

**Jihočeská univerzita v Českých Budějovicích**

**Pedagogická fakulta – Katedra fyziky**

**Bakalářská práce**

**České Budějovice 2007**

**Tomáš Bürger**

**Jihočeská univerzita v Českých Budějovicích**

Pedagogická fakulta – Katedra fyziky

**Generování náhodných čísel a rozehrávání náhodných  
veličin**

Bakalářská práce

Vedoucí práce: Mgr. Petr Bartoš, Ph.D

Autor: Tomáš Bürger

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

Datum:

Touto formou děkuji mému konzultantovi p. Mgr. Petru Bartošovi Ph.D, za cenné rady a připomínky při zpracování této bakalářské práce.

## OBSAH

<b>1. Úvod</b> .....	6
<b>2. Základní pojmy z teorie pravděpodobnosti a matematické statistiky</b> .....	7
2.1 Zavedení náhodných veličin.....	7
2.2 Charakteristika náhodných veličin.....	9
2.3 Přehled nejdůležitějších statistických rozdělení.....	11
2.3.1 Vybrané diskrétní náhodné veličiny.....	11
2.3.2 Vybrané spojité náhodné veličiny.....	12
2.3.3 Vybrané limitní věty.....	14
<b>3. Počítačové modelování</b> .....	16
3.1 Počítačové modelování ve čtyřech krocích.....	16
3.2 Technika počítačového modelování.....	17
<b>4. Modelování Monte Carlo</b> .....	19
4.1 Základní techniky metody Monte Carlo.....	20
4.1.1 Generování náhodných čísel.....	20
4.1.2 Transformace náhodných veličin.....	24
4.1.3 Metoda Monte Carlo a matematická statistika.....	29
4.2 Řešení numerických problémů pomocí metody Monte Carlo.....	30
4.2.1 Výpočet určitých integrálů.....	30
4.2.2 Řešení Laplaceovy rovnice.....	35
4.2.3 Další problémy.....	38
4.3 Použití metody Monte Carlo ve fyzice.....	41
4.3.1 Transportní problém.....	42
4.3.2 Metoda nulové srážky.....	50
<b>5. Rozehrávání veličin</b> .....	52
5.1 Rozehrávání rychlostí s maxwellovským rozdělením rychlostí.....	52
5.2 Rozehrávání rychlostí s nemaxwellovským rozdělením rychlostí.....	56
<b>6. Závěr</b> .....	60
<b>7. Literatura</b> .....	62

## 1. Úvod

Jako téma mé bakalářské práce jsem si vybral generování náhodných čísel a rozehrávání náhodných veličin. Toto téma jsem si zvolil, protože mě zaujal jeho název a později po získání bližších informací od pana Mgr. Petra Bartoše Ph.D se mi toto téma jeví velmi zajímavé. V první kapitole se zabývám základními pojmy z teorie pravděpodobnosti matematiky, ve kterých využiji poznámek z předmětu statistické vyhodnocování experimentálních dat a poznámek z aplikované matematiky.

Ve druhé kapitole popisují počítačové modelování, které obsahuje základní čtyři kroky potřebné k řešení určitého fyzikálního problému. A obsahuje také do několika skupin rozdělené techniky počítačového modelování.

Třetí část mé práce zaujímá metoda Monte Carlo. V této části se seznamuji s metodou Monte Carlo. Poukazuji na základní techniky této metody. Dále se v této kapitole zmiňuji o řešení problémů pomocí Metody Monte Carlo. Na závěr této části se zabývám použitím metody Monte Carlo ve fyzice.

V poslední části mé práce rozehrávám veličiny s maxwellovským a nemaxwellovským rozdělením rychlosti. V této kapitole jsem čerpal ze zbytku mé bakalářské práce. Pro rozehrávání veličin jsem použil program MATLAB, se kterým jsem se naučil pracovat a ovládat základní funkce, které mi postačí pro rozehrávání těchto veličin.

Než začneme generovat náhodná čísla a rozehrávat náhodné veličiny vysvětleme některé důležité pojmy. Jsou to pojmy, počítačové modelování, statické rozdělení, metoda Monte Carlo atd. Nejprve se seznámíme se základními pojmy z teorie pravděpodobnosti a matematické statistiky, neboť na nich je Metoda Monte Carlo založena.

## **2. Základními pojmy z teorie pravděpodobnosti a matematické statistiky**

V této kapitole shrneme základní pojmy z teorie pravděpodobnosti a matematické statistiky. Začneme tedy u zavedení náhodných veličin.

### **2.1 Zavedení náhodných veličin**

V praxi se (mimo jiné) setkáváme s veličinami, u nichž nedovedeme určit jejich hodnotu v konkrétním případě. Takovým veličinám říkáme náhodné veličiny. Tyto náhodné veličiny dělíme na spojité a diskrétní. Diskrétní náhodné veličiny mohou nabývat spočetně (konečně nebo nekonečně) mnoha hodnot (jsou to čísla z konečného intervalu). Pro každou náhodnou veličinu se zavádí tzv. „zákon rozdělení náhodné veličiny“. Existují dvě varianty formulace tohoto zákona:

#### **a) Distribuční funkce**

Přiřazuje každému reálnému číslu pravděpodobnost, že náhodná veličina nabude hodnot menší než toto číslo. Máme-li náhodnou veličinu  $\xi$  a její distribuční funkci označíme  $F(x)$  potom:

$$F(x) = P\{\xi < x\} \quad (1)$$

Distribuční funkce nabývá hodnot z intervalu  $\langle 0, 1 \rangle$  a jestliže  $x_1 < x_2$ , potom

$$P\{x_1 \leq \xi < x_2\} = F(x_2) - F(x_1) \quad (2)$$

#### **b) Hustota pravděpodobnosti**

Distribuční funkce má integrální význam a shrnuje výslednou pravděpodobnost za určitý interval. Pokud chceme pracovat s konkrétními pravděpodobnostmi musíme zavést odpovídající diferenciální veličiny, například  $p(x)$

$$p(x) = \frac{dF(x)}{dx} \quad (3)$$

neboli

$$F(x) = \int_{-\infty}^x p(y) dy \quad (4)$$

Nyní si tyto údaje upřesníme pro diskrétní a spojité náhodné veličiny.

**Diskrétní náhodné veličiny:**

Zákon rozdělení náhodných veličin popisuje množinu hodnot  $x_i$  odpovídajících pravděpodobnosti  $p_i$ , kde  $p_i = P\{\xi = x_i\}$  a kde diskrétní náhodná veličina je dána funkcí

$$\xi = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix} \quad (5)$$

Hodnoty  $x_1, \dots, x_n$ , druhy definičního oboru pro pravděpodobnost  $p_1, \dots, p_n$  platí

$$p_i > 0, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n p_i = 1 \quad (6)$$

**Spojité náhodné veličiny:**

Spojité náhodné veličiny  $\xi$  nabývají hodnot  $x$  z nějakého otevřeného, nebo uzavřeného intervalu. Při použití distribuční funkce  $F(x)$ , kde bude náhodná veličina spojitá, dostaneme tvar úplné charakteristiky spojité náhodné veličiny

$$\xi: x \in \langle a, b \rangle, \quad p(x) \quad (7)$$

Nebo můžeme použít tzv. „hustotu pravděpodobnosti náhodné veličiny  $\xi$  v bodě  $x$ “. Hustota pravděpodobnosti  $p(x)$  má tyto vlastnosti:

a)  $p(x) \geq 0$ , pro všechna  $x \in \langle a, b \rangle$ ;

b)  $\int_{-\infty}^{\infty} p(y) dy = 1$ ;

c)  $P\{x_1 \leq \xi \leq x_2\} = \int_{x_1}^{x_2} p(y) dy$

S spojitými náhodnými veličinami lze provádět běžné matematické operace. Například spojité náhodné veličiny mohou být argumenty běžné matematické funkce  $f(x)$  a výsledkem je nová náhodná veličina  $\eta = f(\xi)$  (náhodná funkce). Nebo můžeme pracovat i s vícerozměrnými náhodnými veličinami, kde pro popis pravděpodobnosti použijeme slouženou distribuční funkci, která má například ve dvou rozměrech tvar



$$F(x, y) = P\{x < \xi, \eta < y\}. \quad (8)$$

## 2.2 Charakteristika náhodných veličin

Danou náhodnou veličinu zcela popisují vztahy (5) či (7). Někdy však vystačíme jen se základními vlastnostmi náhodné veličiny, protože použitá data jsou zatížena například šumem. Tento šum vytváří pozorovaný náhodný proces a proto je zbytečné tento šum popisovat. Z tohoto důvodu byl navržen systém **momentů náhodných veličin**. Každý moment je číslo, které v sobě obsahuje část informace o náhodné veličině. Úplná informace by musela být popsána pomocí nekonečné soustavy momentů. My se omezíme na popis pomocí konečného počtu momentů. Protože momenty tvoří hierarchickou posloupnost (od významnějších k méně významným), dostane odříznutím vyšších členů náhodné veličiny popis nejpodstatnějších rysů.

V matematice se používá větší počet posloupností momentů, my si tu uvedeme ty nejužívanější.

### a) Charakteristika polohy

První moment každé náhodné veličiny  $\xi$  je její střední hodnota s ohledem na rozdělení pravděpodobností a značí se  $E\xi$ . Nazýváme jí očekávaná hodnota a vztahy pro diskrétní a spojitou náhodou veličinou jsou

$$E\xi = \sum_i x_i \cdot p_i \quad (9)$$

$$E\xi = \int_{-\infty}^{\infty} x \cdot p(x) dx. \quad (10)$$

Je-li  $c$  číselná konstanta a symboly  $\xi$  a  $\eta$  náhodné veličiny, potom bude platit

$$Ec = c$$

$$E(\xi + c) = E\xi + c$$

$$E(c \cdot \xi) = c \cdot E\xi$$

$$E(\xi + \eta) = E\xi + E\eta.$$

Když veličiny  $\xi$  a  $\eta$  jsou nezávislé, potom platí

$$E(\xi \cdot \eta) = E\xi \cdot E\eta.$$

### b) Charakteristiky variability

Další moment se nazývá *rozptyl*, *variace* nebo *disperze* a značí se  $D\xi$ . Udává rozptyl možných hodnot náhodné veličiny  $\xi$  kolem její střední hodnoty  $E\xi$  a je dán vztahem

$$D\xi = E(\xi - E\xi)^2. \quad (11)$$

Definiční vztah (10) je nepraktický pro výpočet, proto se častěji používá

$$D\xi = E(\xi^2) - (E\xi)^2. \quad (12)$$

Je-li  $c$  konstanta a  $\xi$  a  $\eta$  náhodné veličiny, potom pro rozptyl platí

$$Dc = 0$$

$$D(\xi + c)$$

$$D(c \cdot \xi) = c^2 \cdot D\xi.$$

a pro nezávislé náhodné veličiny  $\xi$  a  $\eta$  platí

$$D(\xi + \eta) = D\xi + D\eta$$

$$D(\xi - \eta) = D\xi + D\eta.$$

Odmocnina z rozptylu je směrodatná odchylka pro níž platí

$$\sigma_\xi = \sqrt{D\xi}.$$

### c) Charakteristika vyšších řádů

Existuje-li  $E\xi^k$  a má konečnou hodnotu, definujeme *centrální moment  $k$ -tého řádu*

$$\mu_k = E(\xi - E\xi)^k, \quad k = 0, 1, 2, \dots \text{ Pro } k=2 \text{ dostaneme } \mu_2 = D\xi$$

Momenty vyšších řádů vlivem mocniny v definičním vztahu buď prudce klesají, nebo rostou (v závislosti na vztahu prvního momentu prvního řádu k hodnotě 1). Tato vlastnost je nepraktická pro praktické využití, proto je zvykem momenty normalizovat:

- Normováním momentu 3.řádu definujeme šikmost

$$\alpha_3 = \frac{\mu_3}{\sigma^3}. \quad (13)$$

U symetrického rozdělení je tato charakteristika nulová. Je-li kladná je zešikmena doleva, záporná doprava.

- Momentem 4. řádu definujeme špičatost

$$\alpha_4 = \frac{\mu_4}{\sigma^4}. \quad (14)$$

Hodnoty  $\alpha_4$  nabývá pro Gaussovské rozdělení, je-li  $\alpha_4 > 3$ , potom je studované rozdělení „špičatější“ než rozdělení normální, pro  $\alpha_4 < 3$  je rozdělení „plošší“.

V konkrétních případech použijeme pro momenty vyšších řádu vztahy

$$\mu_3 = E\xi^3 - 3 \cdot E\xi^2 \cdot E\xi + 2 \cdot (E\xi)^3 \quad (15)$$

$$\mu_4 = E\xi^4 - 4 \cdot E\xi^3 \cdot E\xi + 6 \cdot E\xi^2 \cdot (E\xi)^2 + 3 \cdot (E\xi)^4 \quad (16)$$

## 2.3 Přehled nejdůležitějších statistických rozdělení

Jak již bylo uvedeno můžeme statistické rozdělení rozdělit na diskrétní a spojitá.

