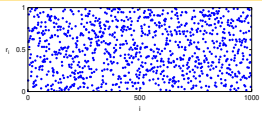
Náhodná čísla v algoritmech

- **Deterministický** algoritmus = posloupnost operací dávající správnou odpověď nebo oznamující neúspěch.
  Příklad: inverze matice Gaussovou–Jordanovou eliminací s výběrem hlavního prvku (pivoting).
- **Monte Carlo** algoritmus = procedura používající (pseudo)náhodná čísla k získání výsledku, který je správný s určitou pravděpodobností; typicky numerický výsledek zatížený náhodnou (stochastickou) chybou.
  Příklad: Výpočet vnitřní energie, $\langle E_{\mathrm{kin}} + E_{\mathrm{pot}} \rangle$, v MD simulaci v *NVT* souboru
- **Las Vegas** algoritmus používá náhodná čísla k získání deterministického výsledku.
  Příklad: inverze matice Gaussovou–Jordanovou eliminací s tím, že k výběru dostatečně velkého hlavního prvku se používají náhodná čísla, např. výběrem z tabulky dost velkých kandidátů.

Příklad generátoru pseudonáhodných čísel +

$$n_i = 7^5 n_{i-1} \bmod (2^{31}-1), \quad r_i = n_i/2^{31}$$



Monte Carlo integrace (naivní Monte Carlo)

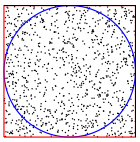
Příklad: Výpočet čísla π

```
INTEGER n celkový počet bodů
INTEGER i
INTEGER nu počet bodů v kruhu
REAL x,y souřadnice bodu ve čtverci
REAL rnd(-1,1) funkce vracející náhodné číslo v intervalu (−1, 1)

nu := 0
FOR i := 1 TO n DO
   x := rnd(-1,1)
   y := rnd(-1,1)
   IF x*x+y*y < 1 THEN nu := nu + 1

PRINT "pi=", 4*nu/n plocha čtverce = 4

PRINT "chyba=", 4*sqrt((1-nu/n)*(nu/n)/(n-1))
```



Též "random shooting". Obecně:

$$\int_\Omega f(x_1,\ldots,x_D)\,\mathrm{d}x_1\ldots\mathrm{d}x_D \approx \frac{|\Omega|}{K}\sum_{k=1}^{K} f(x_1^{(k)},\ldots,x_D^{(k)})$$

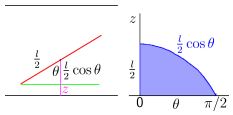
kde $(x_1^{(k)},\ldots,x_D^{(k)})$ je náhodný vektor z oblasti Ω
($|\Omega|$ = plocha, objem...; výpočet π: $\Omega = (-1,1)^2$, $|\Omega| = 4$)

Cvičení I – Buffonova jehla

Mějme linkovaný papír s linkami vzdálenými d. Pravděpodobnost, že náhodně hozená jehla délky l, $l \le d$, protne linku, je $p = 2l/\pi d$

[Georges-Louis Leclerc, Comte de Buffon, 1707–1788]

Důkaz:



definice: symbol $(a < b)$ dává 1, pokud nerovnost platí, jinak 0 (Iversonova závorka)

$$p = \frac{1}{d/2}\int_0^{d/2}\frac{\mathrm{d}z}{\pi/2}\int_0^{\pi/2}\mathrm{d}\theta\left(z < \frac{l}{2}\cos\theta\right) = \frac{1}{d/2}\frac{1}{\pi/2}\int_0^{\pi/2}\frac{l}{2}\cos\theta\mathrm{d}\theta = \frac{2l}{\pi d}$$

Použití (δp je standardní chyba p)

rel. chyba

$$\pi \approx \frac{2l}{pd}, \quad \text{kde } p = \frac{n_{\text{protne}}}{n_{\text{celkem}}}, \quad \delta p \approx \sqrt{\frac{p(1-p)}{n-1}}, \quad \delta\pi = \frac{2l}{pd}\frac{\delta p}{p}$$

Poděkování: animace Bc. Martin Mašek (2024)

for me: grid: pic/buffon-grid.pdf and buffon.sh

Cvičení II +

Lehčí. Vypočtěte Monte Carlo integrací

$$\int_{x>0,y>0,z>0,x+y+z<1} \frac{1}{|\vec{r}-\vec{r}_0|}\mathrm{d}\vec{r}$$

kde $\vec{r}_0 = (1,1,1)$

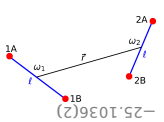
0.12522728...

