Podstawy metody Monte Carlo i jej zastosowania w modelowaniu komputerowym

Plan wykładu

1 cegiełka

- elementy rachunku prawdopodobieństwa i statystyki w ekspresowym skrócie
  - zmienna losowa dyskretna → rozkład prawdopodobieństa
  - moment zmiennej losowej, moment centralny, wartość oczekiwana i wariancja
  - zmienna losowa ciągła → funkcja gęstości prawdopodobieństwa i dystrybuanta

2 cegiełka

- komputerowe generatory liczb pseudolosowych
  - rozkład jednorodny → generator liniowy
  - dowolny rozkład → odwracanie dystrybuanty, algorytm Metropolisa

3 cegiełka

- podstawowa metoda MC dla całkowania numerycznego
  - interpretacja wyniku → prawo wielkich liczb i CTG
  - algorytm MC całkowania

- przykłady modelowania MC
  - model Isinga
  - optymalizacja wartości funkcji → symulowane wyżarzanie

- podsumowanie

- literatura

Elementy rachunku prawdopodobieństwa i statystyki w ekspresowym skrócie

Zmienna losowa – to wynik zdarzenia losowego, który może mieć wymiar:

- liczbowy (wynik rzutu kostką, liczba zliczeń w detektorze, wartość zmiennej o rozkładzie ciągłym)

$$X \in \{1, 2, 3, 4, 5, 6\} \qquad X \in [a, b]$$

- logiczny/binarny (losowanie bitów 0-1, prawda/fałsz, rzut monetą)

$$X \in \{\text{orzeł, reszka}\}$$

$$X \in \{0, 1\} \qquad X \in \{\text{prawda, fałsz}\}$$

- nieliczbowy (losowanie karty z tali, kolorów), ale z możliwością przypisania wynikowi unikatowej wartości liczbowej (1,2,..)

$$X \in \{\blacksquare, \blacksquare, \blacksquare, \blacksquare, \ldots\} \quad \Longrightarrow \quad \{1, 2, 3, \ldots\}$$

$$X \in \{2\clubsuit, 3\clubsuit, 4\clubsuit, \ldots\} \quad \Longrightarrow \quad \{1, 2, 3, \ldots, \}$$

Zazwyczaj interpretacja pojedynczego wyniku zdarzenia losowego nie ma znaczenia. Istotny (statystycznie) jest opis wyniku serii zdarzeń losowych, do tego wykorzystamy narzędzia jakie daje nam statystyka - wnioskowanie na podstawie informacji zawartych w generowanej przy użyciu Monte Carlo próbce danych.

Zjawisko losowe – proces stochastyczny generujący ciąg zmiennych losowych $\{X_1, X_2, X_3, ...\}$, których wartości nie możemy przewidzieć, ale możemy przypisać im określone prawdopodobieństwo wystąpienia $\{p_1, p_2, p_3, ...\}$.

$$\underbrace{X_1}_{p_1}, \underbrace{X_2}_{p_2}, \underbrace{X_3}_{p_3}, \ldots, \underbrace{X_n}_{p_n} \qquad P\{X_i\} = p_i$$

Prawdopodobieństwo p_i jest granicą, do której dąży asymptotycznie częstość wystąpień zmiennej X_i

$$\lim_{N\to\infty} \frac{n_i}{N} = p_i$$

Uwaga: należy pamiętać o różnicy pomiędzy tymi wielkościami

- prawdopodobieństwo jest określone w sposób ścisły (dokładny)
- częstość wystąpień to zmienna losowa i podlega pewnym fluktuacjom statystycznym
- równość tylko w granicy, której nigdy na komputerze nie osiągniemy

Rozkład dyskretny zmiennej losowej

- prawdopodobieństwo zdarzenia losowego ma określone własności

$$0 \leq p_i \leq 1 \qquad \begin{aligned} p_i = 0 &\Longrightarrow \quad \text{zdarzenie nigdy nie zajdzie} \\ p_i = 1 &\Longrightarrow \quad \text{zdarzenie pewne} \end{aligned}$$

- prawdopodobieństwo otrzymania jakiegokolwiek wyniku losowego (**warunek normalizacji prawdopodobieństwa**)

$$\sum_{i=1}^{N} p_i = 1$$

- w metodzie MC musimy zapewnić spełnienie warunku normalizacji aby wyniki miały prawidłową interpretację ilościową (porównanie różnych wyników)

- prawdopodobieństwo zdarzenia losowego a prawdopodobieństwo jego braku (zdarzenie przeciwne)

$$P\{X_i\} = p_i \qquad P\{\overline{X}_i\} = 1 - p_i$$

- istotna własność wykorzystywana przy podejmowaniu decyzji w MC

Wielkości charakterystyczne dla rozkładu –> **momenty rozkładu**

- n-ty moment

$$E(X^n) = \langle X^n \rangle \equiv \sum_i p_i \, x_i^n$$

W metodzie MC najczęściej interesują nas 2 pierwsze momenty

- **wartość oczekiwana** (przeciętna) zmiennej losowej – 1 moment

$$E(X) = \langle X \rangle = \sum_i p_i \, x_i = \mu$$

- **wariancja** – to kombinacja 1 i 2 momentu rozkładu

$$var\{X\} \equiv \langle (X-\mu)^2 \rangle = \sum_i p_i (x_i - \mu)^2 = \langle X^2 \rangle - \langle X \rangle^2$$

odchylenie standardowe – to miara rozrzutu zmiennej losowej wokół wartości oczekiwanej

$$\sigma_X = \sqrt{\langle X^2 \rangle - \langle X \rangle^2}$$

- jeśli zmienna losowa jest argumentem funkcji to wartości funkcji automatycznie stają się zmiennymi losowymi

$$g = g(x_i), \qquad x_i \in \{X_1, X_2, \ldots, X_N\}$$

a skoro tak, to identycznie możemy określić wartość oczekiwaną i wariancję (odchylenie standardowe)

$$\langle g(X) \rangle = \sum_i p_i g(x_i)$$

$$var\{g(X)\} = \langle g^2(X) \rangle - \langle g(X) \rangle^2$$

Uwaga: w prawie każdej wersji metody Monte Carlo, jedną z głównych czynności jest szacowanie wartości oczekiwanej i wariancji

Dystrybuanta i funkcja gęstości prawdopodobieństwa

Dla rozkładu dyskretnego definiowaliśmy

$$X \in \{X_1, X_2, \ldots, X_n\} \qquad P\{X_i\} = p_i$$

możemy zatem określić prawdopodobieństwo wylosowania zmiennej z podzbioru

$$P\{X \in \{X_1, X_2, \ldots, X_k\}\} = P\{X \leq X_k\} = p_1 + p_2 + \ldots + p_k = \sum_{i=1}^{k} p_i, \qquad k \leq n$$

funkcję określającą tak skumulowane prawdopodobieństwo nazywamy **dystrybuantą rozkładu**

$$\underbrace{F(x_k) \equiv P\{X \leq X_k\}}_{\text{dyskretny}} \qquad \vee \qquad \underbrace{F(x_k) \equiv P(X \leq x_k)}_{\text{ciagły}}$$

Własności dystrybuanty:

$F(x) \in [0, 1]$ - warunek normalizacji (określa prawdopodobieństwo)

$F(x + \delta x) \geq F(x), \quad \delta x > 0$ - funkcja niemalejąca

Przykład: rozkład Bernoulliego

Rozkład Bernoulliego opisuje proces, którego wynik może przyjmować jedynie dwie wartości (np.: rzut monetą, przejście cząstki przez tarczę)

$$X \in \{0,1\} \qquad (X \in \{a,b\})$$

obu zmiennym możemy przypisać prawdopodobieństwo wylosowania

$$P\{0\} = 1 - p = q, \qquad P\{1\} = p, \qquad p + q = 1$$

wówczas **rozkład prawdopodobieństwa** opisuje

$$p_r(n) = p^n (1-p)^{1-n}, \qquad n = 0,1$$

Wartość oczekiwana zmiennej (pierwszy moment) $\langle X \rangle = (1-p)\cdot 0 + p \cdot 1 = p$

drugi moment $\langle X^2 \rangle = p$

wariancja $var\{X\} = \langle X^2 \rangle - \langle X \rangle^2 = p - p^2 = p(1-p)$

Otrzymaliśmy wynik analityczny, jak proces losowania zmiennej o takim rozkładzie zasymulować na komputerze? Potrzebujemy dodatkowych narzędzi:

- miary prawdopodobieństwa (dystrybuanta + funkcja gęstości prawdopodobieństwa)
- generatora liczb pseudolowych o rozkładzie Bernoulli'ego

- dla zmiennej o ciągłym rozkładzie prawdopodobieństwa sumę zastępujemy całkowaniem, dyskretny rozkład prawdopodobieństwa zastępujemy ciągłą funkcją **f(x)**

$$P(x \leq x_k) = F(x_k) = \int_{-\infty}^{x_k} f(x)dx \leq 1$$

- **funkcja gęstości prawdopodobieństwa f(x)** (fgp) jest nieujemna i unormowana

$$f(x) \geq 0 \qquad \int_{-\infty}^{\infty} f(x)dx = 1$$

- fgp wykorzystujemy do wyznaczenia 1 i 2 momentu rozkładu

$$\langle X \rangle = \int_{-\infty}^{\infty} x f(x)dx$$

$$\langle X^2 \rangle = \int_{-\infty}^{\infty} x^2 f(x)dx$$

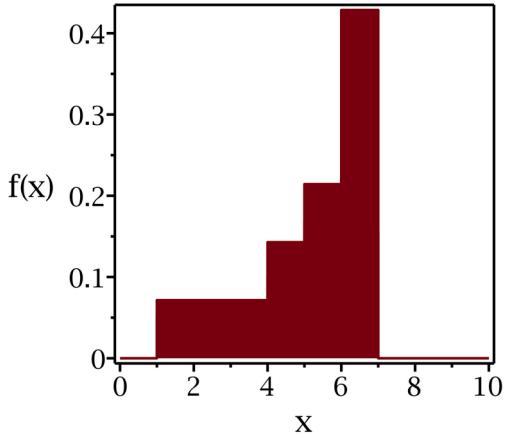
$$var\{X\} = \langle X^2 \rangle - \langle X \rangle^2$$

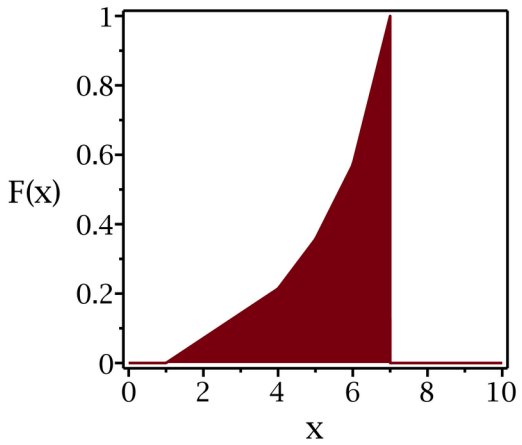
Dystrubuantę i fgp wykorzystujemy też do konstruowania generatorów liczb pseudolosowych – o tym za chwilę.

