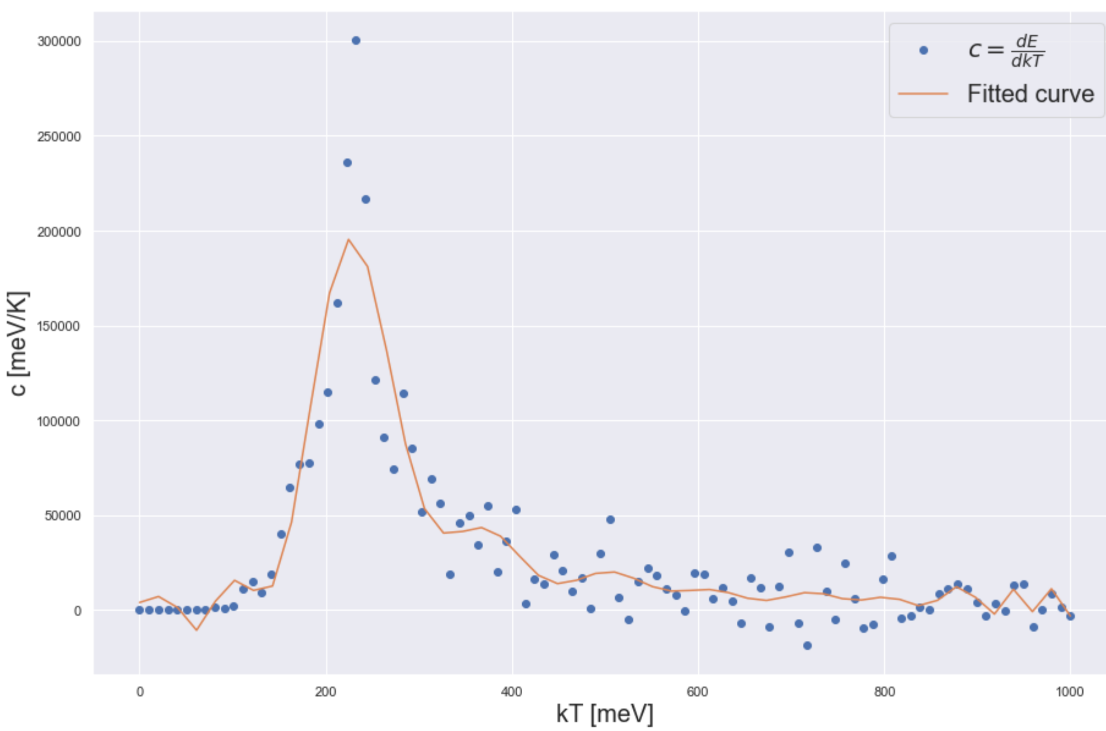
Za pomocą metody tej klasy metropolis_algorithm() dokonano obliczeń energii za pomocą schematu Metropolisa. Narysowano wykres średniej energii w funkcji kT:



Oraz średnią magnetyzację w funkcji kT:

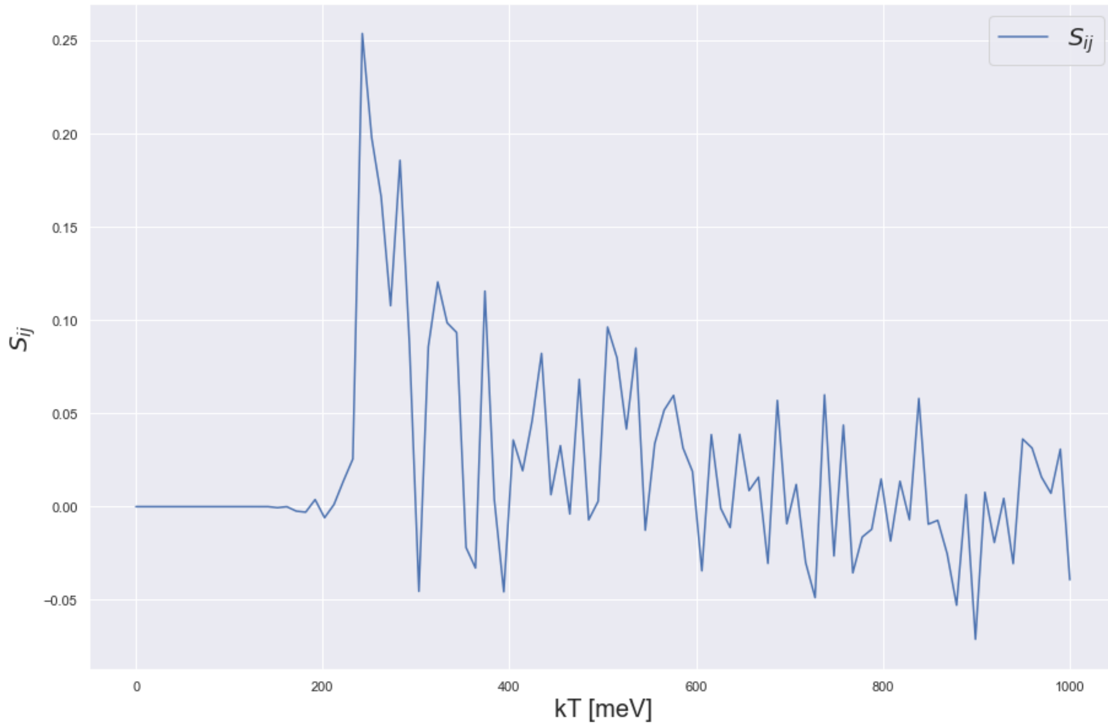



Następnie obliczono gradient średniej enrgii $\frac{dE}{d(kT)}$ wyznaczając ciepło właściwe układu. Narysowano wykres ciepła właściwego c w funkcji kT:




I wyznaczono kT przy którym następuje przejście fazowe: 224.31 meV. Wyznaczono tą wartość znajdując maksimum globalne wykresu ciepła właściwego.

Na samym końcu zbadano średnią wartość skorelowania zdefiniowanego jako $S_{ij} =< \sigma_i \sigma_j > - < \sigma_i >< \sigma_j >$ dla i w środku pudła oraz dla spinóW j położonych w pewnej odległości od i. Do naszych obliczeń przyjęto odległość d=2. Następnie narysowano wykres $S_{ij}$ w funkcji kT:




Obliczenia wykonano dla N=6 000 000 losować w schemacie Metropolisa. Zauważono największą korelację dla kT=242.43 meV co jest bardzo bliskie punktu przejścia fazowego.

Poniżej przedstawiono stan siatki w modelu Isinga po całej symulacji zmian układu schematem Metropolisa: