



Modelowanie wieloskalowe

Metoda Monte Carlo - podstawy

Dr hab. inż. **Łukasz Madej**, prof. AGH
Katedra Informatyki Stosowanej i Modelowania
Wydział Inżynierii Metali i Informatyki Przemysłowej

Budynek B5
p. 716
lmadej@agh.edu.pl
home.agh.edu.pl/lmadej



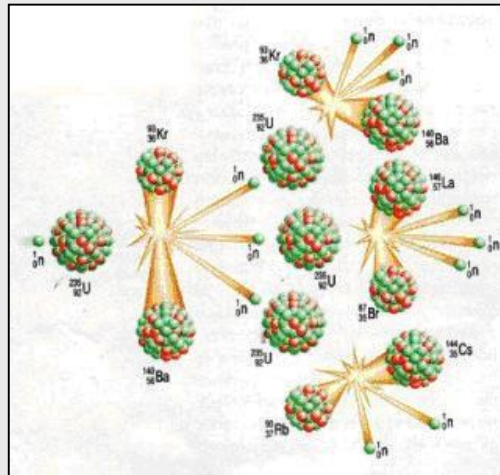
Podstawą metody Monte Carlo jest losowe próbkowanie przestrzeni rozwiązań mające na celu rozwiązanie rozpatrywanego zagadnienia.

Nazwa metody została wprowadzona do powszechnego użytku w latach 40 dwudziestego wieku, przez naukowców pracujących nad rozwojem broni jądrowej w Instytucie Los Alamos.

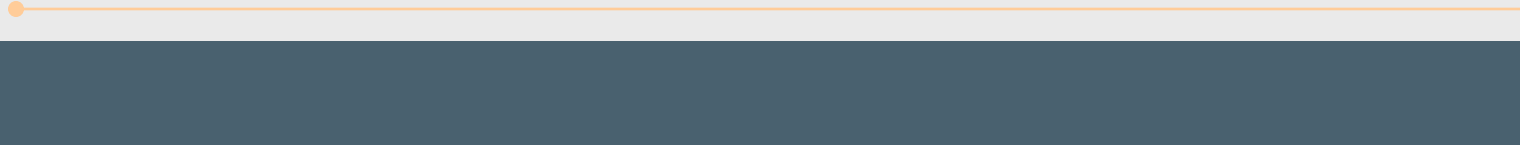




Metoda ta znalazła zastosowanie do symulacji losowego zachowania się neutronów w materiałach rozszczepialnych.



Wraz z rozwojem mocy obliczeniowych komputerów zaczęto ją wykorzystywać do symulacji wielu **fizycznych i matematycznych** zagadnień.



Pod nazwą Monte Carlo nie kryje się jedna konkretna metoda obliczeniowa, a raczej cała klasa zbliżonych do siebie metod, których podstawowe założenia bazują na jednym algorytmie:

Krok 1: Definicja przestrzeni możliwych danych wejściowych,

Krok 2: Losowe określenie danych wejściowych z wcześniej określonej przestrzeni,

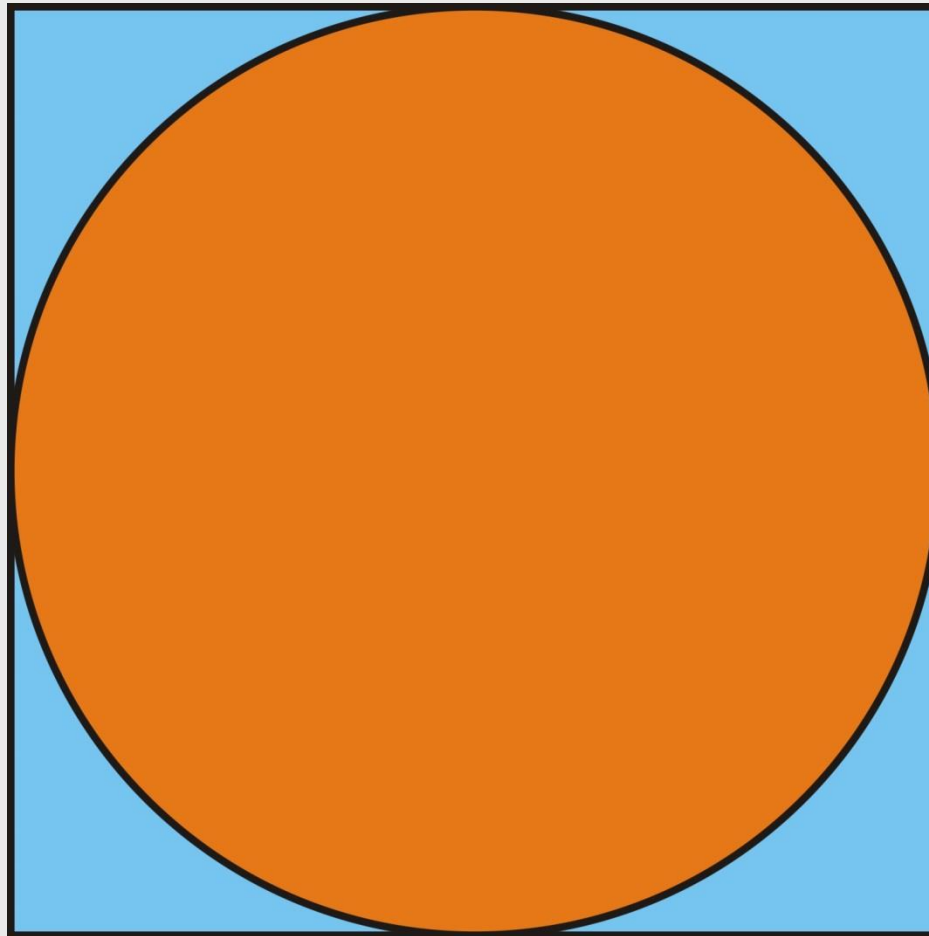
Krok 3: Przeprowadzenie obliczeń o charakterze probabilistycznym wykorzystując w/w dane wejściowe,