- przykład fgp (górny wiersz) i dystrybuanty (dolny wiersz)

rozkład schodkowy

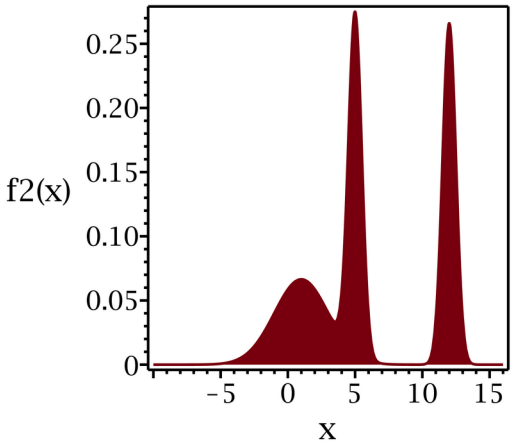
$x \in [1, 7]$

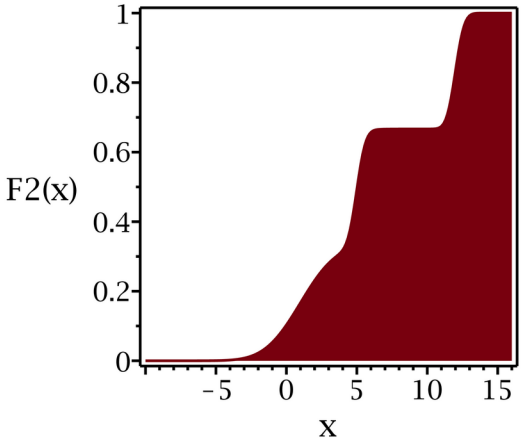




rozkład nieograniczony

$x \in (-\infty, \infty)$





Komputerowe generatory liczb pseudolosowych

Metoda MC wymaga uzycia ciągu liczb o zadanym rozkładzie – te uzyskujemy z generatorów liczb pseudolosowych. Liczby generowane są według określonego algorytmu – dlatego pseudolosowe – jeśli znamy algorytm to możemy określić każdą kolejną liczbę (i gdzie tu losowość?).

Zalety generatorów komputerowych:

- proste w obsłudze
- bardzo szybkie
- dowolny rozkład prawdopodobieństwa
- ciągi liczb o bardzo dużym okresie
- powtarzalność ciągów (cecha istotna przy testowaniu oprogramowania)

Wady:

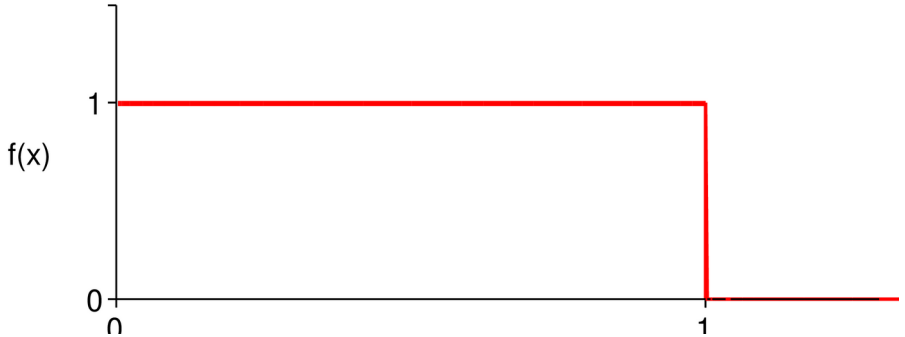
- pseudolosowość a nie prawdziwa losowość
- generatory spełniają tylko część testów statystycznych (nie wszystkie!)
- korelacja elementów w ciągu (np. algorytm Metropolisa, w mniejszym stopniu też inne)

Wady generatorów akceptujemy, bo nie mamy nic lepszego – generatory liczb losowych wykorzystujące zjawiska fizyczne (rozpad promieniotwórczy czy szum termiczny aparatury elektronicznej) są niezwykle kłopotliwe w użyciu.

Proces generowania liczb pseudolosowych zależy od typu żądanego rozkładu.

- podstawą każdego generatora o dowolnym rozkładzie prawdopodobieństwa jest
  - **generator o rozkładzie jednorodnym U(0,1)**

$$f_U(x) = \begin{cases} 1 & x \in (0,1) \\ 0 & x \notin (0,1) \end{cases}$$



- do generowania liczb o dowolnym rozkładzie prawdopodobieństwa używamy **U(0,1)** oraz
  - **metody odwracania dystrybuanty** (najszybsza)
  - **algorytm Metropolisa**
  - **metody eliminacji von Neumanna**
    (wolna – jeśli nie musimy to nie stosujemy)

Generator liczb pseudolosowych o rozkładzie jednorodnym

- najczęściej stosowanym generatorem jest generator liniowy
- zasada działania bazuje na dzieleniu modulo
- kolejny element wyznaczamy na podstawie jednego lub kilku poprzednich elementów ciągu (liczby naturalne)

$$x_i = (a_1 x_{i-1} + a_2 x_{i-2} + \ldots + a_n x_n + c) \mod m$$

$\{a_1, a_2, \ldots, a_n, c, m\}$ - parametry generatora, m - duża liczba pierwsza

$\{x_1, x_2, \ldots, x_n\}$ - ciąg inicujący („ziarno” generatora)

- generator dostarcza liczby całkowite z zakresu

$$x_i \in [1, m-1]$$

Uwaga: pomijamy 0 – odwracając liczbę unikamy osobliwości

należy je znormalizować (unormować)

$$X_i = \frac{x_i}{m} \quad \rightarrow \quad X_i \in (0,1) \quad \rightarrow \quad X_i \sim U(0,1)$$

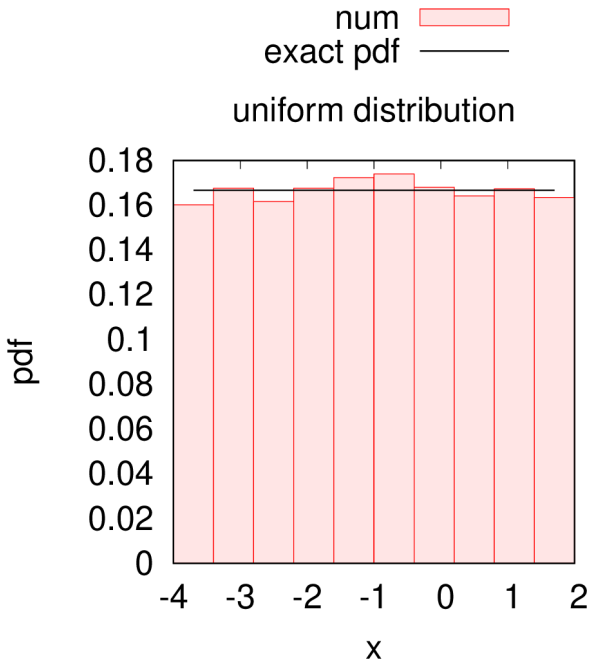
- następnie możemy wykonać transformację dla innego zakresu np. Y=[a,b]

$$Y_i = a + X_i(b-a) \quad \rightarrow \quad Y_i \sim U(a,b)$$

Przykład generowania N liczb o rozkładzie **U(a,b)** w języku C.

- funkcja **rand()** generuje całkowite nieujemne liczby pseudolosowe o rozkładzie jednorodnym
- funkcja **srand(unsigned int)** ustawia punkt startowy ciągu
- tablica **hist[]** przechowuje informacje o ilości liczb wylosowanych z danego przedziału (N-normalizacja dla całkowitej liczby zliczeń, Δx-normalizacja jak dla fgp)

```
unsigned int z0
inicjalizacja:   a,b,N,z0,k

      srand(z0)    //ziarno podajemy opcjonalnie
      Δx = (b-a)/k
      hist[]=0

for(i=1; i <= N; i++){
      x=(double)rand()/RAND_MAX     //x ~ U(0,1)
      x=a+(b-a) x                   //x ~ U(a,b)
      l=int( (x-a)/Δx )
      hist[l]+=1/N/Δx               //histogram

}
```



Oczekujemy, że generator dostarcza liczb „prawie” losowych – czy tak rzeczywiście jest?
Generator musi spełniać zestaw testów statystycznych – w przeciwnym wypadku czeka nas niemiła niespodzianka.

Przykład „słabego” generatora – generator multiplikatywny (c=0) RANDU – test wizualny (najprostszy)

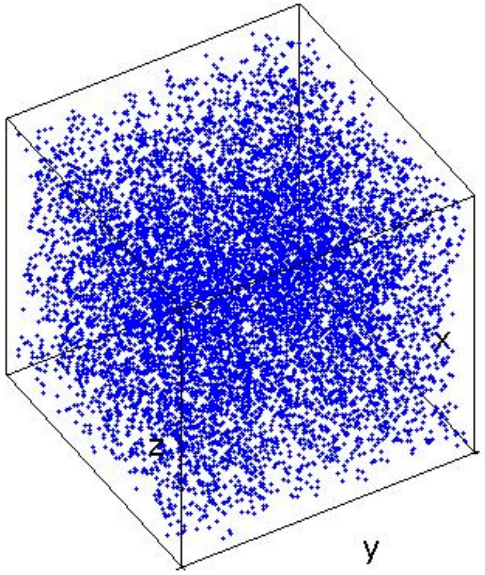
$$x_{i+1} = a x_i \mod m \qquad a = 65539 = 2^{16} + 3 \qquad m = 2^{31}$$

$$x_{i+2} = a x_{i+1} \mod m = (2^{16}+3) x_{i+1} \mod m = (2^{16}+3)^2 x_i \mod m$$

$$x_{i+2} = 6x_{i+1} - 9x_i$$ **- silna zależność (korelacja) trzech kolejnych liczb**

Silna korelacja elementów ciągu powoduje że układają się one na hiperpowierzchniach
- generowane punkty próbkują w sposób wybiórczy przestrzeń.

przypadkowy rzut wygląda dobrze

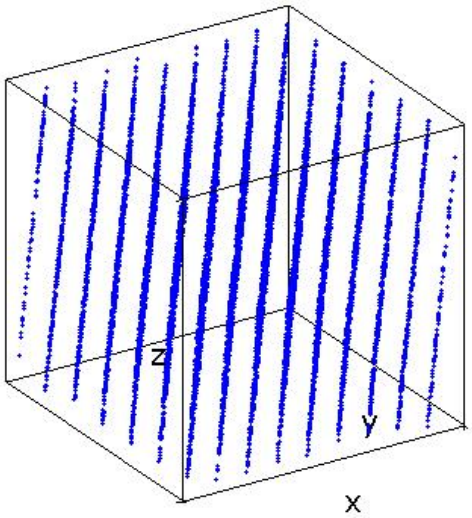


10 tysięcy punktów wygenerowanych przez RANDU

rysujemy punkty

$[x_i, x_{i+1}, x_{i+3}]$

… ale po obrocie odkrywamy, że wyniki nie są takimi jakich oczekujemy – widać hiperpłaszczyzny



na hiperpłaszczyznach losowość jest, ale między nimi już nie

Rozkład dowolny - metoda odwracania dystrybuanty

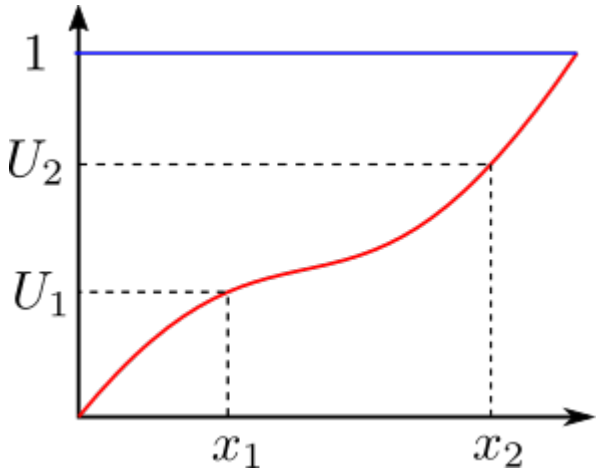
Wykorzystujemy tu własność dystrybuanty

$$F(x) \in [0,1]$$

Losując zmienną U_i z rozkładu U(0,1) określamy losową wartość dystrybuanty

$$U_i = F(x_i) \in (0,1)$$

$$x_i = F^{-1}(U_i), \quad x \in (-\infty, \infty)$$



Metoda w zasadzie sprowadza się do znalezienia funkcji odwrotnej:

1) jeśli ją znamy to problem jest rozwiązany
2) funkcję odwrtoną można przybliżać/aproksymować – wówczas rozwiązanie przybliżone

Sposób generowania zmiennych losowych x z rozkładu o dystrybuancie F sprowadza się do wylosowania ciągu liczb o rozkładzie jednorodnym i ich transformacji w elementy ciągu o docelowym rozkładzie

$$U_1, U_2, \ldots, U_n \in (0,1) \quad \rightarrow \quad X_1, X_2, \ldots, X_n \in (-\infty, \infty)$$

Metodę odwracania dystrybuanty można stosować do rozkładów ciągłych i dyskretnych

przykład – rozkład eksponencjalny

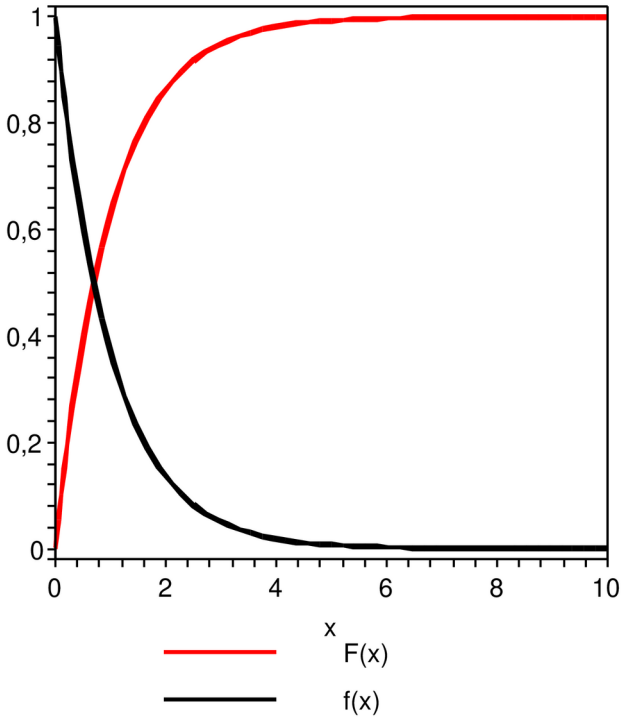
- funkcja gęstości prawdopodobieństwa

$$f(x) = \lambda e^{-\lambda x}, \quad x \in [0, \infty), \quad \lambda > 0$$

- dystrybuanta rozkładu

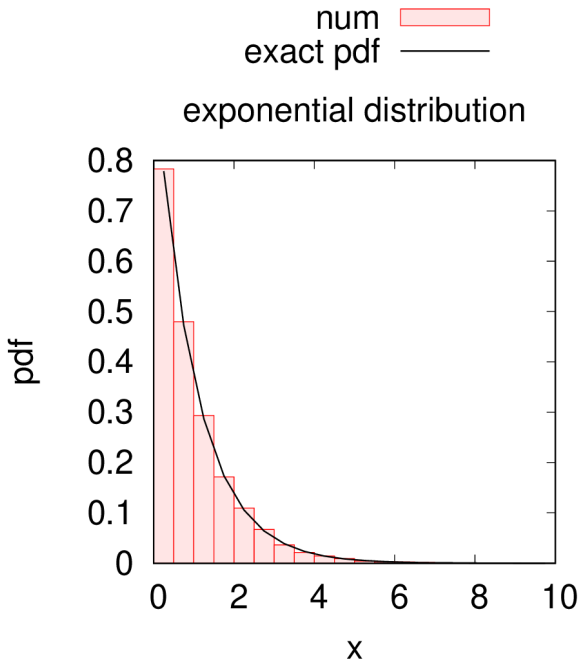
$$F(x) = \lambda \int_0^x dx' e^{-\lambda x'} = 1 - e^{-\lambda x} = U_1, \qquad U_1 \sim U(0,1)$$

$$x = -\frac{1}{\lambda} \ln(1 - U_1), \quad x \in (0, \infty)$$



Przykład generowania N liczb o rozkładzie Exp(λ) w języku C

```
unsigned int z0
inicjalizacja:  λ,N,z0,k

      srand(z0)    //ziarno podajemy opcjonalnie
      Δx = 10λ/k
      hist[]=0

for(i=1; i <= N; i++){
      U1=(double)rand()/RAND_MAX  //U1 ~ U(0,1)
      x = -ln(1-U1)/λ
      l=int( x/Δx )
      if(l<k) hist[l]+=1/N/Δx       //histogram
}
```

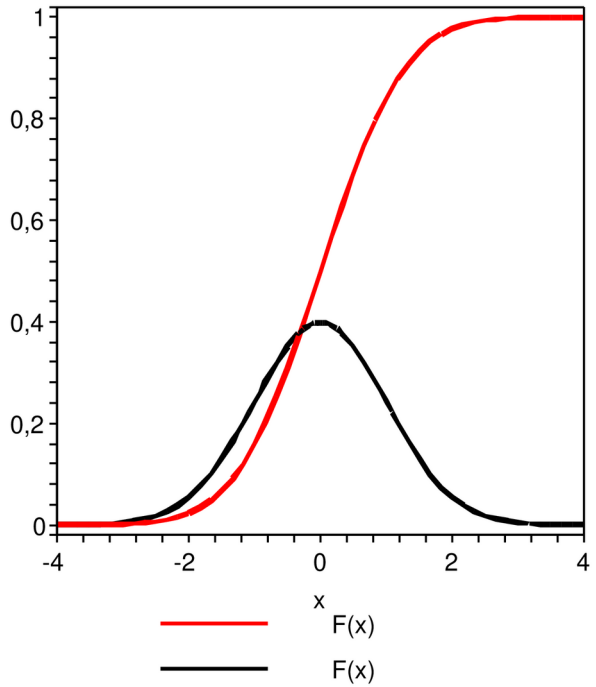


- konstruując histogram musimy pamiętać, że rozkład jest prawostronnie nieograniczony - część wylosowanych liczb znajdzie się poza zakresem obejmowanym przez histogram (te liczby można akumulować w ostatniej komórce – gdyby zaszła taka potrzeba)

przykład – rozkład normalny (Gaussa) N(0,1)

- funkcja gęstości prawdopodobieństwa

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



- nie znamy funkcji odwrotnej funkcji błędu erf() – problem w 1D
- zamiast tego rozważa się funkcję w 2D: **metoda Boxa-Mullera** generujemy od razu parę liczb o rozkładzie N(0,1)

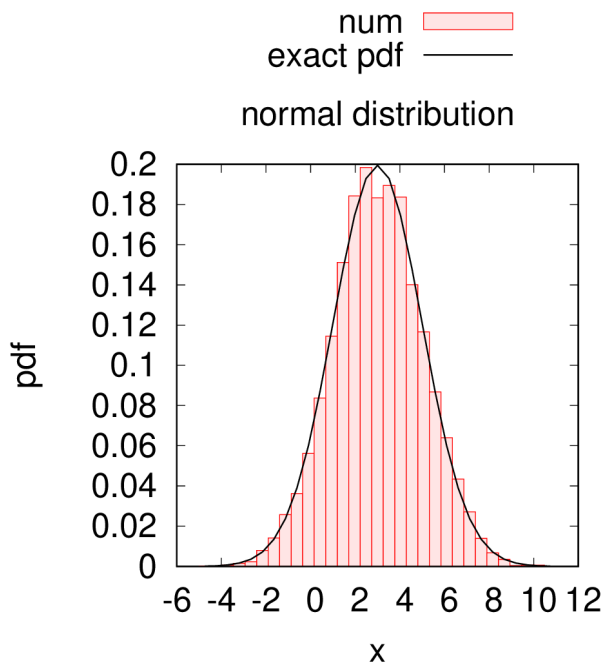
$$U_1,\, U_2 \sim U(0,1)$$

$$x_1 = r\cos(\phi) = \sqrt{-2\ln(1-U_1)}\cos(2\pi U_2)$$
$$x_2 = r\sin(\phi) = \sqrt{-2\ln(1-U_1)}\sin(2\pi U_2)$$

$$x_1,\, x_2 \sim N(0,1)$$

- transformacja dla docelowego rozkładu N(μ,σ)

$$y_i = \mu + \sigma \cdot x_i \quad \rightarrow \quad y_i \sim N(\mu,\sigma)$$



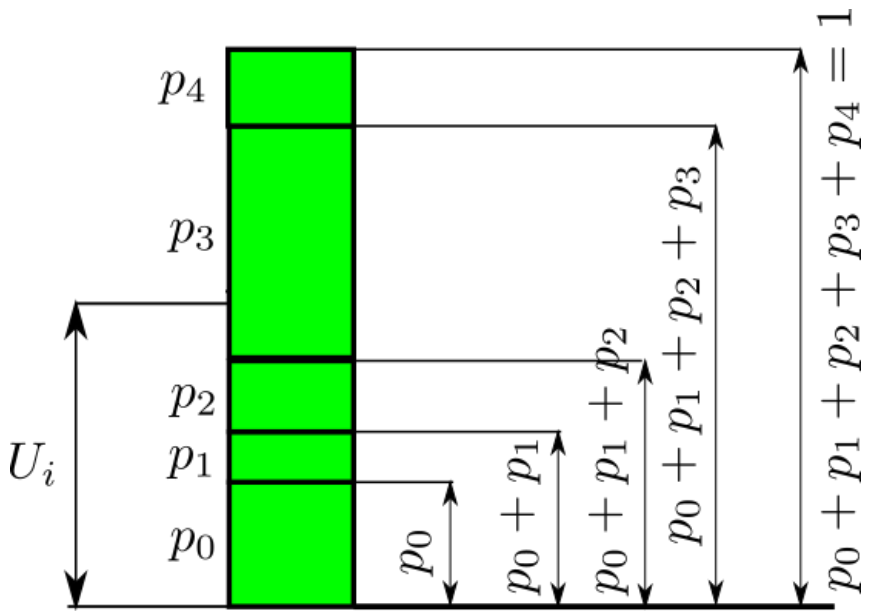
Metoda odwracania dystrybuanty dla rozkładu dyskretnego

przestrzeń dostępnych stanów:

$$\{x_0, x_1, x_2, x_3, x_4\}$$

$$\{p_0, p_1, p_2, p_3, p_4\}$$

$$\sum_i p_i = 1$$



$START \quad X = 0, \quad S = p_0, \quad U > S(?) \quad TAK$

$X = 0 + 1 = 1, \quad S = p_0 + p_1, \quad U > S(?) \quad TAK$

$X = 1 + 1 = 2, \quad S = p_0 + p_1 + p_2, \quad U > S(?) \quad TAK$

$X = 2 + 1 = 3, \quad S = p_0 + p_1 + p_2 + p_3, \quad U > S(?) \quad NIE \rightarrow STOP$

nowa zmienna: $x_i = 3$

- ten prosty algorytm jest skuteczny, ale jego konstrukcja wymaga sprawdzania warunku dla kolejnych podprzedziałów – idąc z dołu do góry
- wydajność algorytmu maleje w przypadku, gdy ciąg dyskretnych liczb jest duży np. kilkadziesiąt (losowanie kart w talii)

- **przykład – generowanie ciągu liczb o rozkładzie Bernoulli'ego**

$$P\{X=1\}=p \qquad \langle X\rangle = p$$
$$P\{X=0\}=1-p \qquad \langle X^2\rangle = p$$
$$var\{X\} = \langle X^2\rangle - \langle X\rangle^2 = p(1-p)$$

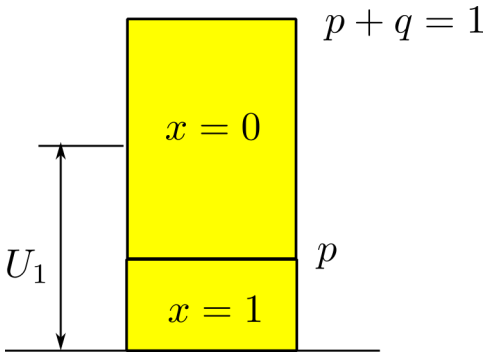
- wartość oczekiwaną zastępujemy estymatorem – średnią arytmetyczną

$$\overline{x^m} = \frac{1}{N}\sum_{i=1}^{N} x_i^m, \quad x_i \sim Ber(p)$$

- w metodzie MC istotna jest jeszcze wartość odchylenia standardowego średniej

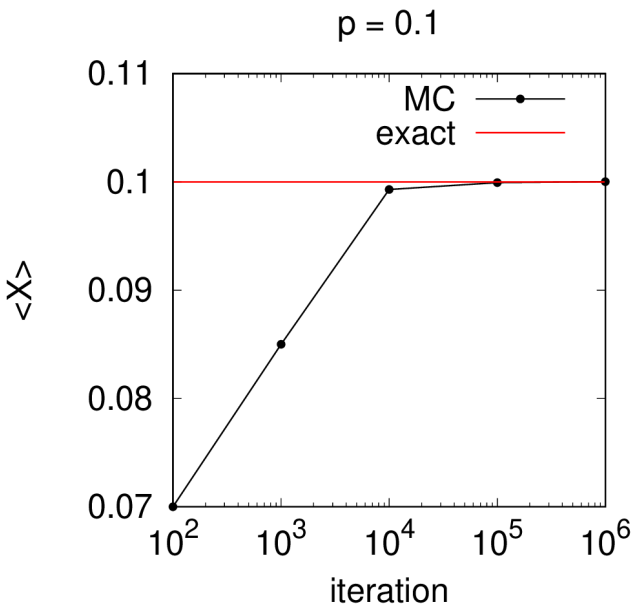
$$\sigma_{\overline{x}} = \sqrt{\frac{var\{X\}}{N}}$$

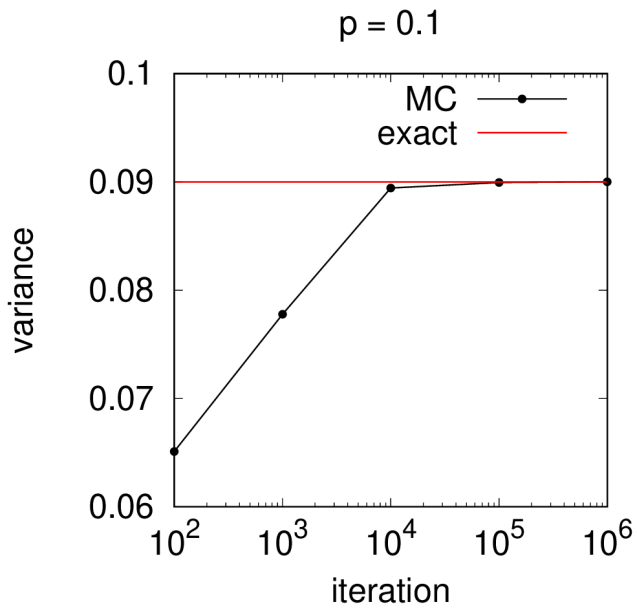
```
inicjalizacja: N, p, S1 = 0, S2 = 0

for(i=1; i <= N; i++){
    U1=(double)rand()/RAND_MAX  //U1 ~ U(0,1)
    if(U1 <= p)  x=1
    else  x=0

    S1+=x
    S2+=x^2
    x̄ = S1/i
    var = S2/i−x̄^2
    σx̄ = √(var/i)
}
```

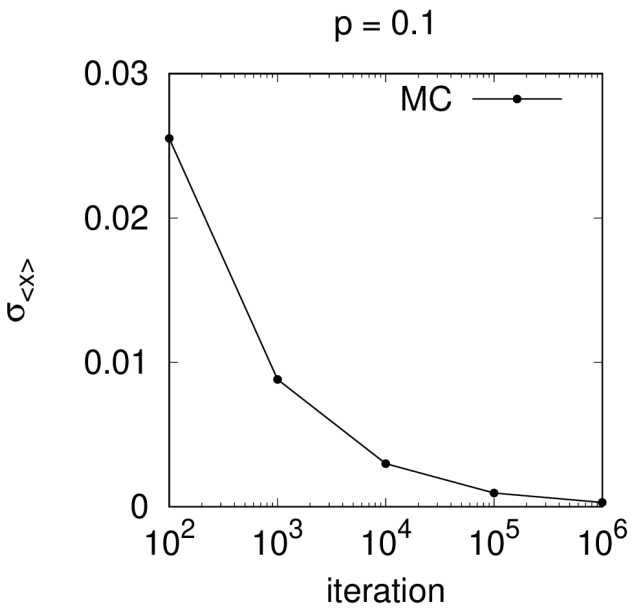


Wyniki dla rozkładu Bernoulliego generowanego metodą MC

$$\langle X \rangle = p \qquad var\{X\} = p(1-p)$$







- algorytm MC działa: średnia i wariancja dążą do wartości dokładnych rozkładu
- wariancja dotyczy pojedynczej zmiennej losowej (drugi rysunek)
- odchylenie standardowe średniej (trzeci rysunek) określa rozrzut wyniku numerycznego wokół średniej, maleje jak 1/N – wynik typowy dla metody MC

Dowolny rozkład - algorytm Metropolisa

Kolejny element ciągu $X_1, X_2, \ldots, X_i, X_{i+1} = ?$

generujemy wykonując poniższe czynności

tu musimy zagwarantować symetrię: lewo-prawo

- losujemy nowe **proponowane** położenie

$$\widetilde{X}_{i+1} = X_i + U_1, \quad U_1 \sim U\left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$$

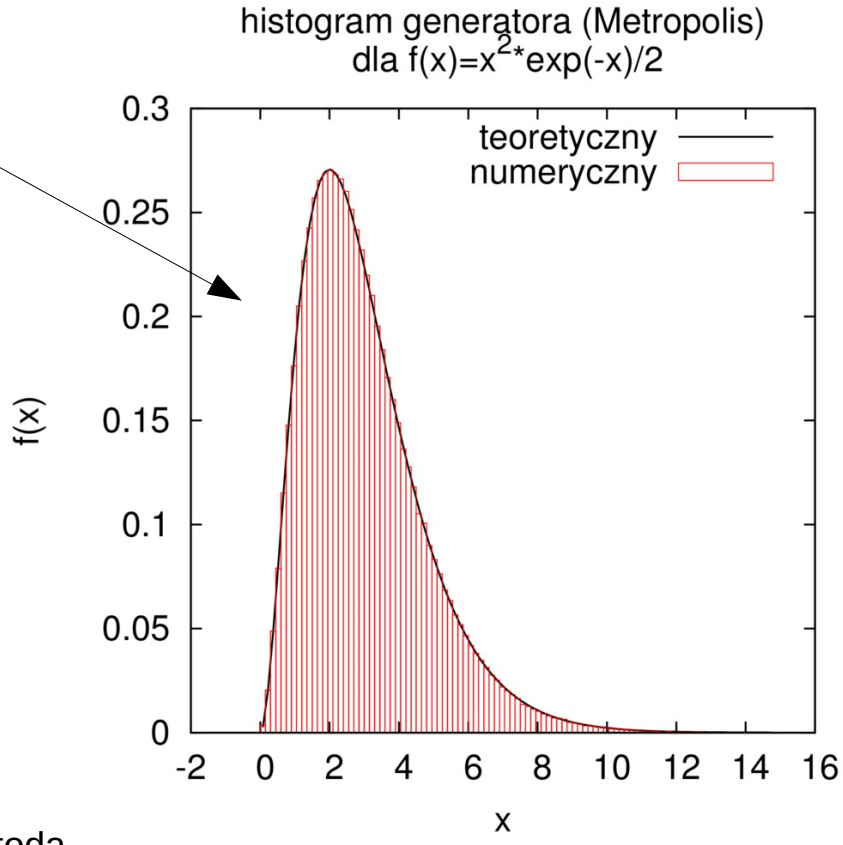
np.: $\Delta = 0.1$

- określamy prawdopodobieństwo akceptacji nowego położenia

$$p_{acc} = \min\left\{\frac{f\left(\widetilde{X}_{i+1}\right)}{f\left(X_i\right)}, 1\right\}$$

- w sposób losowy sprawdzamy czy nowe położenie jest akceptowane

$$U_2 \sim U(0,1) \quad \rightarrow \begin{cases} U_2 \leq p_{acc} & \Rightarrow \quad X_{i+1} = \widetilde{X}_{i+1} \\ U_2 > p_{acc} & \Rightarrow \quad X_{i+1} = X_i \\ \widetilde{X}_{i+1} \notin (a,b) & \Rightarrow \quad X_{i+1} = X_i \end{cases}$$



- metoda jest skuteczna i w pewnych przypadkach jest jedyną metodą pozwalającą próbkować fgp – gdy rozkład jest wielowymiarowy
- wadą może być silna korelacja pomiędzy kolejnymi elementami

Całkowanie MC – metoda podstawowa

- najprostszy sposób zastosowania metody MC związany jest z całkowaniem, co wynika bezpośrednio z relacji pomiędzy wartością oczekiwaną a jej estymatorem – średnią artmetyczną

- wiele problemów rozwiązujemy wykonując całkowanie metodą MC

$$C = \langle z \rangle = \int_{-\infty}^{\infty} z\, g(z) dz \approx \bar{z} = \frac{1}{N} \sum_{i=1}^{N} z_i, \quad z_i \sim Dist\{g(z)\}$$

Zazwyczaj tego problemu nie rozwiążemy, bo nie znamy g(z) – wiemy tylko, że ta fgp istnieje

Dlaczego?

- wartość **z** może być wynikiem działania skomplikowanego algorytmu
- rozkład **g(z)** możemy dopiero próbkować (np. tworząc łańcuch Markowa)

Argumentami funkcji podcałkowej z są zmienne o znanym rozkładzie

$$z = z(\vec{x}), \qquad \vec{x} = [x_1, x_2, \ldots, x_n]$$

$$f(\vec{x}) = f(x_1, x_2, \ldots, x_n)$$

wtedy dokonujemy transformacji

$$g(z)dz = f(\vec{x})d^n x$$

$$\langle z \rangle = \int_{-\infty}^{\infty} z g(z) dz = \int_{a_1}^{b_1} \ldots \int_{a_n}^{b_n} z(\vec{x})\, f(\vec{x})\, dx_1 \ldots dx_n$$

- postać wyrażenia identyczna, ale używamy generatora o znanym rozkładzie f(x)

$$\overline{z^m} = \frac{1}{N}\sum_{i=1}^{N} z^m(\vec{x}_i), \qquad \vec{x}_i \sim Dist\{f(\vec{x})\}$$

$$var = \overline{z^2} - \overline{z}^2$$

$$\sigma_{\overline{z}} = \sqrt{\frac{var}{N}}$$

- elementy wektora losujemy tylko raz i używamy go dla wszystkich momentów

inicjalizacja: N, $S_1 = 0$, $S_2 = 0$

for (i=1; i <= N; i++){

$\vec{x} \sim Dist\{f(\vec{x})\}$

$S_1 += z(\vec{x})$

$S_2 += z^2(\vec{x})$

$\overline{z} = S_1/i$

$\overline{z^2} = S_2/i$

$var = \overline{z^2} - \overline{z}^2$

$\sigma_{\overline{z}} = \sqrt{var/i}$

}

Interpretacja statystyczna wyniku modelowania/symulacji Monte Carlo

- z **centralnego twierdzenia granicznego** wynika, że obliczona średnia (estymator wartości oczekiwanej) ma rozkład normalny (Gaussa)

$$\overline{z} \sim N(\mu, \sigma)$$

- natomiast **prawo wielkich liczb** mówi, że błąd oszacowania średniej maleje z liczbą losowań

$$\sigma_{\overline{z}} = \frac{\sigma_z}{\sqrt{N}}$$

σ_z - odchylenie standardowe pojedynczej zmiennej losowej z

- oba twierdzenia stanowią fundament działania metody Monte Carlo
- wyniki MC mają jednoznaczną interpretację, niezależnie od sposobu wykonania symulacji czy użytego rozkładu prawdopodobieństwa (generatora liczb pseudolosowych)
- MC jest metodą stochastyczną – wynik obarczony jest niepewnością (błędem?), a ten możemy na bieżąco kontrolować tj. prowadzić obliczenia tak długo aż jego wartość obniży się poniżej założonego minimalnego progu

Przykład – całkowanie wielowymiarowe metodą MC

$$C = \int_0^1 \dots \int_0^1 \sqrt{x_1 + x_2 + x_3 + x_4 + x_5}\, dx_1 \dots dx_5$$

$C = 1.56705355$ - wynik dokładny

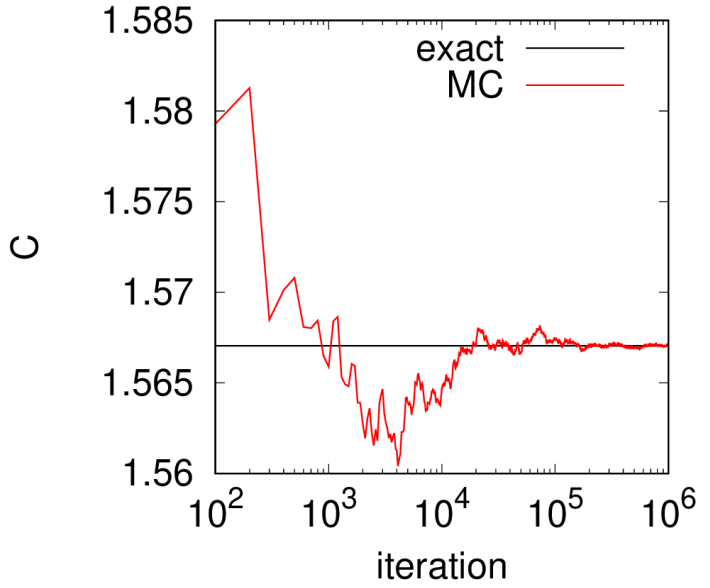
$$z(\vec{x}) = \sqrt{x_1 + x_2 + x_3 + x_4 + x_5}$$

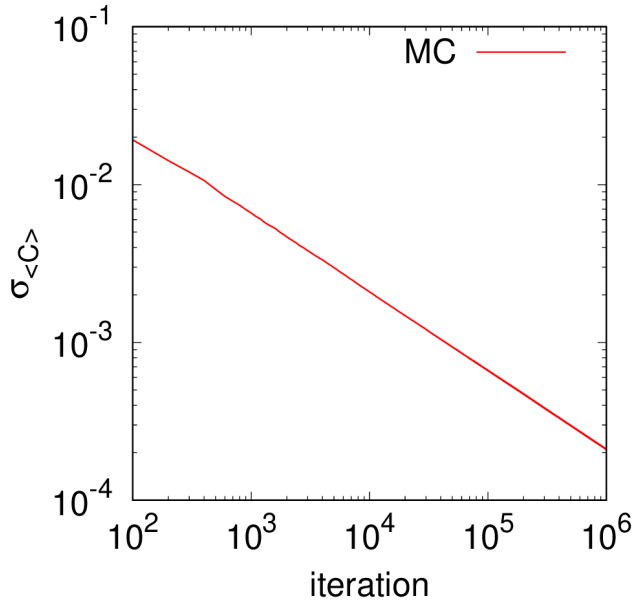
$$f(\vec{x}) = 1 = U^5(0,1)$$

$$\overline{z} = \frac{1}{N}\sum_{i=1}^{N} z(\vec{x}_i), \qquad x_1, \dots, x_5 \sim U(0,1)$$

$$\overline{z^2} = \frac{1}{N}\sum_{i=1}^{N} z^2(\vec{x}_i)$$

$$\sigma_{\overline{z}} = \sqrt{\frac{\overline{z^2} - \overline{z}^2}{N}}$$

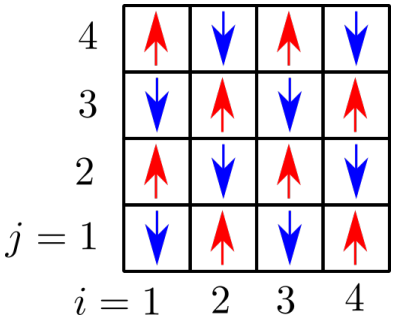




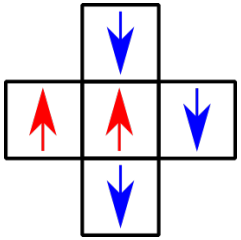
Modelowanie Monte Carlo – model Isinga

- **model Isinga** to prosty model układu spinów (1d, 2d, 3d), w którym uwzględniamy tylko oddziaływanie spin-spin pomiędzy najbliższymi sąsiadami

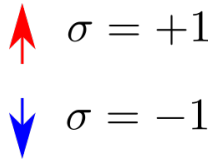
układ spinów 4x4



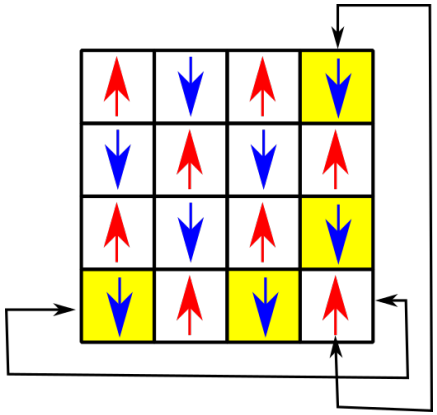
oddziaływanie:
spin oddziałuje tylko 4 sąsiadami



reprezentacja liczbowa spinów:



- wprowadzamy **periodyczne warunki brzegowe**, aby małe (skończone) układy imitowały układ nieskończony (otwarty)



- energię układu liczymy sumując wkłady od par spin-spin (tylko najbliżsi sąsiedzi)

$$E(\vec{\sigma}) = -J \sum_{\mu,\nu} \sigma_\mu \sigma_\nu \qquad \vec{\sigma} = [\sigma_1, \ldots, \sigma_N]$$

$$\mu = (i,j) \quad \nu = \{(i-1,j),\ (i+1,j),\ (i,j-1),\ (i,j+1)\}$$

J > 0 – całka wymiany

- zgodnie z mechaniką statystyczną prawdopodobieństwo realizacji danej konfiguracji zależy od energii oraz temperatury, dlaczego?

temperatura → drgania atomów → zmiana pola E/B w układzie własnym spinu → możliwy obrót spinu

$$p(\vec{\sigma};\ \beta) = \frac{e^{-\beta E(\vec{\sigma})}}{Z(\beta)}, \qquad \beta = \frac{1}{k_B T}$$

k_B -stała Boltzmanna
T - temperatura

Z(β) to czynnik normalizacyjny – **funkcja rozdziału**

$$Z(\beta) = \sum_{\sigma_1=\pm1,\ldots,\sigma_N=\pm1} e^{-\beta E(\vec{\sigma})} = \sum_E D(E) e^{-\beta E}$$

sumowanie po wszystkich możliwych konfiguracjach

sumowanie po wszystkich możliwych energiach układu

- dla T>0 spodziewamy się że spiny będą się obracać, w danej chwili realizowana jest jedna konfiguracja, energia układu musi być średnią po tych konfiguracjach

$$\langle E \rangle = \sum_{\vec{\sigma}} E(\vec{\sigma}) p(\vec{\sigma}; \beta)$$

$$\langle E^2 \rangle = \sum_{\vec{\sigma}} E^2(\vec{\sigma}) p(\vec{\sigma}; \beta)$$

- obie wielkości możemy wykorzystać np. do policzenia **ciepła właściwego na pojedynczy spin**

$$c_v = \frac{\beta^2}{N} \underbrace{\left(\langle E^2 \rangle - \langle E \rangle^2 \right)}_{\text{wariancja}}$$

$$Z(\beta) = \sum_E D(E) e^{-\beta E}$$

D(E) – gęstość stanów
(ilość stanów o danej energii)

$E[J]$	$D_{2\times2}$	$D_{4\times4}$
0	12	20524
4	0	13568
8	2	6688
12	–	1728
16	–	424
20	–	64
24	–	32
28	–	0
32	–	2

Obrót pojedynczego spinu powoduje zmianę energii o wartość

$$\Delta E = 0, \pm 4J, \pm 8J$$

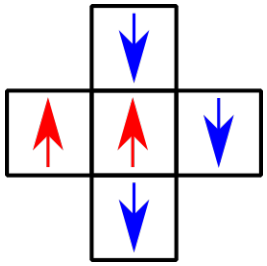
- okazuje się że **naturalne** fluktuacje kierunków spinów będące skutkiem skończonej temperatury są przyczyną specyficznych własności układu fizycznego

- **jak wykonać symulację MC? → iteracyjnie**

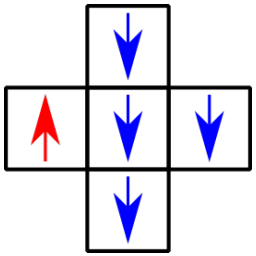
  w pojedynczym kroku dokonujemy próby obrócenia jednego (np. losowego) spinu, nową konfigurację akceptujemy bądź nie zgodnie z **algorytmem Metropolisa,**

  jedna iteracja → próba obrócenia wszystkich spinów w układzie

stara konfiguracja



proponowana nowa konfiguracja



$$E_{old} = -J(1-1-1-1) = 2J \qquad E_{new} = -J(-1+1+1+1) = -2J$$

algorytm Metropolisa

$$\Delta E = E_{new} - E_{old} = -4J$$

$$p_{acc} = \min\left\{\frac{p(E_{new};\beta)}{p(E_{old};\beta)}, 1\right\} = \min\left\{e^{-\beta\Delta E}, 1\right\}$$

$$U_2 \sim U(0,1) \quad \rightarrow \quad \begin{cases} U_2 \leq p_{acc} & \sigma_\mu^{new} = -\sigma_\mu^{old} \\ U_2 > p_{acc} & \sigma_\mu^{new} = \sigma_\mu^{old} \end{cases}$$

mimo iż układ jest mały to liczba dostępnych stanów jest ogromna

- **ile iteracji trzeba wykonać?**

liczba konfiguracji: $K = 2^N \quad \rightarrow \quad n = 6 \quad \rightarrow \quad N = n^2 \quad \rightarrow \quad K = 2^{36} \approx 68.7 \cdot 10^9$

w MC **próbkujemy losowo przestrzeń** dostępnych stanów więc $L_{iter} << K \qquad L_{iter} \sim 10^5 - 10^6$

Pseudokod dla modelu Isinga 2D, wyznaczamy: energię + ciepło właściwe

```
inicjalizacja: T, β,n, N = n^2, S_1 = 0, S_2 = 0, L_iter

    spins[][]=1.0        //spiny = ↑
    E_tot = -J N 4 1/2   //polaryzacja spinowa
    iter=0
for(l=1; l <= L_iter; l++){
    for(i=1; i <= n; i++){
        for(j=1; j <= n; j++){
            σ_μ=spins[i][j]
            σ_ν1, σ_ν2, σ_ν3, σ_ν4 ← spins[i±1][j±1]
            E^old = -J σ_μ (σ_ν1 + σ_ν2 + σ_ν3 + σ_ν4)
            E^new = -E^old
            p_acc = min{ e^{-β(E^new - E^old)}, 1 }
            U_1 ~ U(0,1)
            if(U_1 ⩽ p_acc) spins[i][j]=-σ_μ
            E_tot = E_tot - E^old + E^new
            iter++
            S_1+ = E_tot
            S_2+ = E_tot^2
            Ē = S_1/iter
            Ē^2 = S_2/iter
            c_v = β^2 (Ē^2 - Ē^2) /N
        }
    }
}
```

- wybór stanu początkowego jest nieistotny, ale dla spolaryzowanego układu łatwo określić energię
- pamiętamy o periodycznych WB

- wyniki modelowania układu spinów 2D (**n=2, 4, 16, 24**)

temperatura krytyczna: T_c=2.269 (wynik analityczny)

$T > T_c$ - układ paramagnetyczny

$T < T_c$ - układ ferromagnetyczny

ciepło właściwe na spin

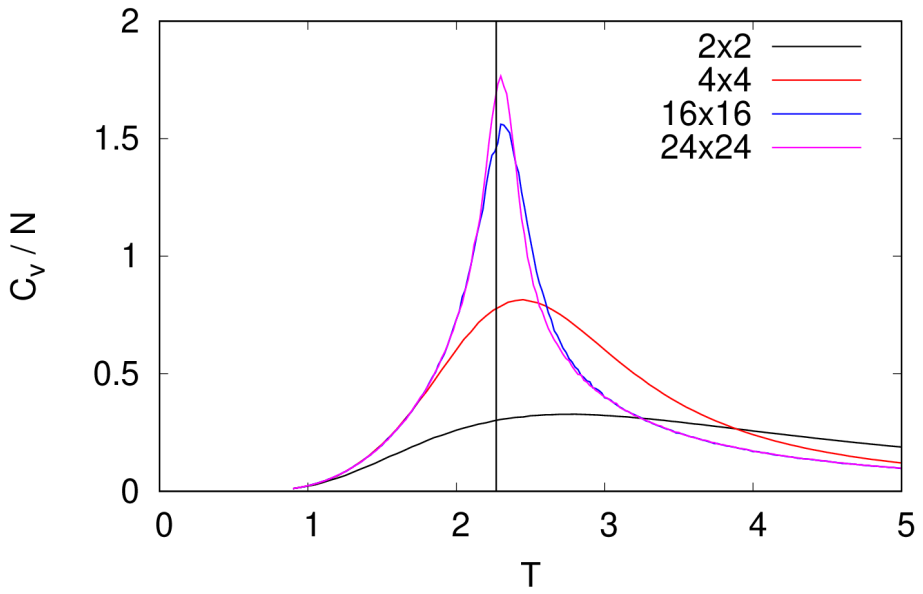
$$c_v = \frac{\beta^2}{N}\left(\langle E^2\rangle - \langle E\rangle^2\right)$$

magnetyzacja układu

$$M = \left\langle \sum_{i=1}^{N} \sigma_i \right\rangle$$

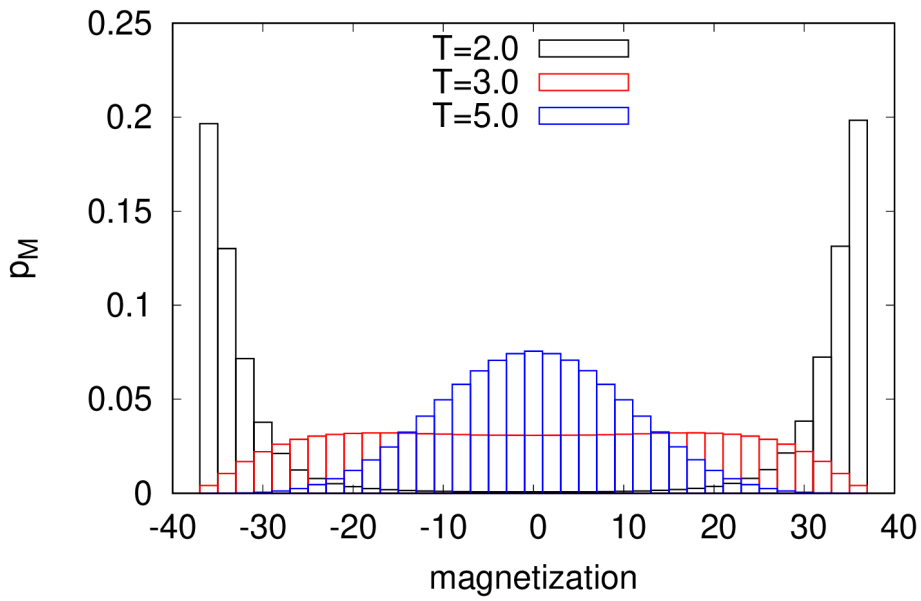
$$M = -N, -N+2, \ldots, 0, \ldots, N-2, N$$

$L_{it}=10^5$ N, T_c=2.269



- widoczny jest efekt rozmiarowy

N=6x6, $L_{it}=10^7$ N



- dla $T<T_C$ układ staje się częściowo spolaryzowany

Modelowanie MC – optymalizacja wartości funkcji (**metoda symulowanego wyżarzania**)

- jednym z zastosowań MC jest poszukiwanie **minimum globalnego** wartości funkcji tzw. **funkcji celu f(x)** (np. energii stanu podstawowego – czyli najstabilniejszej konfiguracji)

- na czym polega problem od strony numerycznej?