### 2.3.1 Vybrané diskrétní náhodné veličiny

#### a) Rovnoměrné rozdělení

Náhodná veličina s rovnoměrným rozdělením nabývá  $m$  hodnot  $1, 2, \dots, m$  se stejnými pravděpodobnostmi

$$p = P\{\xi = x\} = \frac{1}{m} \quad (17)$$

První dva momenty této veličiny jsou

$$E\xi = \frac{m+1}{2}$$

$$D\xi = \frac{m^2 - 1}{12}$$

#### b) Binomické rozdělení

Tímto rozdělením se řídí četnost nějakého jevu v  $n$  nezávislých pokusech, když v každém pokusu má výskyt jevu pravděpodobnost  $p$ , kde  $n$  je přirozené číslo a  $p \in (0,1)$ .

$$\xi = \begin{pmatrix} 0 & 1 & \dots & n \\ p_0 & p_1 & \dots & p_n \end{pmatrix} \quad (18)$$

$$p_x = P\{\xi = x\} = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}$$

Základní charakteristiky binomického rozdělení jsou

$$E\xi = np$$

$$D\xi = np(1-p)$$

$$\alpha_3 = \frac{1-2p}{\sqrt{np(1-p)}}$$

$$\alpha_4 = \frac{1 - 6p(1-p)}{np(1-p)} + 3$$

### c) Poissonovo rozdělení

Studujeme četnost nějakého jevu v mnoha pokusech, přičemž výskyt tohoto jevu v jednotlivých pokusech je velmi málo pravděpodobný. Jediným parametrem Poissonova rozdělení je  $n \cdot p = \lambda$ .

Pravděpodobnost lze vyjádřit funkčním předpisem

$$p_x = P\{\xi = x\} = e^{-\lambda} \cdot \frac{\lambda^x}{x!}.$$

Základní charakteristiky Poissonova rozdělení jsou

$$E\xi = \lambda$$

$$D\xi = \lambda$$

$$\alpha_3 = \frac{1}{\sqrt{\lambda}}$$

$$\alpha_4 = \frac{1}{\lambda} + 3$$

Poissonova náhodná veličina s parametrem  $\lambda = \lambda_1 + \lambda_2$  je součet dvou nezávislých Poissonových náhodných veličin s parametry  $\lambda_1$  a  $\lambda_2$ .

### 2.3.2 Vybrané spojité náhodné veličiny

Se spojitými náhodnými veličinami se ve fyzice setkáváme častěji, než s diskrétními.

#### a) Rovnoměrné rozdělení

Spojitá náhodná veličina  $\xi$  z intervalu  $\langle a, b \rangle$  má rovnoměrné rozdělení tehdy, má-li v tomto intervalu konstantní hustotu pravděpodobnosti, pro kterou platí

$$p(x) = \frac{1}{a-b} \quad x \in \langle a, b \rangle \quad (19)$$

Základní charakteristiky rovnoměrného rozdělení jsou

$$E\xi = \frac{a+b}{2}$$

$$D\xi = \frac{(b-a)^2}{12}$$

$$\alpha_3 = 0$$

$$\alpha_4 = \frac{9}{5}.$$

Toto rozdělení je velmi důležité při řešení problémů metodou Monte Carlo, neboť náhodné veličiny popisující studovaný jev se obvykle získávají vhodnou transformací rovnoměrně náhodné veličiny intervalu  $\langle 0,1 \rangle$ . Tuto veličinu budeme označovat jako  $\gamma$ .

### b) Gaussovo rozdělení (normální)

Gaussovo rozdělení je nejdůležitější z rozdělení spojitě náhodné veličiny a má velký význam ve statistické matematice. Používá se tam, kde na popis daného jevu působí dočasně více nepatrných a nezávislých jevů.

Gaussovo rozdělení je spojitou náhodnou veličinou s parametry  $\sigma$  a  $\mu$ , přičemž hustota pravděpodobnosti je ve tvaru

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad x \in (-\infty, \infty) \quad (20)$$

a distribuční funkce

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt.$$

Základní charakteristiky Gaussovo rozdělení jsou

$$E\xi = \mu$$

$$D\xi = \sigma^2$$

$$\alpha_3 = 0$$

$$\alpha_4 = 3.$$

Gaussova náhodná veličina se označuje písmenem  $N$ . U gaussovského rozdělení se často používá *pravidlo 3 sigma*, které říká „ Při jednom pokusu je prakticky nemožné získat hodnotu  $\zeta$  lišící se od střední hodnoty o více než  $3\sigma$ “. Pro Gaussovo náhodnou veličinu  $\zeta$  plyne

$$P\{\mu - 3\sigma < \zeta < \mu + 3\sigma\} = 0,9973.$$

### c) Maxwellovo rozdělení

Používá se například v kinetické teorii plynů a hustota pravděpodobnosti je dána předpisem

$$p(x) = \frac{2}{a^3 \sqrt{2\pi}} \cdot x^2 \exp\left(-\frac{x^2}{2a^2}\right) \quad x \in (-\infty, \infty). \quad (21)$$

Základní charakteristiky Maxwellova rozdělení jsou

$$E\xi = \frac{3a}{\sqrt{2}}$$

$$D\xi = 3a^2$$

s parametrem  $a > 0$ .