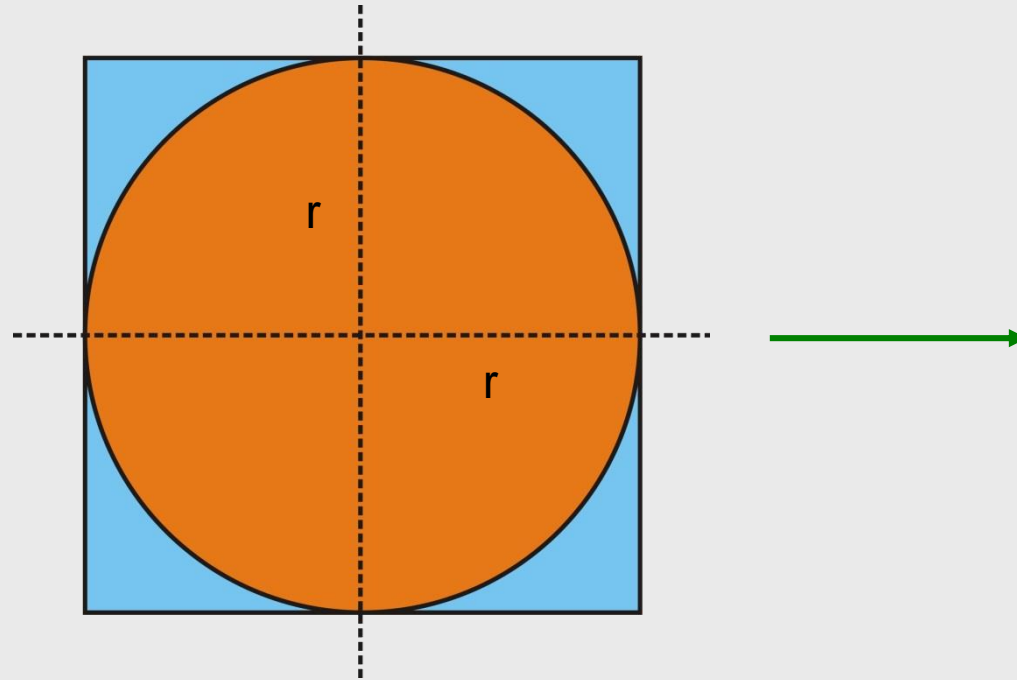
Krok 4: Przeprowadzenie agregacji uzyskanych wyników w jedno rozwiązanie końcowe.





Określenie liczby π



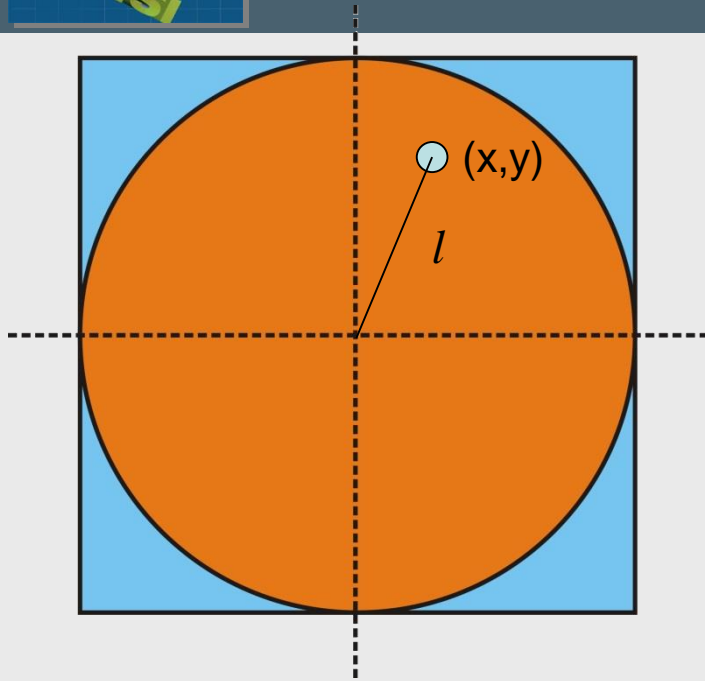


Gra w Darta



Przy całkowicie losowych rzutach ilość trafień w kwadratowy obszar i ilość trafień w tarcze są proporcjonalne do tych dwóch obszarów

$$\frac{\text{ilość trafień w okrągla tarcze}}{\text{ilość trafień w kwadratowy obszar}} = \frac{\text{okrągły obszar tarczy}}{\text{obszar kwadratu}}$$



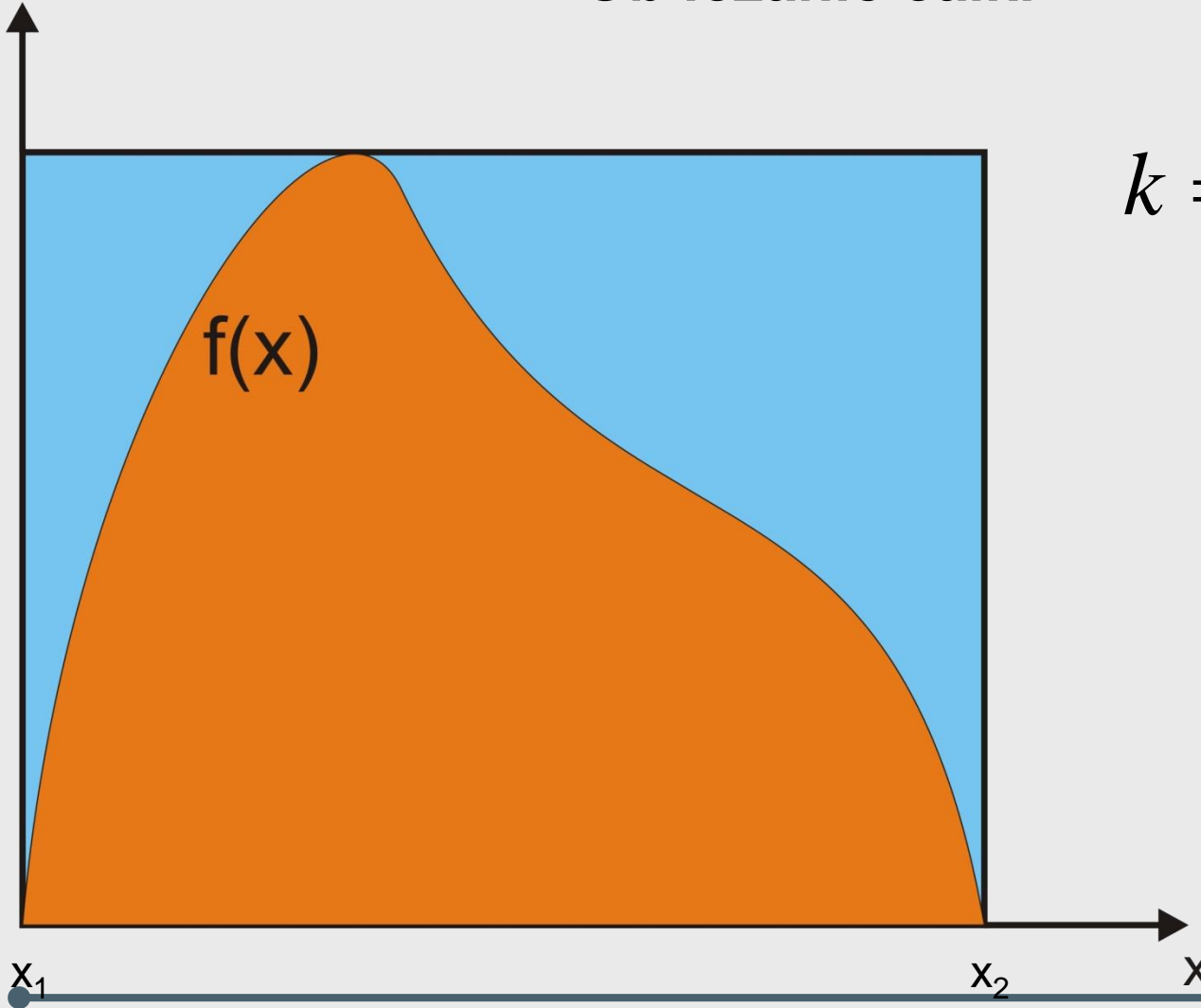
```
for(i=0; i<n;i++){  
    x = rand()  
    y = rand()  
    l = sqrt(x^2 + y^2)  
    if (l <= r) {  
        count++  
    }  
}
```

$$\frac{\text{ilosc trafien w okragla tarcze}}{\text{ilosc trafien w kwadratowy obszar}} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}$$

$$\pi = 4 \frac{\text{ilosc trafien w okragla tarcze}}{\text{ilosc trafien w kwadratowy obszar}}$$



Obliczanie całki

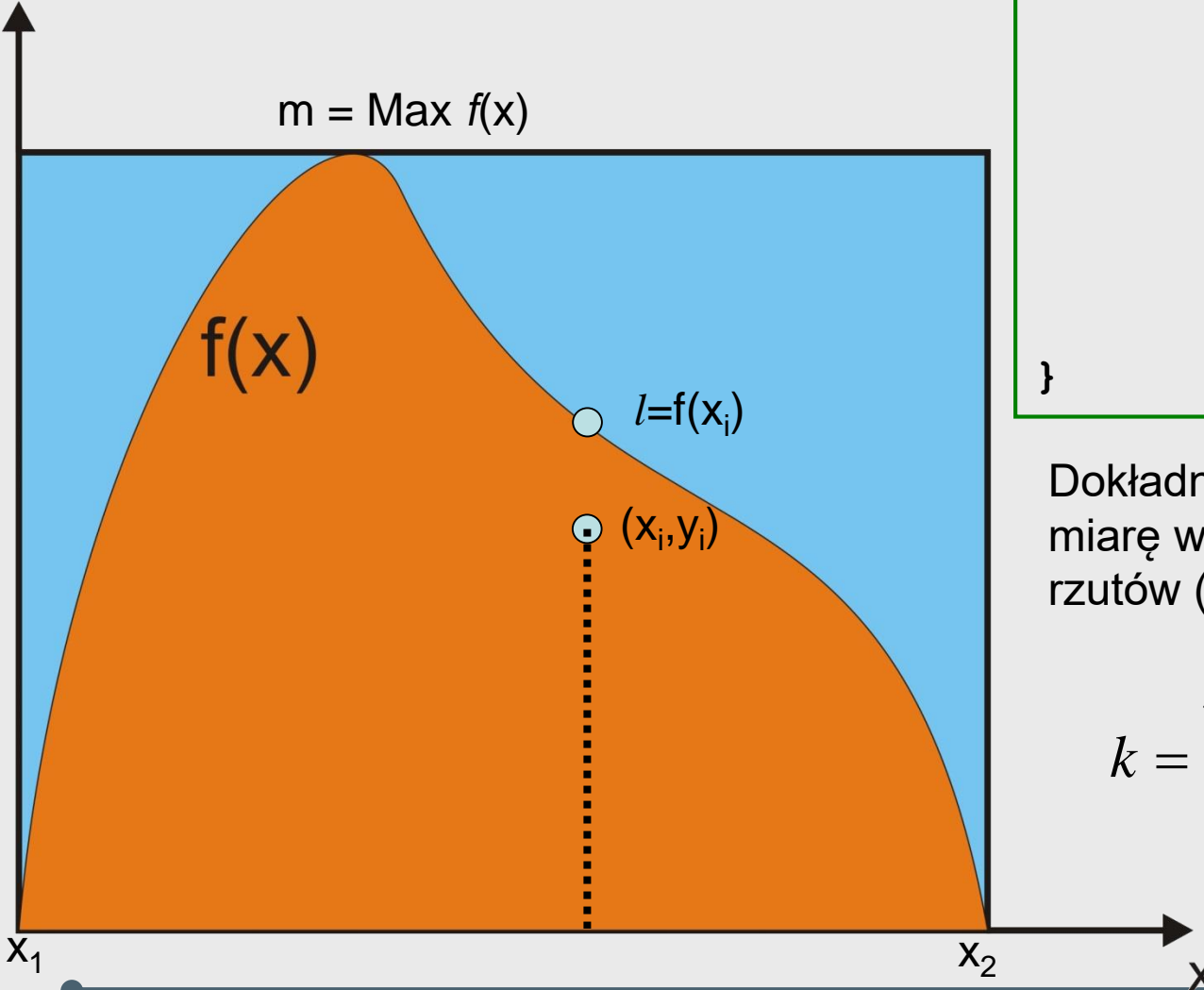


$$k = \int_{x_1}^{x_2} f(x) dx$$



Pole obszaru: $P=(x_2-x_1)*m$

$m = \text{Max } f(x)$



```
for (i=0; i<n;i++) {  
    xi = rand()  
    yi = rand()  
    l=f(xi)  
  
    if (yi <= l) {  
        count++  
    }  
}
```

Dokładność obliczeń wzrasta w miarę wzrostu ilości rzutów (losowań) n

$$k = \int_{x_1}^{x_2} f(x) dx = \frac{\text{count}}{n} \cdot P$$



**Kluczowy dla dokładności i poprawności metody Monte Carlo
jest generator liczb losowych**





Przykładowe algorytmy należące do grupy metod Monte Carlo to:

- Bezpośrednia metoda MC,
- Dynamiczna i Kinematyczna metoda MC,
- Kwantowa metoda MC,
- metoda quasi-MC,
- łańcuchy Markowa,
- metoda Metropolis,
- itp.

Metody te znalazły szerokie spektrum zastosowania np. do symulacji zjawisk związanych z:

- fizyka kwantowa,
- dynamiką molekularną,
- genetyką,
- ekonomią itp.

W latach 80 metoda Monte Carlo zaczęła być również szeroko wykorzystywana w inżynierii materiałowej i metalurgii do symulacji np. kinetyki wzrostu ziaren w materiałach jedno fazowych, dwufazowych oraz kompozytach, do symulacji zjawisk rekrytalizacji statycznej i dynamicznej czy też do symulacji procesów spiekania.





Model Potts

Model Potts nazywany w literaturze również modelem q-Potts jest powszechnie wykorzystywany do symulacji zbiorowego zachowania się struktur komórkowych.

Obliczenia prowadzone są w zdefiniowanej przestrzeni o regularnym charakterze komórkowym w której każdy element S_i może przyjmować pewną liczbę stanów q .

Dodatkowo każdy element S_i posiada określoną liczbę sąsiadów S_j , $j=1, n$ np. sześciu ($n=6$) lub ośmiu sąsiadów ($n=8$)

Hamiltonian takiego układu zdefiniowano w sposób następujący:

$$H = K \sum_{\langle i, j \rangle} (1 - \delta_{S_i S_j})$$

$\langle i, j \rangle$ analizowana para sąsiednich elementów $\delta_{S_i S_j}$ delta Kroneckera

$$\delta_{S_i S_j} = \begin{cases} 1 \Leftrightarrow i = j \\ 0 \Leftrightarrow i \neq j \end{cases}$$



Ewolucja tak zdefiniowanego układu odbywa się z wykorzystaniem np. **algorytmu Metropolis**. Standardowy algorytm składa się z trzech kroków:

Krok 1: Wybór elementu S_i oraz określenie charakteru zmiany jego stanu q np. zmiana przynależności do ziarna,

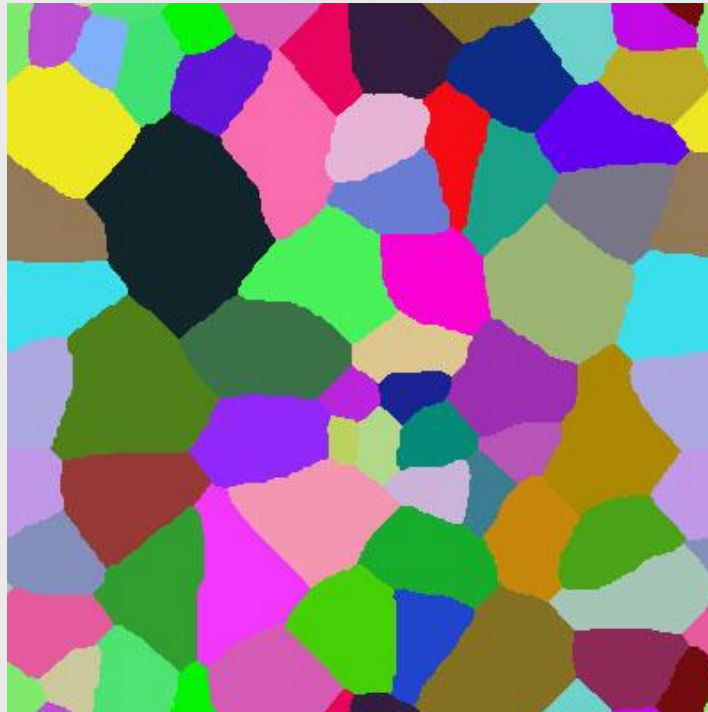
Krok 2: Określenie wartości Hamiltonianu lub energii które są wykorzystywane do zaakceptowania proponowanej zmiany lub też jej odrzuceniu,

Krok 3: Inne akcje nie wymienione w krokach 1 i 2.

W takim ujęciu algorytm q -Potts posiada wiele cech wspólnych w metodą automatów komórkowych



Działanie modelu Potts sprężonego z algorytmem Metropolis można prześledzić na przykładzie symulacji zmian mikrostruktury związanych z rozrostem ziaren, rekrytalizacją dynamiczną czy statyczną.



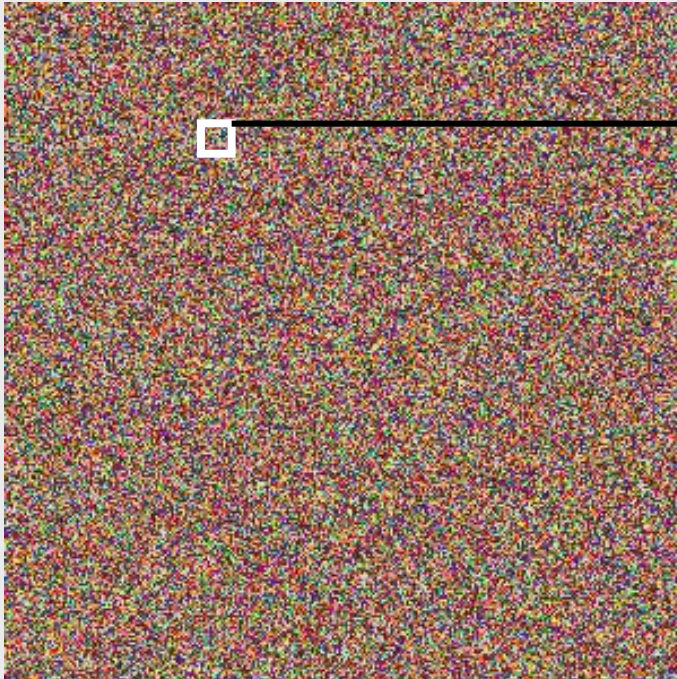


Założenia modelu MC

$n = 50$



Liczba stanów



1 MCS

Q1	Q2	Q3	Q4	Q5
Q6	Q7	Q8	Q9	Q10
Q11	Q6	Q12	Q13	Q14
Q15	Q16	Q17	Q6	Q18
Q19	Q20	Q21	Q22	Q6

Zbiór stanów komórek – przynależność do ziarna

$$\Omega = \{Q_0, \dots, Q_{n-1}\}$$



Etapy algorytmu:

Krok 1: Losowa selekcja elementu o danej orientacji.

Q1	Q1	Q2
Q3	Q3	Q2
Q3	Q2	Q2

Krok 2: Określenie energii sieci otaczającej rozpatrywany element Q_i . Energia określana jest z wykorzystaniem standardowej formuły:

$$E = J_{gb} \sum_{\langle i,j \rangle} (1 - \delta_{S_i S_j})$$



delta Kroneckera

współczynnik określający energie granicy ziarna

analizowana para sąsiednich elementów

Krok 3: Przypisanie rozpatrywanemu elementowi nowego stanu. Orientacja ta wybierana jest losowo ze zbioru Ω dostępnych orientacji.

Q1	Q1	Q2
Q3	Q4	Q2
Q3	Q2	Q2

Krok 4: Określenie zmiany energii sieci otaczającej rozpatrywany element Q_i , spowodowanej modyfikacją stanu.



2	2	2
3	2	4
3	3	3



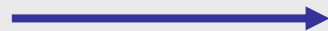
$$E=5$$

2	2	2
3	3	4
3	3	3



$$E=4$$

2	2	2
3	4	4
3	3	3



$$E=7$$



2	2	2
3	14	4
3	3	3

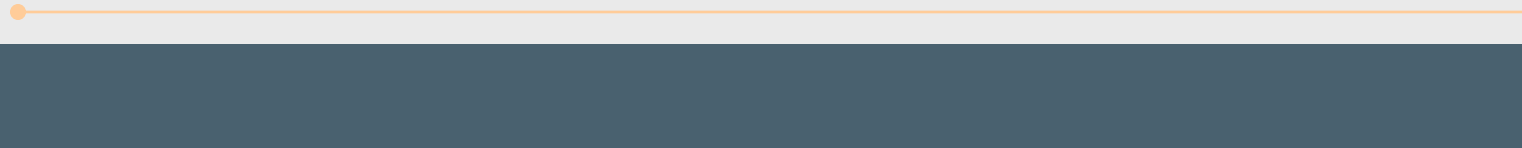


$$E=8$$

2	2	2
3	6	4
3	3	3



$$E=8$$



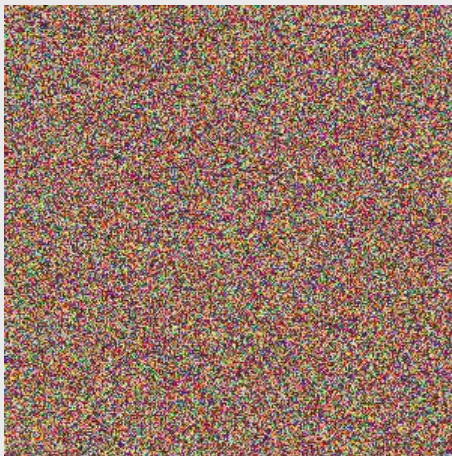
Q1	Q1	Q2
Q3	Q3	Q2
Q3	Q2	Q2

Q1	Q1	Q2
Q3	Q4	Q2
Q3	Q2	Q2

Krok 5: Zaakceptowanie nowej orientacji z prawdopodobieństwem p :

$$p(\Delta E) = \begin{cases} 1 & \Delta E \leq 0 \\ 0 & \Delta E > 0 \end{cases}$$

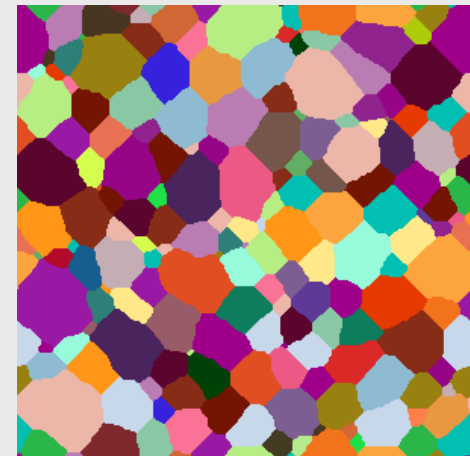
Q = 50



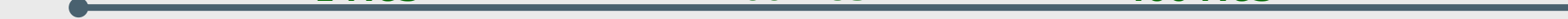
1 MCS

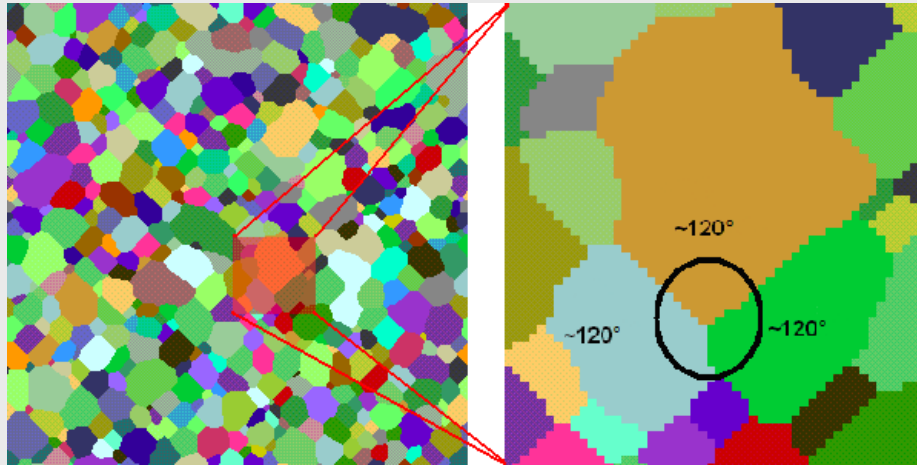


100 MCS

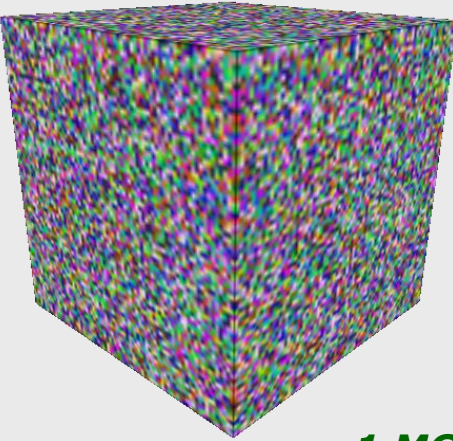


400 MCS

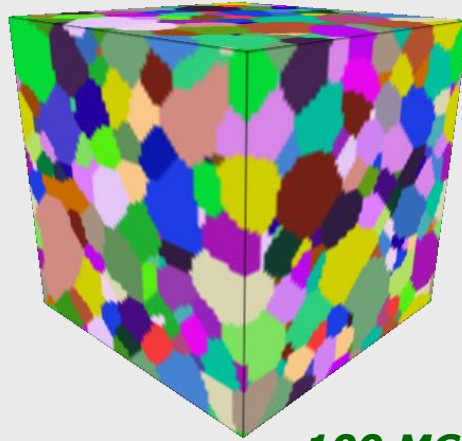




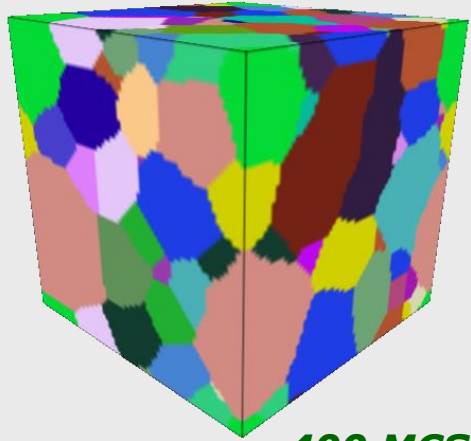
$n = 50$



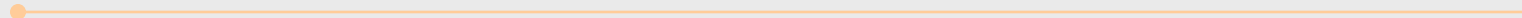
1 MCS



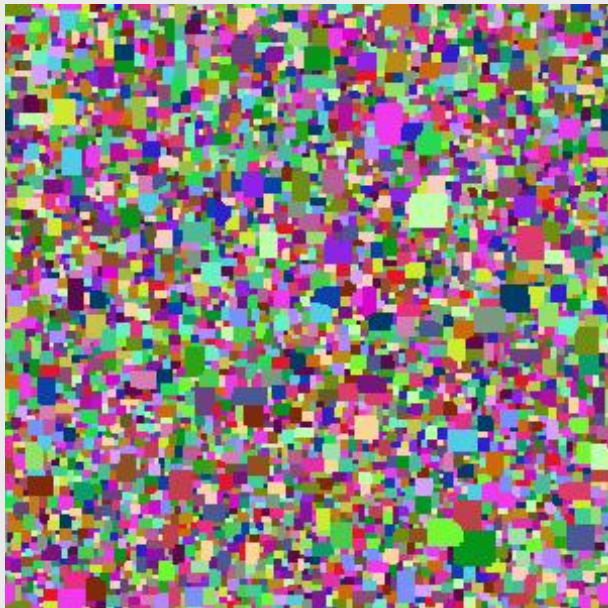
100 MCS



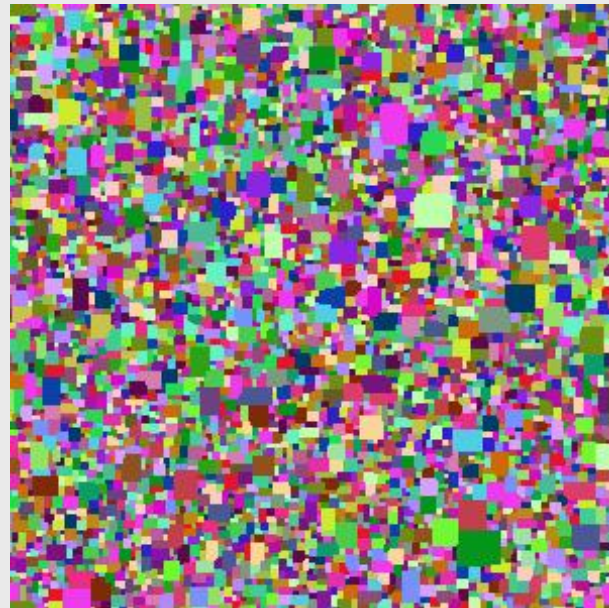
400 MCS



Efekt zatrzymania rozrostu

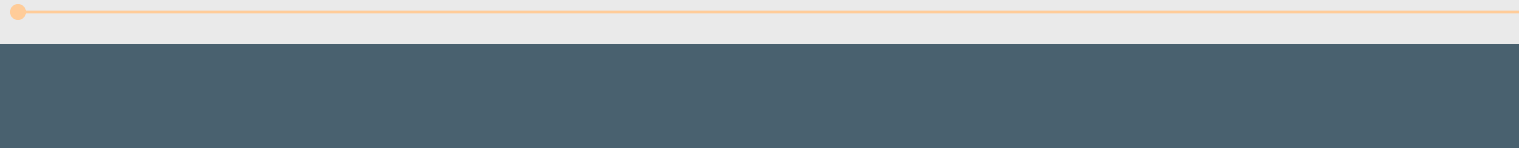


200 MCS



400 MCS





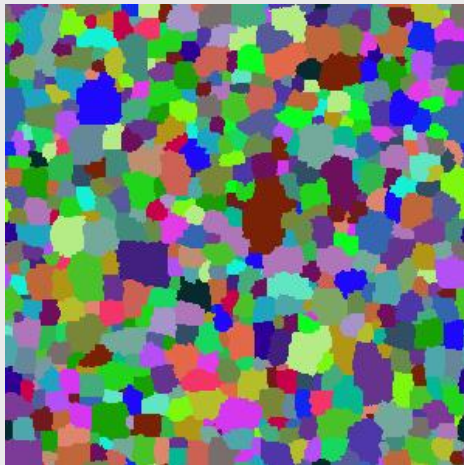
Krok 5: Zaakceptowanie nowej orientacji z prawdopodobieństwem p :

$$p(\Delta E) = \begin{cases} 1 & \Delta E \leq 0 \\ \exp\left(-\frac{\Delta E}{kT}\right) & \Delta E > 0 \end{cases}$$

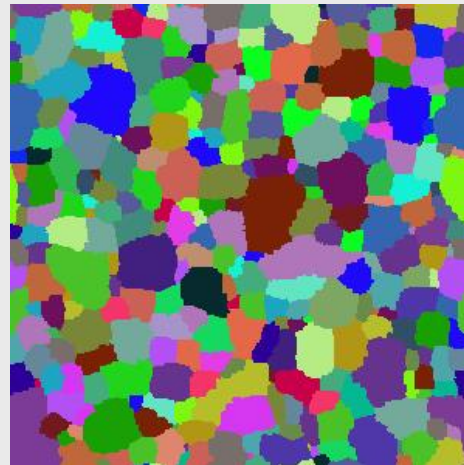
stała Boltzmana

temperatura symulacji (współczynnik modelu MC)

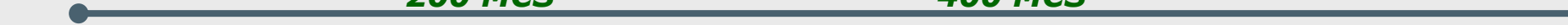
$kT=0.6$



200 MCS

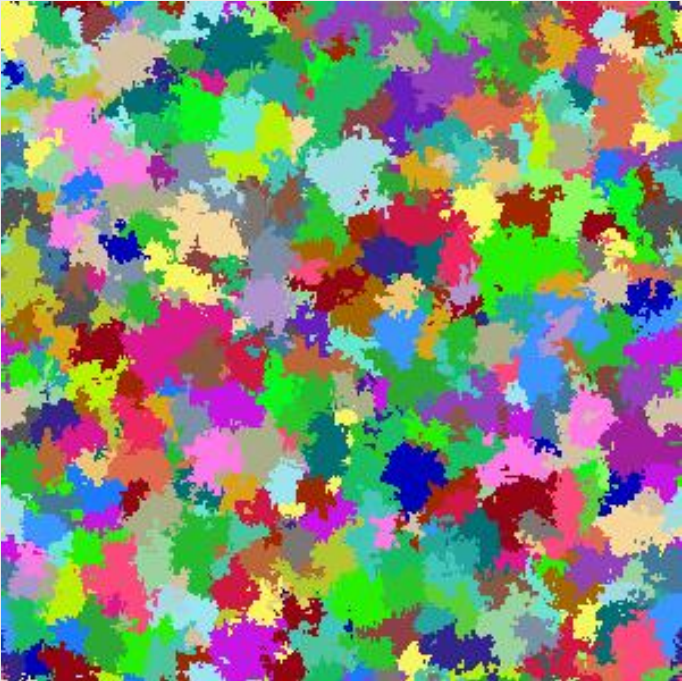


400 MCS

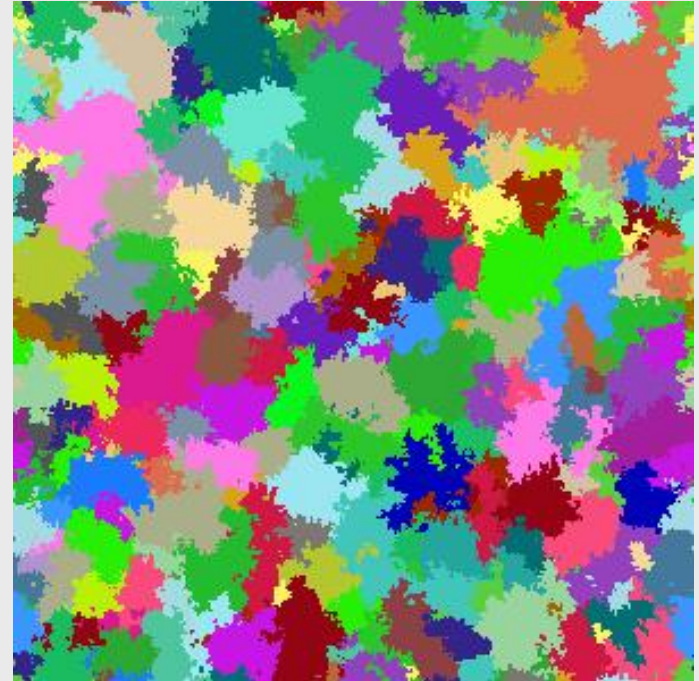




$kT=6$



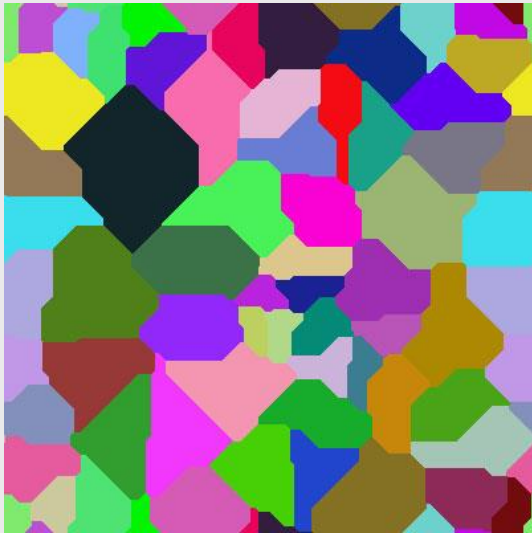
200 MCS



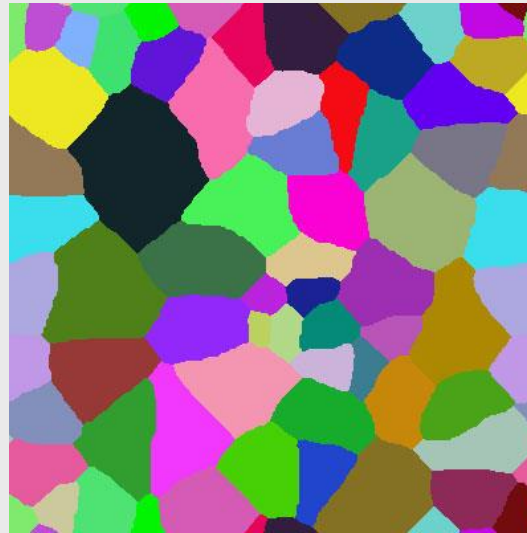
400 MCS



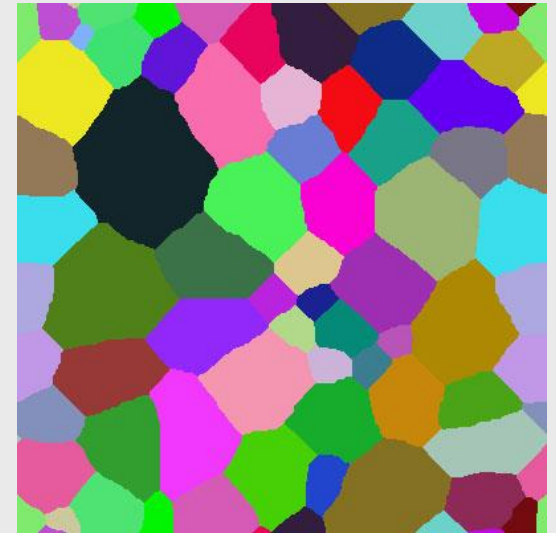
Cellular Automata + Monte Carlo



Dane wejściowe



100 MCS



200 MCS

$$p(\Delta E) = \begin{cases} 1 & \Delta E \leq 0 \\ \exp\left(-\frac{\Delta E}{kT}\right) & \Delta E > 0 \end{cases}$$

$$E = J_{gb} \sum_{\langle i,j \rangle} (1 - \delta_{S_i S_j})$$



Podstawową wadą opisanego algorytmu MC jest bardzo **długi czas obliczeniowy**. Jest to związane z probabilistycznym charakterem modelu.

Większość z proponowanych reorientacji elementów prowadzi do wzrostu energii sieci, a zatem zmiana nie jest akceptowana i nie mamy do czynienia z rozrostem ziarna.

Ten problem jest jeszcze większy w przypadku gdy wzrasta liczba ziaren (liczba możliwych rotacji q) oraz w przypadku symulacji 3D.

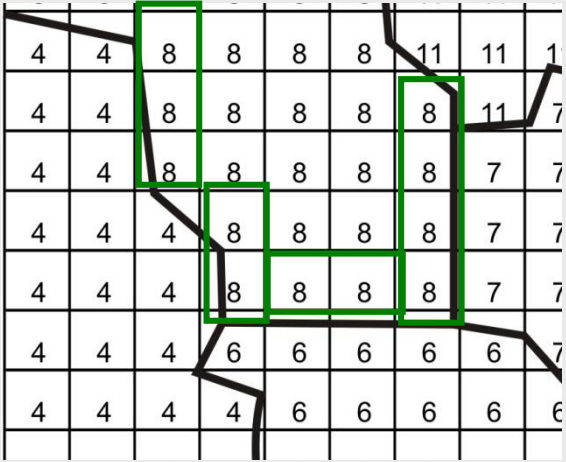
Modyfikacje modelu



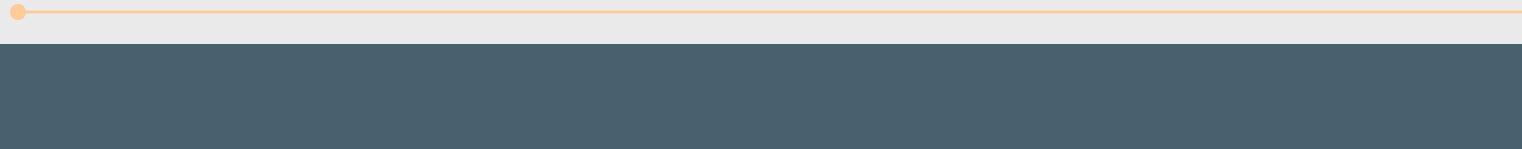
Ruch granicy ziarna zachodzi na drodze przeskoku atomu z jednego ziarna do drugiego tylko granica ziarna bierze udział w migracji.



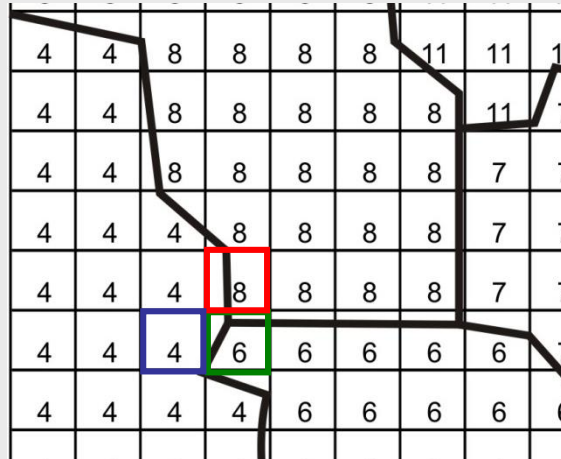
Modyfikacja zakłada podzielenie elementów na dwie grupy: pierwsza to elementy leżące na granicy ziarna, druga to elementy tworzące wnętrze ziarna.



Algorytm Metropolis jest zatem wykorzystywany do analizy zmiany orientacji tylko elementów pierwszej grupy.