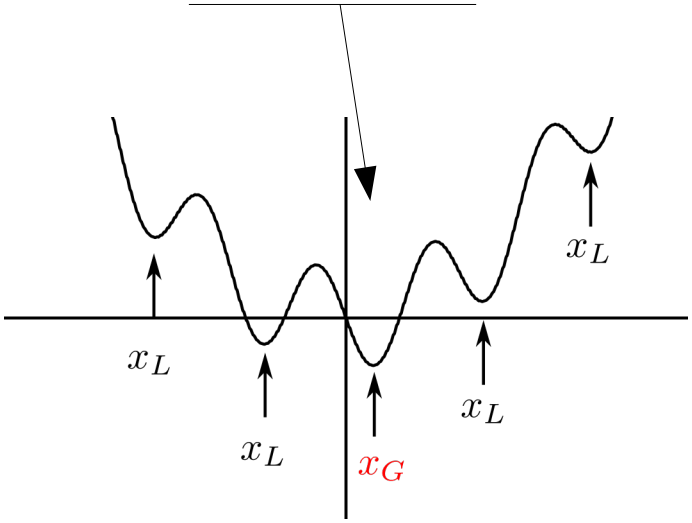
przykład: szukamy energii stanu podstawowego N oddziałujących cząstek

$$E_N = \sum_{i=1}^{N} \sum_{j>i}^{N} V(\vec{r}_i, \vec{r}_j)$$

wymiarowość problemu (3D)
(liczba stopni swobody)

$$dim = 3N$$

$$N = 100 \quad \rightarrow \quad dim = 300$$



minimum globalne

W jaki sposób przeszukać przestrzeń o takiej liczbie wymiarów?

- w tego typu problemach, klasyczne metody numeryczne stają się bezużyteczne (klasyczne iteracyjne: Simplex lub gradientowe)

- przestrzeń dostępnych rozwiązań należy przeszukiwać w sposób losowy (podobnie jak to robiliśmy w problemie Isinga), jedyną użyteczną metodą pozostają metody stochastyczne (Monte Carlo)

  - **metoda symulowanego wyżarzania (SA)**
  - algorytmy ewolucyjne: algorytmy genetyczne (GA), algorytm kolonii mrówek (AC) etc.

uwaga: generalnie metody oparte na **GA** są wydajniejsze/skuteczniejsze niż **SA**
ale są bardziej skomplikowane i nie będziemy ich omawiać

Algorytm symulowanego wyżarzania

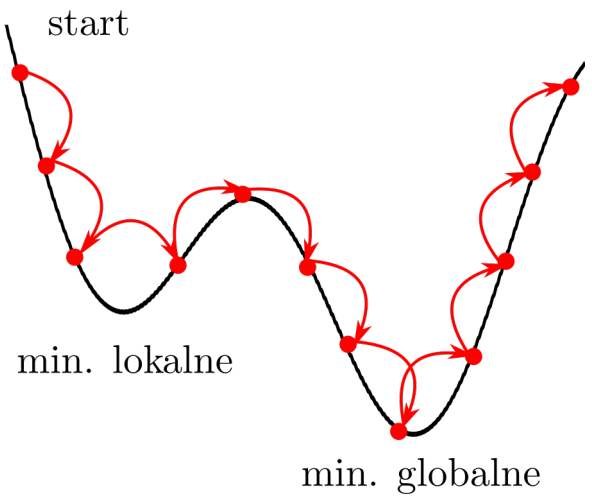
1) ustalamy temperaturę T_m i generujemy losowy ciąg L położeń pojedynczego wędrowca w przestrzeni zgodnie z **algorytmem Metropolisa** (znowu)

$$T_m: \quad x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_L$$

$$x_{i+1}^{new} = x_i + U(-\Delta, \Delta) \qquad (\Delta\text{-empirycznie})$$

$$\mathrm{p}_{acc} = \min\left\{1, \exp\left(-\frac{f(x_{new}) - f(x_{old})}{kT_m}\right)\right\}$$

$$T >> 1 \rightarrow p_{acc}(E_{new} > E_{old}) > 0$$



- jeśli T jest duże, to prawdopodobieństwo akceptacji nowego „**gorszego**” wyniku też jest duże
- wędrowiec może w sposób losowy odwiedzić dowolne miejsce w przestrzeni znajdując minimum

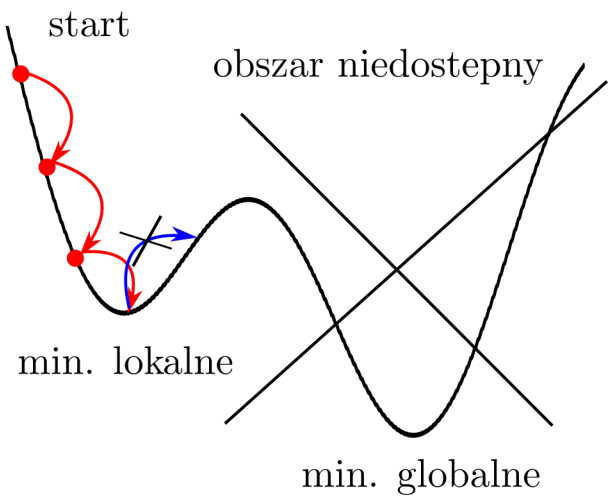
2) obniżamy temperaturę i powtarzamy procedurę

$$T_{m+1} < T_m \qquad \left(T_{m+1} = \frac{T_m}{2}\right)$$

$$T_{m+1}: \quad x_L \rightarrow x_{L+1} \rightarrow x_{L+2} \rightarrow \ldots \rightarrow x_{2L}$$

$$\mathrm{p}_{acc} = \min\left\{1, \exp\left(-\frac{f(x_{new}) - f(x_{old})}{kT_{m+1}}\right)\right\}$$

$$T << 1 \rightarrow p_{acc}(E_{new} > E_{old}) \sim 0$$



- małe T oznacza bardzo małe prawdopodobieństwo zaakceptowania wyniku o większej wartości funkcji celu f(x)
- ruch wędrowca ograniczony jest do lokalnej doliny f(x)
- dostęp do minimum globalnego może zostać zablokowany

Symulowane wyżarzanie: czego się spodziewać? oraz zalecany sposób postępowania

- po obniżeniu temperatury do T~ 0 oczekujemy, że wędrowiec zlokalizuje się w minimum: **lokalnym** (najczęściej) lub **globalnym** (baaardzo rzadko)
- aby zwiększyć szanse znalezienia minimum globalnego, zamiast pojedynczego wędrowca używamy N wędrowców (np. N=100, 1000, 10000)
- metodę łatwo zrównoleglić, wędrowcy działają niezależnie od siebie
- liczbę kroków w danej temperaturze, zakres temperatur oraz sposób zmiany temperatury należy dobierać empirycznie
- im bardziej złożony jest problem (np. ze względu na dużą liczbę wymiarów) tym mniejsza szansa na dotarcie do minimum po jednym wykonaniu programu, zazwyczaj trzeba algorytm wykonać wielokrotnie startując z różnych punktów w przestrzeni
- nie ma 100% gwarancji na to, że minimum globalne uda nam się znaleźć - często jednak znalezione minima nie odbiegają znacznie wartością od minimum globalnego

Pseudokod symulowanego wyżarzania dla N wędrowców

inicjalizacja: T_{min}, T_{max}

M - liczba punktow posrednich temperatury

n – liczba wymiarow

N – liczba wedrowcow

$\vec{\Delta}_{max} = [\Delta_1, \Delta_2, \ldots, \Delta_n]$

$\vec{r}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ – wektor startowy i–tego wedrowca

$\Delta T = \frac{T_{max} - T_{min}}{M}$

for(it=1; it <= IT_{max}; it++){

$T = T(it)$ — sposób zmiany temperatury →

for(i=1; i<= N; i++){

$\vec{\Delta} \sim dist\{U^n(\cdot,\cdot)\}$ $\quad |\vec{\Delta}| \leq |\vec{\Delta}_{max}|$

$\vec{r}_i^{new} = \vec{r}_i + \vec{\Delta}$

$p_{acc} = \min\left\{1, \exp\left(-\frac{f(\vec{r}_i^{new}) - f(\vec{r}_i)}{kT}\right)\right\}$

$U_1 \sim U(0,1)$

if $(U_1 \leq p_{acc})$ {

$\vec{r}_i \leftarrow \vec{r}_i^{new}$

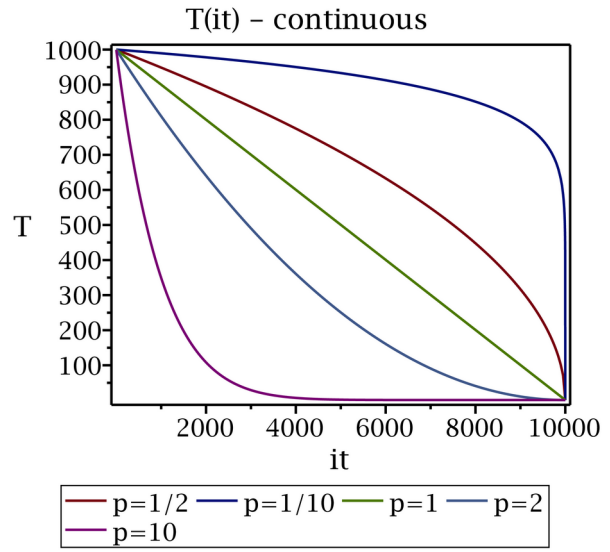
}else{

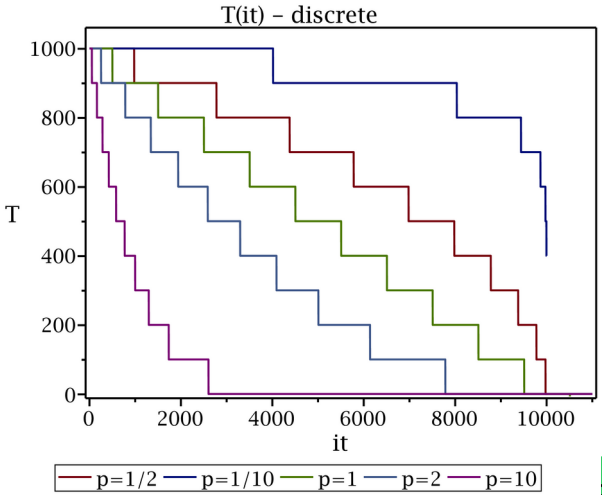
$\vec{r}_i$ – bez zmian

}

}

}



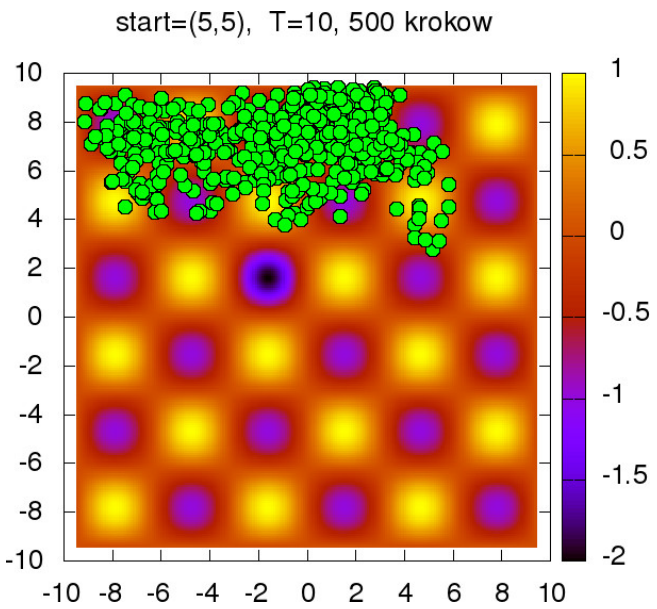


Przykład - szukamy minimum globalnego funkcji celu f(x,y)