Těžší. Vypočtěte Monte Carlo integrací druhý viriálový koeficient B_2 Lennard-Jonesova diatomiku ($\epsilon/k_{\mathrm{B}}T = 1$, $\sigma = 1$) pro vazebnou délku $L = \sigma$.

$$B_2 = -\frac{1}{2}\int\left[\exp\left(-\frac{u}{k_{\mathrm{B}}T}\right)-1\right]\mathrm{d}\vec{r}\,\frac{\mathrm{d}\omega_1}{4\pi}\frac{\mathrm{d}\omega_2}{4\pi}$$

$$u = u_{\mathrm{LJ}}(|\vec{r}_{1\mathrm{A}}-\vec{r}_{2\mathrm{A}}|) + u_{\mathrm{LJ}}(|\vec{r}_{1\mathrm{A}}-\vec{r}_{2\mathrm{B}}|) + u_{\mathrm{LJ}}(|\vec{r}_{1\mathrm{B}}-\vec{r}_{2\mathrm{A}}|) + u_{\mathrm{LJ}}(|\vec{r}_{1\mathrm{B}}-\vec{r}_{2\mathrm{B}}|)$$

Rady:
– $\mathrm{d}\vec{r} \to 4\pi r^2\mathrm{d}r$
– substituce $r = 1/w - 1$ (MC $\int$ bude přes $w \in (0,1)$)
– $\mathrm{d}\omega_i = \mathrm{d}\cos\theta_i\,\mathrm{d}\phi_i$ ($\cos\theta_i \in (-1,1)$, $\phi_i \in (0,2\pi)$)



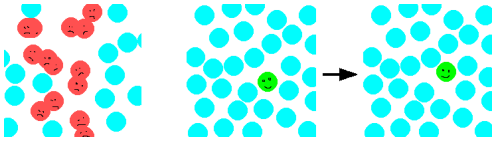
−25.1036(2)

Importance sampling

$$\langle f\rangle \approx \frac{\sum_{k=1}^{K} \mathrm{e}^{-\beta U(\vec{r}_k^N)} f(\vec{r}^N)}{\sum_{k=1}^{K} \mathrm{e}^{-\beta U(\vec{r}_k^N)}} \qquad \vec{r}_k^N = \text{random vector uniformly in the space (naive MC)}$$

$$\langle f\rangle \approx \frac{1}{K}\sum_{k=1}^{K} f(\vec{r}^{N,(k)}) \qquad \vec{r}^{N,(k)} = \text{random vector with a probability} \propto \mathrm{e}^{-\beta U(\vec{r}_k^N)}$$

Metropolisův algoritmus: generujeme postupně $\vec{r}^{N,(k+1)}$ z $\vec{r}^{N,(k)}$



naive MC (náhodné střílení) importance sampling („vzorkování podle důležitosti")

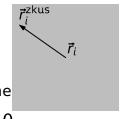
Metropolisova metoda (intuitivně)

- Vybereme částici, i (např. náhodně)
- Zkusíme s ní náhodně hýbnout, např.:

$$x_i^{\mathrm{zkus}} = x_i + u_{(-d,d)},$$
$$y_i^{\mathrm{zkus}} = y_i + u_{(-d,d)},$$
$$z_i^{\mathrm{zkus}} = z_i + u_{(-d,d)}$$

nebo na/v kouli, Gaussovsky,...

tak, že **pravděpodobnost opačného pohybu je stejná**

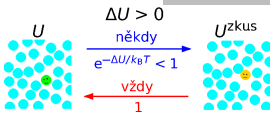


- Spočteme změnu potenciální energie, $\Delta U = U^{\mathrm{zkus}} - U$
- **Je-li** $\Delta U \le 0$, změnu přijmeme
  Je-li $\Delta U > 0$, změnu přijmeme s pravděpodobností $\exp(-\beta\Delta U)$, jinak odmítneme

Neboť pak pro poměr pravděpodobností platí:

nová : stará $= p^{\mathrm{zkus}} : p = \exp(-\beta\Delta U)$

(porovnáme pohyby tam a zpátky: vždy pohyb zmenšující U má pravděpodobnost 1 a opačný pohyb Boltzmannovu)



Trocha teorie: náhodné veličiny

Náhodná veličina $\mathcal{S}$ nabývá hodnoty z $\{A_i\}$, $i = 1,\ldots M$.
Dány jsou pravděpodobnosti $\pi(A_i) = \pi_i$.
Normalizace: $\sum_i \pi_i = 1$

Markovův řetězec je posloupnost $\mathcal{S}^{(k)}$, $k = 1,\ldots,\infty$ taková, že $\mathcal{S}^{(k+1)}$ závisí jen na $\mathcal{S}^{(k)}$, tj. matematicky:

$$\pi_j^{(k+1)} = \sum_{i=1}^{M} \pi_i^{(k)} W_{i\to j} \qquad \text{vektorově: } \boldsymbol{\pi}^{(k+1)} = \boldsymbol{\pi}^{(k)}\cdot\mathbf{W}$$

Normalizace:

$$\sum_{j=1}^{M} W_{i\to j} = 1 \quad \text{pro všechna } i$$

Příklad

Počítačová síť: $\begin{cases} 1. & \text{funguje} \\ 2. & \text{nefunguje} \end{cases}$

Když funguje: spadne s 10% pravděpodobností
(následující den nefunguje)
Když nefunguje: spraví ji s 30% pravděpodobností
(následující den funguje)

$$W = \begin{pmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$$

$$\lim_{k\to\infty} \boldsymbol{\pi}^{(k)} = (0.75,\ 0.25)$$

Výdělek: $\begin{cases} 2000 & \text{funguje} \\ 500 & \text{nefunguje} \end{cases}$

$$X = \begin{pmatrix} 2000 \\ 500 \end{pmatrix}$$

Průměrný výdělek $= \sum \pi_i X_i = \boldsymbol{\pi}\cdot\mathbf{X} = 1625$

for me: xoctave waits 3 s to switch desktop

Detailní rovnováha a mikroreversibilita

Hledám W, aby $\pi_i = \dfrac{\exp[-\beta U(A_i)]}{\sum_j \exp[-\beta U(A_j)]}$

W = stochastická matice, matice přechodu, pravděpodobnostní matice, Markovova matice...

Podmínky:

$$W_{i\to j} \ge 0 \quad \text{pro všechna } i,j = 1,\ldots,M$$
$$\sum_{j=1}^{M} W_{i\to j} = 1 \quad \text{pro všechna } i = 1,\ldots,M$$
$$\boldsymbol{\pi}\cdot\mathbf{W} = \boldsymbol{\pi} \quad \text{někdy „detailní rovnováha"}$$

$$\Uparrow$$

$$\pi_i W_{i\to j} = \pi_j W_{j\to i} \quad \text{mikroskopická reverzibilita (detailní rovnováha)}$$

Jestliže

- všechny stavy jsou dosažitelné z libovolného stavu v konečném čase s nenulovou pravděpodobností a
- žádný stav není periodický,

pak se množina stavů nazývá **ergodická** a pro libovolné počáteční rozložení pravděpodobností $\boldsymbol{\pi}^{(1)}$ **existuje limita** $\boldsymbol{\pi} = \lim_{k\to\infty}\boldsymbol{\pi}^{(k)}$

Metropolisova metoda (vědecky)

Jedno z mnoha řešení (Metropolis):

$$W_{i\to j} = \begin{cases} \alpha_{i\to j} & \text{pro } i \ne j \text{ a } \pi_j \ge \pi_i \\ \alpha_{i\to j}\dfrac{\pi_j}{\pi_i} & \text{pro } i \ne j \text{ a } \pi_j < \pi_i \\ 1 - \sum\limits_{k,\,k\ne i} W_{i\to k} & \text{pro } i = j \end{cases}$$

Ekvivalentní zápis:

$$W_{i\to j} = \alpha_{i\to j}\min\left\{1, \frac{\pi_j}{\pi_i}\right\} \quad \text{pro } i \ne j$$

kde matice $\alpha_{i\to j} = \alpha_{j\to i}$ popisuje zkušební změnu konfigurace

... algoritmus jsme již popsali

Algoritmus – detaily

- Zvolíme částici, kterou se bude hýbat, mřížkový bod, ...
- $A^{\text{zkus}} := A^{(k)}$ + změníme náhodně polohu (spin) vybrané částice
- $\Delta U := U(A^{\text{zkus}}) - U(A^{(k)}) \equiv U^{\text{zkus}} - U^{(k)}$
- Konfiguraci přijmeme ($A^{(k+1)} := A^{\text{zkus}}$) s pravděpodobností $\min\{1, \mathrm{e}^{-\beta\Delta U}\}$ v opačném případě odmítneme:

