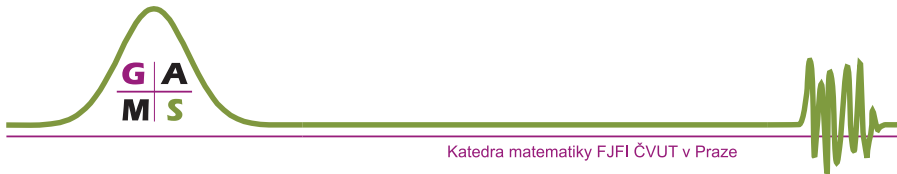


Metropolisův algoritmus

Sociální systémy a jejich simulace
Jana Vacková

13.10.2021



Katedra matematiky FJFI ČVUT v Praze

- Nakódění Metropolisova algoritmu
- Pozorování průběhu potenciální energie

Algorithm 0.1: METROPOLIS()

$x \in \mathcal{X}$

$U = U(x)$

while nejsme spokojeni

do $\left\{ \begin{array}{l} y \sim \Psi(x, \cdot) \\ \text{if } U(y) \leq U(x) \\ \quad \text{then } \begin{cases} x \leftarrow y \\ U \leftarrow U(y) \end{cases} \\ \text{else } \begin{cases} \text{if } \text{rand} < e^{-\beta(U(y)-U(x))} \\ \quad \text{then } \begin{cases} x \leftarrow y \\ U \leftarrow U(y) \end{cases} \end{cases} \end{array} \right.$

Úkol č. 1 - mutace konfigurace - zadání

- Vytvořte funkci na mutování konfigurace `cnew = mutate(c)`.

Úkol č. 1 - mutace konfigurace - zadání

- Vytvořte funkci na mutování konfigurace `cnew = mutate(c)`.

Vzpomeňme si, že:

- Hýbáme všemi částicemi najednou:

$$y_j = x_j + \delta_j, \quad \forall j \in \hat{N}$$

kde $\delta_j \sim \mathcal{U}(-R, R)$, $R = \frac{L}{N} \cdot 0.1$ je šum a L je délka kruhu

- Předpokládáme pro jednoduchost, že $L = N$
- Předjíždění je zakázáno, tj. musí platit

$$y_j < y_{j-1}, \quad \forall j \in \hat{N}$$

Úkol č. 1 - mutace konfigurace - řešení

- Vytvořte funkci na mutování konfigurace `cnew = mutate(c)`.

```
function cnew = mutate(c)
% zasumi soucasnou konfiguraci

h = 0.1; % sum
N = length(c); % pocet castic
L = N; % delka kruznice
R = L/N * h; % pro uniformni rozdeleni
delta = -R + 2*R*rand(size(c));
cc = c + delta; % nova konfigurace

while ~issorted([cc, cc(1)+L]) % pokud doslo k predjizdeni
    delta = -R + 2*R*rand(size(c));
    cc = c + delta;
end

cnew = cc;
end
```

Úkol č. 2 - potenciální energie - zadání

- Vytvořte funkci na výpočet potenciální energie konfigurace $u = U(c)$.

Úkol č. 2 - potenciální energie - zadání

- Vytvořte funkci na výpočet potenciální energie konfigurace $u = U(c)$.

Vzpomeňme si, že:

$$U(x) = U(x_1, \dots, x_N) = \sum_{j=1}^N U_j(x_1, \dots, x_N),$$

kde $U_j(x_1, \dots, x_N) = g(x_j - x_{j-1})$, kde g je např. $g(d) = \frac{1}{d}$

Úkol č. 2 - potenciální energie - řešení

- Vytvořte funkci na výpočet potenciální energie konfigurace $u = U(c)$.

```
function u = U(c)
% potencialni energie konfigurace c

L = length(c);
dx = diff([c, c(1)+L]);
u = sum(1./dx);
end
```

Úkol č. 3 - Metropolisův algoritmus - zadání

- Nakóďte Metropolisův algoritmus.

Úkol č. 3 - Metropolisův algoritmus - zadání

- Nakóděte Metropolisův algoritmus.