### 2.3.3 Vybrané limitní věty

Z velkého množství vět matematických si uvedeme jen ty, na nichž je založeno statistické vyhodnocování experimentálních dat a modelování reálných procesů metodou Monte Carlo.

#### a) Zákon velkých čísel

Budou-li  $\xi_1, \xi_2, \dots, \xi_n$  nezávislé náhodné veličiny se stejným rozdělením, tj. se stejnými středními hodnotami  $\mu$ , pak jejich aritmetický průměr

$$\bar{\xi} = \frac{1}{n} \cdot \sum_{i=1}^n \xi_i. \quad (22)$$

Konverguje podle pravděpodobnosti k  $\mu$ . To znamená, že pro každé  $\varepsilon$  platí

$$\lim_{n \rightarrow \infty} P\left\{\left|\frac{1}{n} \cdot \sum_{i=1}^n \xi_i - \mu\right| < \varepsilon\right\} = 1.$$

S využitím vlastností rozptylu náhodné veličiny  $\xi$  můžeme určit i rozptyl aritmetického průměru  $\bar{\xi}$

$$D\bar{\xi} = \frac{D\xi_1 + \dots + D\xi_n}{n^2} = \frac{\sigma^2}{n}. \quad (23)$$

Směrodatná odchylka  $\sqrt{D\bar{\xi}}$  pak bude úměrná druhé mocnině z počtu pozorovaných  $n$ .

**b) Centrální limitní věta počtu pravděpodobnosti**

Budou-li  $\xi_1, \xi_2, \dots, \xi_n$  nezávislé náhodné veličiny se stejným rozdělením, které má střední hodnotu rozptylu  $\mu$  a rozptyl  $\sigma^2$ , pak jejich součet má pro velká  $n$  přibližně Gaussovo rozdělení s parametry  $N(n\mu, n\sigma^2)$ .

### **3. Počítačové modelování**

Počítačové modelování nám představuje základní směr počítačové fyziky. V literatuře se můžeme s počítačovým modelováním setkat také pod názvem jako je počítačové simulace nebo počítačové experimenty. Řešení určitého fyzikálního problému za použití počítačového modelování lze rozdělit do čtyř kroků.

#### **3.1 Počítačové modelování ve čtyřech krocích**

Počítačové modelování jsem si rozdělili do čtyřech kroků a jimi jsou: Formulace problému, vytvoření modelu, řešení modelu a srovnání výsledků modelování s experimentálními údaji.

##### **a) Formulace problému**

Nejprve analyzujeme studovaný problém, zvážíme význam jednotlivých fyzikálních jevů a uvědomíme si vzájemné souvislosti.

##### **b) Vytvoření modelu**

Podle studovaného jevu navrhne model. Model je aproximací studovaného jevu, proto:

1. Model je prakticky vždy jednodušší než studovaný jev

Model je prakticky vždy jednodušší než studovaný jev tehdy, když všechny vlastnosti studovaného jevu neznáme, a nebo by jsem jinak nebyli schopni úlohu vyřešit. Poté dochází k velkému zjednodušení při tvorbě modelu.

2. Při formulaci modelu nemáme jistotu o jeho správnosti

Při vytváření studovaného jevu se snažíme co nejvíce zjednodušovat. Proto zahrnujeme jen základní rysy, ostatní vypouštíme. Nevýhoda tohoto zjednodušení je ta, že my nevíme přesně jaké rysy jsou podstatné a jaké nepodstatné, proto musíme výsledky srovnávat s experimentálními daty .

3. Výsledky počítačového modelování vypovídají pouze o modelu, nikoliv o studovaném jevu

Výsledek který získáme vypovídá pouze o modelu. Můžou nám nastat problémy a to v případě, jestliže jsem model příliš zjednodušili a nebo špatně zformulovali, což poznáme ve výsledcích. Přílišné zjednodušení a špatné zformulování se nám bude



promítat ve výsledcích i při hodně přesném modelování, protože výsledek modelování je dán přesností dat na vstupu.

### **c) Řešení modelu**

Podle techniky modelování si zvolíme numerické metody. Nejčastěji se jedná o řešení soustavy obyčejných nebo parciálních diferenciálních rovnic.