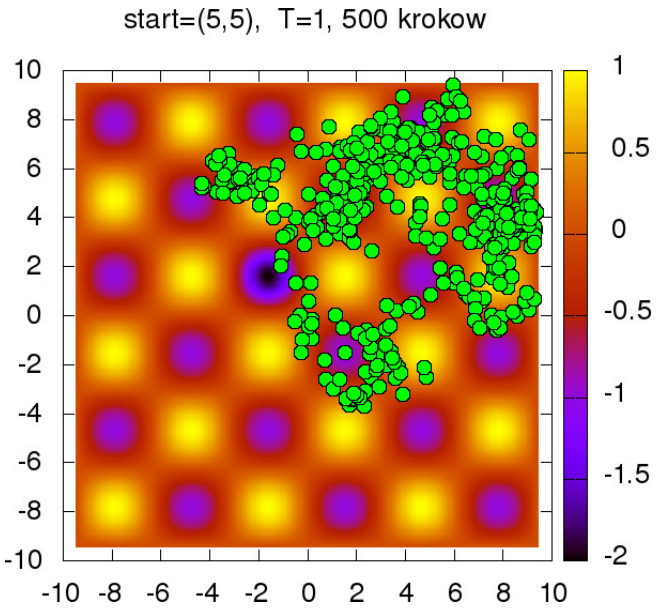
$$f(x,y) = \underbrace{\sin(x)\sin(y)}_{\text{min. lokalne}} - \underbrace{\exp\left[-\left(x+\frac{\pi}{2}\right)^2 - \left(y-\frac{\pi}{2}\right)^2\right]}_{\text{min. globalne}}$$

$$x, y \in [-3\pi,\, 3\pi]$$

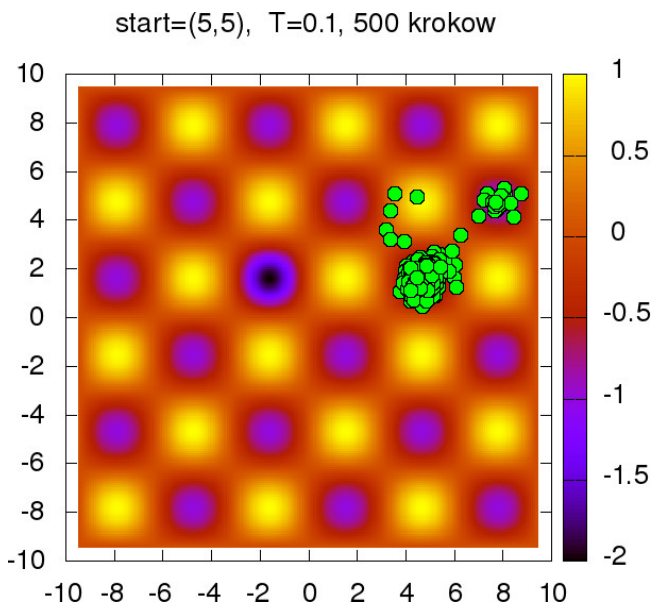
Wynik dla pojedynczego wywołania algorytmu (jeden wędrowiec) – kropki to kolejne położenia.
Dla każdego przypadku - start z tego samego punku.



- temperatura za duża,
  zbyt duży krok, wędrowiec
  odwiedza maksima



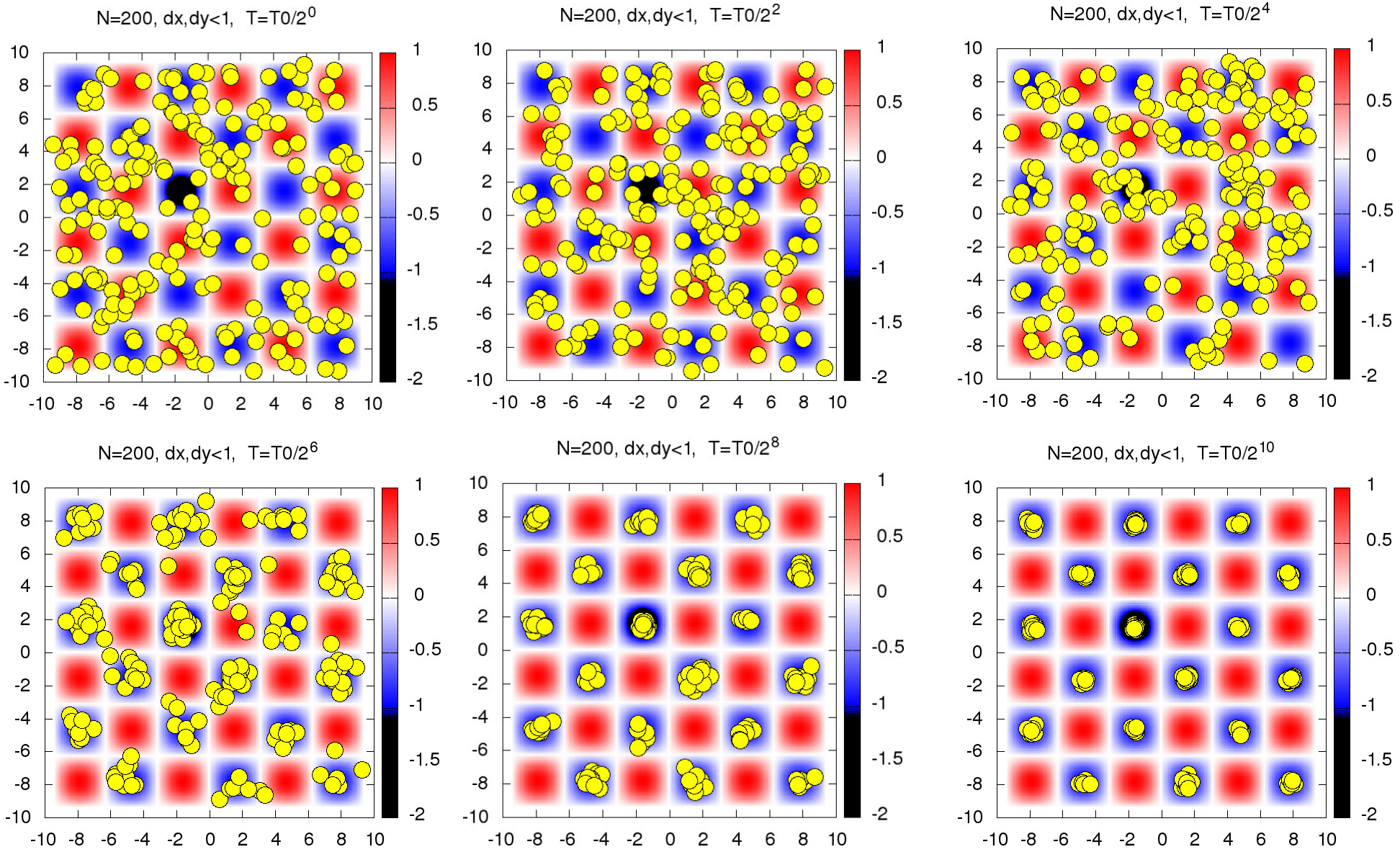
- temperatura średnia,
  wędrowiec omija maksima
  i przemieszcza się dolinami



- temperatura niska,
  wędrowiec bardzo szybko
  i lokalizuje się w minimum

- wyniki dla M=200 wędrowców i temperatury obniżanej co 100 iteracji

$$T_{new} = \frac{T_{old}}{2^2}$$



jedno ze znalezionych minimów to minimum globalne

Podsumowanie

- generalnie metoda MC to algorytm numeryczny dedykowany problemom o „charakterze probabilistycznym” (losowym) – ale nie tylko....
- MC możemy użyć do rozwiązywania problemów z natury losowych (fluktuacje spinów w modelu Isinga) oraz problemów, którym taką interpretację możemy nadać (całkowanie numeryczne)
- „silnikiem” MC jest generator liczb pseudolosowych o zadanym rozkładzie
- rozwiązanie problemu w MC obarczone jest niepewnością statystyczną – wielkość tego „błędu” możemy na bieżąco kontrolować (automatyczna kontrola błędu)
- MC nie rozwiążemy wszystkich problemów i zazwyczaj nie jest to metoda numeryczna pierwszego wyboru, ale w przypadku problemów o dużej złożoności obliczeniowej może być jedyną użyteczną (dostępną) metodą
- kod źródłowy programu MC zazwyczaj jest krótszy w porównaniu do innych metod, łatwiej też jest zaimplementować skomplikowane warunki brzegowe - to też może być istotna korzyść przemawiająca za użyciem MC
- nie ma jednej sformalizowanej wersji metody MC, dlatego jest bardzo elastyczna, w zasadzie każdy problem rozwiązywalny tą metodą możemy rozwiązywać na wiele sposobów (dotyczy to zwłaszcza minimalizacji niepewności), od naszej inwencji zależy gdzie w jakim zakresie ją wykorzystamy

Literatura

- J. E. Gentle - „Random number generation and Monte Carlo methods”
- D.P. Kroese, T. Taimre, Z.I. Botev - „Handbook of Monte Carlo methods”
- F. J. Vesely - „Computational Physics. An Introduction”
- M.H. Kalos, P.A. Withlock - „Monte Carlo Methods”
- W.L. Dunn, J.K. Shultis - „Exploring Monte Carlo Methods”
- T. Pang - „Introduction to computational physics”
- A.K. Hartmann, H. Rieger - „Optimization algorithms in physics”
- R.L. Haupt, D.H. Werner - „Genetic algorithms in electromagnetics”