Varianta 1	Varianta 2	Varianta 3
$u := u_{(0,1)}$ IF $u < \min\{1, \mathrm{e}^{-\beta\Delta U}\}$ THEN $A^{(k+1)} := A^{\text{zkus}}$ ELSE $A^{(k+1)} := A^{(k)}$	$u := u_{(0,1)}$ IF $u < \mathrm{e}^{-\beta\Delta U}$ THEN $A^{(k+1)} := A^{\text{zkus}}$ ELSE $A^{(k+1)} := A^{(k)}$	IF $\Delta U < 0$ THEN $A^{(k+1)} := A^{\text{zkus}}$ ELSE $u := u_{(0,1)}$ IF $u < \mathrm{e}^{-\beta\Delta U}$ THEN $A^{(k+1)} := A^{\text{zkus}}$ ELSE $A^{(k+1)} := A^{(k)}$

- $k := k + 1$ a opakujeme od začátku

Volba částice

- V cyklu – pozor na reverzibilitu!
  Odstrašující příklady porušení mikroreverzibility:
  – Střídání pohybů u ternární směsi ze složek A, B, C systematicky v pořadí: A–B–C–A–B–C– ⋯
  – Střídání: pohyb molekuly A – pohyb molekuly B – změna objemu – atd.
- Náhodně

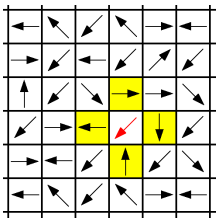
Chaos je lepší než špatné řízení

Metoda tepelné lázně +

vhodná pro mřížkové modely:

$$W_{i\to j} = \frac{\exp(-\beta U_j)}{\sum_{A_k \in \mathcal{C}_{\text{part}}} \exp(-\beta U_k)} \quad \text{pro } A_i, A_j \in \mathcal{C}_{\text{part}}$$

- $W_{i\to j}$ nezávisí na i
- interpretace: i získá hodnotu po termalizaci za daného okolí
- vyberu (zpravidla jeden) spin, množina jeho stavů je $\mathcal{C}_{\text{part}}$
- zvolím nový podle Boltzmannovy pravděpodobnosti, která závisí na okolí
- všechny potřebné hodnoty $W_{i\to j}$ (resp. vhodnou kumulativní distribuční funkci) mám předem tabelovánu pro všechna okolí

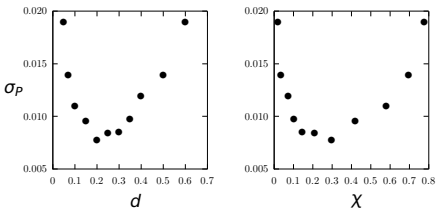


Zlomek přijatých konfigurací (acceptance ratio)

$$\chi = \frac{\text{počet přijatých konfigurací}}{\text{počet všech konfigurací}}$$

Závisí na délce kroku. Optimum závisí na systému, měřené veličině, algoritmu. Často **0.3 je dobrá volba**. Výjimky: řídké systémy...

$\chi = 0.3$



LJ (redukované jednotky): $T = 1.2$, $\rho = 0.8$

Cvičení

- Naprogramujte Metropolisův algoritmus pro pohyb jedné molekuly dusíku v homogenním tíhovém poli. Zvolte konstantní teplotu $T = 300$ K. Jaký je tlak ve výšce 8850 m, je-li u moře 1 bar? Stanovte také zlomek přijatých konfigurací.
  – potenciál molekuly je $u(z) = \begin{cases} \infty & \text{pro } z < 0 \\ mgz & \text{pro } z \geq 0 \end{cases}$, kde z je její nadmořská výška
  – zkušební pohyb použijte tvaru $z^{\text{zkus}} = z + \Delta z u_{[-1,1]}$
  – vhodné Δz je asi 30 km (viz níže)
  – na začátku dejte molekulu do výšky $z = 0$ a proveďte aspoň 20 kroků „míchání“
  – měřte aspoň 10000 kroků
  – stanovte počet případů, kdy byla molekula ve výšce v intervalech [0,100) a [8850,8950)
  – tlak je $p_{\text{moře}} \frac{\#([8850,8950))}{\#([0,100))}$