### **d) Srovnání výsledků modelování s experimentálními údaji**

Pro zlepšení přesnosti počítačového modelování se do počítačového experimentu zabuduje „zpětná vazba“. Zpětná vazba znamená, že výsledek modelování se porovnává s výsledky přímého měření. Zpětnou vazbou také zjistíme, jestli jsme se nedopustili chyb při řešení modelu. Správný test modelu je založen na nezávislých datech, která jsem při navrhování modelu znali a která nám posléze vyjdou jako výsledek modelování. Pokud nám toto nevyjde musíme model opravit a projít znovu kroky **b**, **c** a **d**.

Nejdůležitější práce je v přechodu mezi kroky **a** a **b**, což je správné naformulování modelu.

## **3.2 Techniky počítačového modelování**

Techniky počítačového modelování můžeme rozdělit do několika skupin:

1. Časticové techniky
2. Spojité modelování
3. Hybridní modelování

### **1. Časticové techniky (mikroskopické)**

Časticová technika nám poskytuje přesnější výsledky na úkor velké spotřeby strojového času.

Základem této techniky je detailně studovat chování částic (atomů, iontů, elektronů, ...), ze kterých se studovaný jev skládá. Při testování chování modelu použijeme statické vyhodnocení dat, protože nejsme schopni poskytnout například informaci o poloze a rychlosti všech atomů ve studovaném souboru, čímž ztrácíme část informací z modelování.

Podle toho, jak chování části studovaného jevu popisujeme rozeznáváme metody:

a) Metoda Monte Carlo

Chování jednotlivých částic popisujeme stochasticky pomocí zákonů počtu pravděpodobnosti.

b) Metoda molekulární dynamiky

Jedná se o deterministickou metodu, kde numericky řešíme pohybové rovnice popisující studovaný soubor částic (řešení obyčejných diferenciálních rovnic).

c) Hybridní techniky

Kombinace dvou předchozích tak, aby výpočet byl co nejefektivnější.

## **2. Spojité modelování (makroskopické)**

Spojité modelování je popis studovaného jevu na makroskopické úrovni, kde platí zákony o zachování- energie, hybnosti, náboje, apod. Tyto zákony jsou popisovány parciálními diferenciálními rovnicemi. Spojité modelování přináší jen hrubé, ale zato rychlé výsledky.

## **3. Hybridní modelování**

Používáme kombinace obou dvou předchozích technik (částicové a spojité) s cílem zachovat rychlost výpočtu spojitěho přístupu a přesnost zvýšit částicovou technikou.

#### **4. Metoda Monte Carlo**

Metoda Monte Carlo označuje souhrn postupů dovolujících pomocí mnohonásobných náhodných pokusů získat řešení v oblasti vědy. Mezi tvůrce této metody patří J. von Neumann, S.A. Ulam, N. Metropolis, H. Kahn a E. Fermi. V roce 1949 poprvé užívali pojem „Monte Carlo“ Metropolis a Ulman ve své publikaci [1]. Nejstarší známou aplikací metody Monte Carlo je úloha o Buffonově jehle [2] (obr.1), která slouží k určení hodnoty čísla  $\pi$ . Tato úloha se datuje k roku 1777.



**Obrázek 1**

Formulace problému s Buffonovou jehlou (obr.1): Na vodorovnou desku nakresleme rovnoběžky vzdálené od sebe ve vzdálenosti  $d$ . Jehlu o délce  $l \leq d$  náhodně hodíme na desku. Výsledek pokusu je úspěšný tehdy, protne-li jehla jednu z čar, jinak je pokus neúspěšný. Pokus  $N$ -krát opakujeme a určíme  $\frac{M}{N}$ , kde  $M$  nám udává počet úspěšných pokusů. Nalezením poměru  $\frac{M}{N}$  experimentálně určíme konstantu  $\pi$ . Výsledek této metody je však při konečném počtu hodnot zatížen chybou: při deseti pokusech jsme schopni například odhadnout hodnotu  $\pi$  jako 3, při 1 000 pokusech jako 3,1, při 100 000 pokusech jako 3,14 atd.

##### **a) Analýza problému a vytvoření modelu**

Protože se jedná o stochastickou metodu modelování, musíme popsat zkoumaný jev pomocí náhodné veličiny. Vytvoření modelu potom znamená zjednodušeně popsat zkoumaný jev náhodnou veličinou s daným oborem hodnot a rozdělení pravděpodobnosti a určit která charakteristika náhodné veličiny obsahuje hledanou odpověď.

##### **b) Generování náhodné veličiny**

Vytvoření náhodné veličiny na počítači se obvykle provádí ve dvou krocích. Nejprve nageneryjeme na počítači náhodnou veličinu s pevně daným rozdělením pravděpodobností. Potom veličinu transformujeme v bloku c).