Algorithm 0.3: METROPOLIS()

$x \in \mathcal{X}$

$U = U(x)$

while nejsme spokojeni

do $\left\{ \begin{array}{l} y \sim \Psi(x, \cdot) \\ \text{if } U(y) \leq U(x) \\ \quad \text{then } \begin{cases} x \leftarrow y \\ U \leftarrow U(y) \end{cases} \\ \text{else } \begin{cases} \text{if } \text{rand} < e^{-\beta(U(y)-U(x))} \\ \quad \text{then } \begin{cases} x \leftarrow y \\ U \leftarrow U(y) \end{cases} \end{cases} \end{array} \right.$

Úkol č. 3 - Metropolisův algoritmus - řešení

- Nakóďte Metropolisův algoritmus.

```
N = 5;
c0 = sort(rand(1, N)*N);           % nahodna pocatecni konfiguraci
L = length(c0);                   % L = N
c = c0; u = U(c0);                 % poc. konfigurace a její energie
n = 1; nmax = 100000;              % maximalni pocet kroku
beta = 10;
```

Úkol č. 3 - Metropolisův algoritmus - řešení

```
while n <= nmax
    cnew = mutate(c);
    unew = U(cnew);
    if rand < exp(-beta*(unew - u))
        c = cnew; u = unew;
        C{n} = c;
        n = n + 1;
        plot(mod(c, L), 0, 'o', 'MarkerSize', 15)
        axis([0 L -1 1]);
        pause(0.1);
    end
end
```

Úkol č. 4 - průběh potenciální energie

- Vykreslete průběh potenciální energie systému, jak se měnila v průběhu simulace.
- Vyzkoušejte si to s různým počátečním rozestavením částic (náhodné, deterministické).

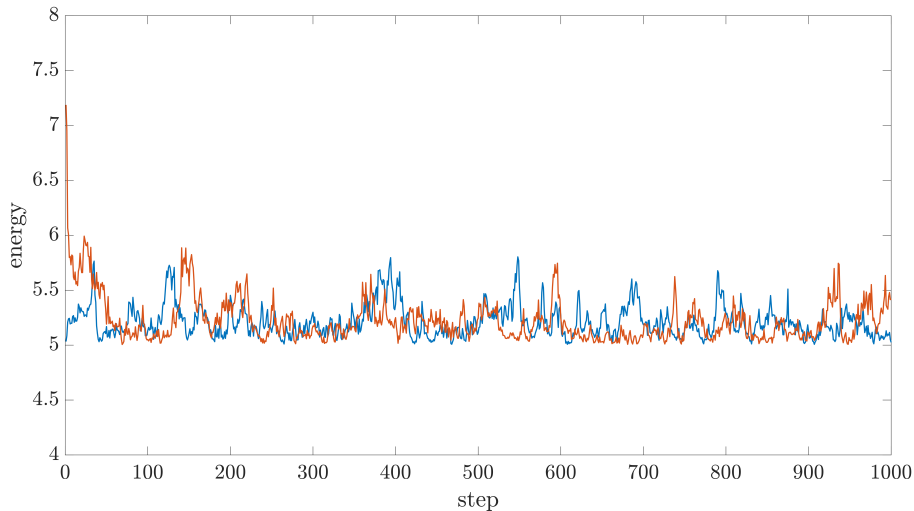
Úkol č. 4 - průběh potenciální energie

- Vykreslete průběh potenciální energie systému, jak se měnila v průběhu simulace.
- Vyzkoušejte si to s různým počátečním rozestavením částic (náhodné, deterministické).

```
c0 = 1:N;           % deterministicke pocatecni rozestaveni castic

while n <= nmax
    cnew = mutate(c);
    unew = U(cnew);
    if rand < exp(-beta*(unew - u))
        c = cnew; u = unew;
        C{n} = c; E(n) = u;
        n = n + 1;
        plot(mod(c, L), 0, 'o', 'MarkerSize', 15)
        axis([0 L -1 1]);
        pause(0.1);
    end
end
```

Úkol č. 2 - průběh potenciální energie - výstup



Pro dnešek máme hotovo. :-)