  0.377 bar

- Stanovte optimální velikost zkušebního posunutí Δz (resp. zlomku přijatých konfigurací χ) vzhledem k veličině střední výška molekuly $\langle z \rangle$. Tedy zvolte několik hodnot Δz (třeba 5, 10, 20, 30, 50, 100 km) a stanovte $\langle z \rangle$ včetně odhadu chyby $\sigma(z)$, např. blokovou metodou (např. 100 bloků po 100 MC krocích). Nakreslete graf závislosti $\sigma(z)$ na Δz resp. χ.

  cca. 30 km, $\chi = 0.3$

(Pseudo)náhodná čísla

$$r_i = F(r_{i-1}, r_{i-2}, \ldots, r_{i-m})$$

Požadavky:

- perioda generátoru (nejmenší číslo p takové, že $r_{i+p} = r_i$) je co nejdelší;
- rozdělení r_i je (v daném intervalu) rovnoměrné, speciálně: generují se správně i nejnižší bity;
- (r_i, r_{i+1}), trojice (r_i, r_{i+1}, r_{i+2}), atd. jsou nekorelované;
- to samé platí pro „všechny“ funkce f_i: páry $(f_0(r_i), f_1(r_{i-1}))$, trojice $(f_0(r_i), f_1(r_{i+1}), f_2(r_{i+2}))$, atd. jsou nekorelované;
- výpočet je rychlý.

Historický příklad špatného generátoru od IBM: $K(2^{16}+3, 2^{31})$

Generátory s posuvným registrem

feedback shift-register

$$R(A, B, C, \ldots): \quad r_i = r_{i-A} \oplus r_{i-B} \oplus r_{i-C} \oplus \ldots,$$

$\oplus$ = sčítání modulo 2 = XOR: $0 \oplus 0 = 1 \oplus 1 = 0$, $1 \oplus 0 = 0 \oplus 1 = 1$

Max. možná perioda je $2^{\max\{A,B,\ldots\}} - 1$

Generujeme slovo (32 nebo 64 bitů) najednou
Např. $R(108, 250)$, $R(471, 1586, 6988, 9689)$

Příklad. $R(5, 2)$:

1 krok:
5 4 3 2 1
1 1 0 1 1 0 $1 \oplus 1 = 0$

více kroků:
1101100011111001101001000010101 11011...

zde perioda $= 2^5 - 1 = 31$ (maximální)

Generátory s posuvným registrem +

Algoritmus:

```
CONST A=103
CONST B=205
CONST M=255  kde M je nejmenší číslo tvaru 2^k − 1 takové, že B ≤ M
INTEGER n  nezáporné celé číslo
INTEGER r[0..M]  pole, naplněno předem náhodnými čísly libovolného původu

jeden krok generující náhodné číslo (všechny bity):
n := n+1
r[n and M] := r[(n-A) and M] xor r[(n-B) and M]
        kde and a xor pracují po bitech
RETURN r[n and M]
```

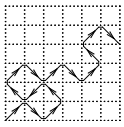
Kód je zvláště jednoduchý jako C/C++ makro:

```
#define rnd (++n, r[n&M] = r[(n-A)&M] ^ r[(n-B)&M])
```

Výhody: rychlé, matematická teorie pro periodu i korelace

Nevýhody: neprojde některými testy, např. náhodná procházka → → →

Náprava:
– kombinace dvou (stále rychlý)
– Mersenne twister (velmi kvalitní, populární)



Kongruenční generátory

$$K(C, M): \quad r_i = C r_{i-1} \bmod M$$

kde $A \bmod B$ je zbytek po dělení čísla A číslem B

$K(5^7, 2^{32})$: perioda $2^{32}/8$

$K(7^5, 2^{31}-1)$: perioda $2^{31}-2$

Příklad. $K(5, 31)$:
1 7 18 2 14 5 4 28 10 8 25 20 16 19 9 1 7 18 2 14 5 4 28 10 8 25 20 16 19 9 1 7 18 ...