### c) Transformace náhodné veličiny

Podle požadavku modelu přetransformujeme veličinu v hledanou náhodou veličinu.

### d) Opakování kroků *b* a *c* a statistické vyhodnocení výsledků

Předchozími kroky dostaneme pouze jednu realizaci hledané náhodné veličiny (stochastickou metodou). Proto se tyto kroky mnohokrát opakují. Poté, co dostane potřebný počet konkrétních realizací naší náhodné veličiny  $\xi_1 \dots \xi_N$ , podrobíme tento statistický soubor analýze a dostane hledanou odpověď. Chyba metody Monte Carlo klesá odmocninou z počtu pokusů tj.:

$$\vartheta = \frac{1}{\sqrt{N}} \quad (24)$$

Jak už jsme si uvedli, metoda Monte Carlo se skládá z kroku *a*) (tvůrčí část) a kroku *b*), *c*) a *d*) (rutinní část).

## 4.1 Základní techniky metody Monte Carlo

Základní schéma řešení metody Monte Carlo se skládá ze čtyř kroků. První krok je z pravidla nejobtížnější, tj. vytvoření vhodného modelu. Zbylé tři nám představují „rutinní postupy“, při kterých však mohou nastat problémy.

Statistické vyhodnocení výsledku (čtvrtý krok) bychom měli být schopni vyřešit na základě poznatků z počtu pravděpodobnosti a matematické statistiky (viz. 1). Modely jsou naformulovány přibližně tak, že pokud hledáme jedno číselný výsledek, nalezneme jej jako první nebo druhý moment námi zvolené náhodné veličiny  $\xi$ ,  $E\xi$  nebo  $D\xi$ . Tyto momenty umíme podle zákona velkých čísel (22) přibližně odhadnout. Nyní si probereme jak budeme realizovat druhý a třetí krok základního schématu.

### 4.1.1. Generování náhodných čísel

Výpočetní schéma metody Monte Carlo předpokládá modelování náhodného procesu pomocí operace s náhodnými čísly. Pro metodu Monte Carlo byly navrženy a použity tyto způsoby:

- Fyzikální generátory
- Tabulky náhodných čísel
- Vypočítaná náhodná čísla

Na začátku byla známa jediná technika, a to použití samostatného zařízení produkující náhodné veličiny, poté přenesení takto vytvořených čísel do počítače. Mezi takovými zařízeními patří ruleta, odkud je také název Monte Carlo. Od začátku šedesátých let se používaly fyzikální procesy náhodné svou podstatou. Jednalo se o radioaktivní rozpad nebo tzv. výstřelkový šum v elektronkách. Mezi jejich nevýhody patří jejich pomalost. Další metoda je používání tabulky náhodných čísel. Vytváří se tak, že fyzikální generátory udělají zásobu náhodných čísel, která jsou uložena na vnějším medium (tabulky na papíře, magnetická páska, CD...), ze které posléze čerpají. U této metody se s zlepšující se technikou začala projevovat její pomalost, proto přišla náhrada za fyzikální generátory a to v podobě tzv. „vypočítaných náhodných čísel“. Toto řešení se používá dodnes, avšak sám pojem obsahuje protimluvu – náhodnost nelze žádným deterministickým postupem vypočítat, o čemž John von Neumann (1951) řekl:

*„Každý, kdo se zabývá aritmetickými metodami vytváření náhodných čísel, se nepochybně dopouští hříchu“.*

Ve skutečnosti jsou to čísla *pseudonáhodná* (posloupnost čísel méně či více podobající se náhodným číslům). Od skutečných čísel se odlišují ve dvou rysech:

- Po uplynutí periody  $p$  se posloupnost začíná opakovat.
- V jedné periodě se čísla vyskytují relativně náhodně, tato náhodnost není dokonalá.

Cílem teorie generátorů je vybrat algoritmus jejich generování tak, aby oba rysy co nejméně rušily a perioda  $p$  byla co nejdelší. Nejprve je vybrán generátor na základě teorie, poté dlouhodobě testován (zda jsou v něm náhodně generována jednotlivé číslice, pak dvojice atd...). K testování se používá  $x^2$  test.