Kolejna modyfikacja zakłada generowanie nowej orientacji rozpatrywanego elementu nie z pośród q dostępnych orientacji, ale tylko z orientacji sąsiadujących ziaren.



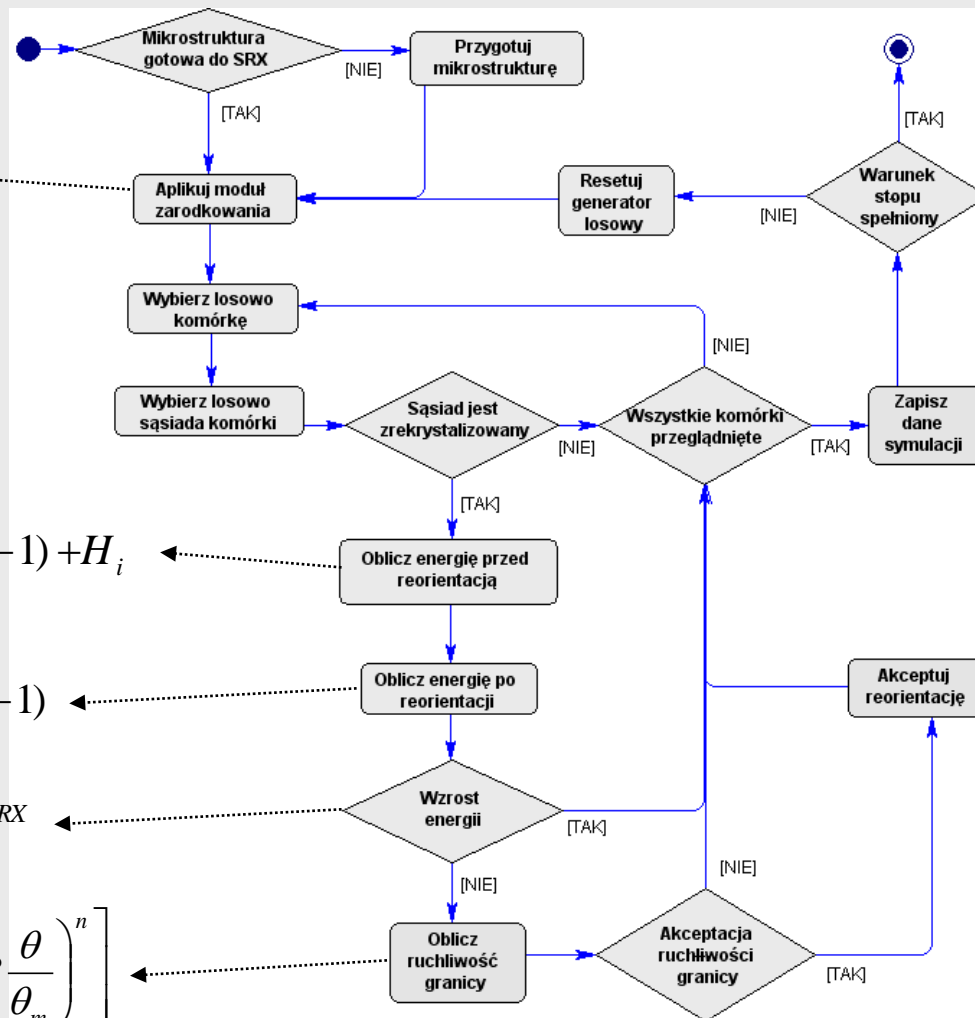
Kolejna modyfikacja zakłada iż element który został losowo wybrany w danym kroku MC z pośród N dostępnych elementów, nie będzie mógł być wylosowany po raz drugi. Zatem losowanie kolejnego elementu odbywa się ze zbioru $(N-k)$, gdzie k – liczba wylosowanych elementów.





Model rekrytalizacji statycznej na bazie metody Monte Carlo

$N = N_{wysycenia}$
 $N = N_{stale}$
 $N = s \text{ MCS}$
 $N = N_{init} - s \text{ MCS}$



$$E_i^{preSRX} = -J \sum_{j=1}^Z (\delta_{S_i S_j} - 1) + H_i$$

$$E_i^{postSRX} = -J \sum_{j=1}^Z (\delta_{S_i S_j} - 1)$$

$$\Delta E_i = E_i^{preSRX} - E_i^{postSRX}$$

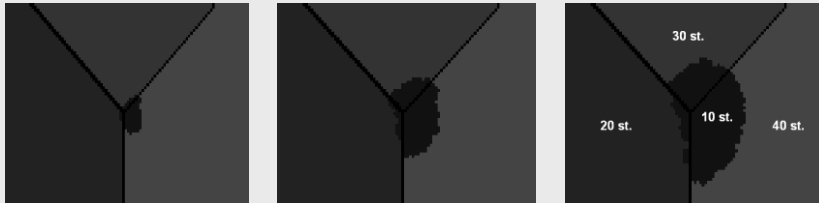
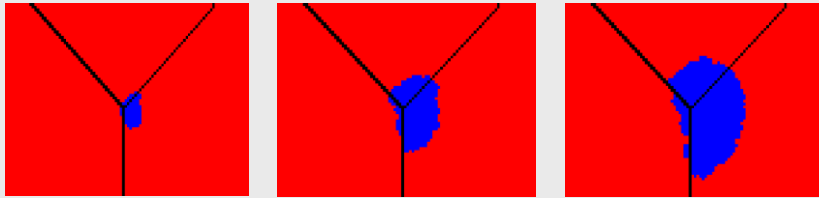
$$M(\theta) = M_m \left[1 - \exp\left(-B \frac{\theta}{\theta_m}\right)^n \right]$$

kinetyka rekrytalizacji				Energia zgmagazynowana	
t	F	ln[1/(1-F)]	t	Energia zgmagazynowana:	
0	0,002322	0,002325	0	448955,0	
1	0,002600	0,002603	1	448930,0	
2	0,002967	0,002971	2	448665,0	
3	0,003389	0,003395	3	448475,0	
4	0,004233	0,004242	4	448095,0	
5	0,005300	0,005314	5	447815,0	
6	0,006700	0,006723	6	446985,0	
7	0,008444	0,008480	7	446200,0	
8	0,010322	0,010376	8	445355,0	
9	0,012544	0,012624	9	444355,0	
10	0,014978	0,015091	10	443260,0	
11	0,017556	0,017711	11	442100,0	
12	0,020600	0,020815	12	440730,0	
13	0,023800	0,024088	13	439290,0	
14	0,027811	0,028205	14	437485,0	
15	0,032256	0,032787	15	435485,0	
16	0,036689	0,037379	16	433490,0	
17	0,041333	0,042212	17	431400,0	
18	0,045953	0,047022	18	429330,0	
19	0,051811	0,052002	19	426685,0	
20	0,058267	0,060033	20	423780,0	
21	0,064844	0,067042	21	420820,0	



Rozkład energii zmagazynowanej

homogeniczny



heterogeniczny