Omezení korelací – kombinace dvou generátorů

Deklarujeme tabulku a naplníme ji náhodnými čísly pomocí generátoru č. 1

- vezmeme náhodně zvolené (index = náhodné číslo podle generátoru č. 2) číslo z tabulky
- použité číslo nahradíme novým náhodným (podle generátoru č. 1)

Jiná rozdělení (rozložení)

Obvykle je v knihovnách k dispozici náhodné číslo rnd(), rand(), random() rovnoměrně rozdělené v (0, 1) (nebo [0, 1) nebo [0, 1] – pozor!). Označíme ho $u_{(0,1)}$, distribuční funkce je:

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

Číslo rovnoměrně rozdělené v intervalu (a, b) je

$$u_{(a,b)} = a + (b-a) u_{(0,1)}.$$

Obecně: aplikace funkce $f(u)$ na $u_{(0,1)}$ →

$$\phi(y) = \sum_{x, f(x)=y} \frac{1}{|f'(x)|}$$

Opačně: chceme-li rozdělení dané funkcí $\phi(x)$, $\int \phi(x)\mathrm{d}x = 1$, musíme invertovat distribuční funkci $\int_{-\infty}^{y} \phi(x)\mathrm{d}x$.

Příklad: $x = -\ln u$ dává $\phi(x) = \exp(-x)$ (pozor na $u = 0$!)

Gaussovo normální rozdělení

$$u_{\text{Gauss}} = \sqrt{-2 \ln u_{(0,1)}} \cos(2\pi u_{(0,1)})$$

kde obě náhodná čísla $u_{(0,1)}$ jsou nezávislá, generátor tedy voláme dvakrát. Druhé nezávislé číslo získáme záměnou cos → sin.

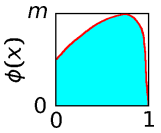
Přibližně:

$$u_{\text{Gauss}} \approx \sqrt{2}(u_{(0,1)} - u_{(0,1)} + u_{(0,1)} - u_{(0,1)} + u_{(0,1)} - u_{(0,1)})$$

Obecné rozdělení $\phi(x)$ na intervalu (a, b)

Když neumíme spočítat jako na předchozí stránce:

1. generuj $x = u_{(a,b)}$,
2. generuj $u = u_{(0,m)}$, kde m je maximum funkce $\phi(x)$ v intervalu (a, b),
3. je-li $u < \phi(x)$, přijmi hodnotu x a skonči, jinak znovu 1.



Vícedimenzionální rozdělení

V jednotkové kouli:

1. generuj $x = u_{(-1,1)}$, $y = u_{(-1,1)}$, $z = u_{(-1,1)}$,
2. spočítej $r^2 = x^2 + y^2 + z^2$,
3. je-li $r^2 < 1$, přijmi vektor (x, y, z) a skonči, jinak pokračuj bodem 1.

Na jednotkové sféře (povrchu koule): dělíme $\frac{r_{\text{v kouli}}}{r}$ (pozor na $r \approx 0$) nebo:

1. $z = u_{(-1,1)}$, $\phi = u_{(0,1)}$
2. $x = \sqrt{1 - z^2} \sin(2\pi\phi)$, $y = \sqrt{1 - z^2} \cos(2\pi\phi)$

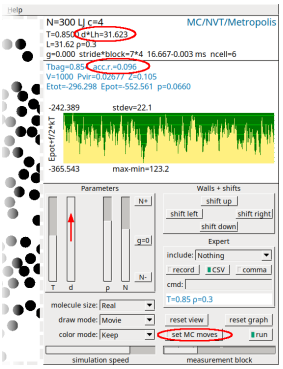
Rovnoměrné diskrétní rozdělení

$$u_N = \text{int}(N u_{(0,1)})$$

Raději ne takto (r je celé náhodné číslo):

$$u_N = r \bmod N$$

(špatně u kongruenčních generátorů – nejnižší bity nejsou náhodné)

SIMOLANT: Vyzkoušejte si MC sami

- Instalujte SIMOLANT podle návodu z minulé přednášky.
- Menu: Method → Monte Carlo NVT (Metropolis)
- Vypněte automatické nastavování délky zkušebního posunutí (set MC move). Objeví se slider (posuvník) "d".
- Měňte délku zkušebního posunu d a pozorujte, jak se mění zlomek přijatých konfigurací (acc.r.) a jak rychle se mění konfigurace.
- Snižte teplotu a zvyšte hustotu a opakujte změnu d. Porovnejte s MD s termostatem.
- Menu: Boundary conditions → Periodic, a nastavte kritickou teplotu a hustotu (přibližně $T = 0.85$ a $\rho = 0.3$) a alespoň $N = 300$ částic. Při jaké délce kroku se nejrychleji vzorkují fluktuace hustoty?