Postupně byla vytvořena celá řada algoritmů s dobrými statistickými vlastnostmi. Některé algoritmy vedou na rekurentní vzorec prvního řádu

$$\xi_{i+1} = a\xi_i + b \pmod{M}. \quad (25)$$

Častěji se však používají rekurentní vzorce vyšších řádů, které nové pseudonáhodné číslo  $\xi_{i+1}$  určují na základě operací s  $k+1$  předchozími čísly  $\xi_i, \xi_{i-1}$  až  $\xi_{i-k}$ . Perioda generátoru závisí na jeho řádu  $k$  podle vztahu

$$p \sim M^k$$

Většina generátorů pseudonáhodných čísel vytváří jednotlivé hodnoty náhodné veličiny  $\gamma$  v rozmezí  $(0,1)$ . Ve skutečnosti existují dvě třídy vypočítaných náhodných čísel:

- a) čísla pseudonáhodná
- b) čísla kvazináhodná

Obě skupiny generátorů se snaží vytvořit náhodná čísla  $\gamma$  rovnoměrně rozdělená na intervalu  $(0,1)$ , ale výrazně se liší ve stupni náhodnosti vytvořené posloupnosti.

### a) Čísla pseudonáhodná

Pseudonáhodná čísla jsou mnohem běžnější než čísla kvazináhodná. Generátory pseudonáhodných čísel jsou založeny na jednom ze tří principů:

- von Neumanovy generátory
- lineární kongruenční generátory
- generátory s posuvnými registry

#### - von Neumanovy generátory

Tyto generátory se používaly jen v samých počítačích od roku 1946. Jsou založeny na manipulaci s jednotlivými číslicemi. Protože jejich periody jsou příliš krátké a náhodnost nedostatečná, mají už jen historickou cenu.

#### - Lineární kongruenční generátory

Jsou to nejpoužitelnější generátory založené na vztahu pro nalezení nového pseudonáhodného čísla  $\xi_{i+1}$  na základě již známých čísel  $\xi_i$  až  $\xi_{i-k}$

$$\xi_{i+1} = a_0 \cdot \xi_i + a_1 \cdot \xi_{i-1} + \dots + a_k \cdot \xi_{i-k} + b \pmod{M} \quad (26)$$

kde  $k \geq 0$ ,  $a_k$ ,  $b$  a  $M$  jsou to konstanty, jejichž vhodnou volbou určujeme konkrétní generátor. Vytvořená pseudonáhodná čísla leží v intervalu  $0$  až  $M-1$  (díky operaci „modulo“). Na interval  $0$  až  $1$  se převede vydělením hodnotou  $M$ , poté takto vzniklé číslo  $\gamma$  leží v intervalu  $(0,1)$ . Pro nastartování takového generátoru musíme použít tzv. násadu, což jsou hodnoty čísel  $\xi_i$ ,  $\xi_{i-1}$  až  $\xi_{i-k}$ . Všechna čísla násady musí být menší než hodnota  $M$ . Násadu musíme měnit, jinak dostaneme vždy stejnou pseudonáhodnou posloupnost čísel, což se využívá jen při ladění. Další nechtěný jev je, že začátek neregenerované posloupnosti pseudonáhodných čísel není dost náhodný, proto se používá zahřívání generátoru, kdy se několik prvních desítek neregenerovaných čísel nepoužije. Konkretizací konstant můžeme vytvořit tři zjednodušené typy generátorů:

- Multiplikativní generátory (od roku 1951- Lehmer)

Jsou založeny na rekurentním vztahu prvního řádu s nulovou hodnotou konstanty  $b$

$$\xi_{i+1} = a_0 \cdot \xi_i \pmod{M}. \quad (27)$$

Tyto generátory jsou dostatečně rychlé, ale jejich kvalitu musíme zvažovat. Pokud použijeme nevhodnou volbu konstant, kvalita generátoru prudce klesá. Příklad multiplikativního generátoru:

$$\begin{aligned} \xi_{i+1} &= 7^5 \cdot \xi_i \pmod{2^{31}-1} \\ \xi_{i+1} &= 630360016 \cdot \xi_i \pmod{2^{31}-1} \end{aligned}$$

Tyto dva generátory používala v 70. a 80. letech firma IBM ( IBM 360, IBM 370)

- Aditivní generátory

Aditivní generátory jsou založeny na vztahu

$$\xi_{i+1} = \xi_{i-k_1} + \xi_{i-k_2} \pmod{M}. \quad (28)$$

Jednoduchý příklad tohoto typu je

$$\xi_{i+1} = \xi_{i-1} + \xi_{i-2} \pmod{3137}.$$

Operace modulo se provádí jednoduchým testem, zda vzniklé číslo je menší než  $M$  a pokud ne, tak se  $M$  odečte, proto je tento generátor velmi rychlý a dosti kvalitní. Nevýhodou je velmi krátká perioda ( $9,8 \cdot 10^9$ ). Proto se používá pro ladění, ale jeho vylepšená varianta se používá už i profesionálně.

- Smíšené generátory

Smíšené generátory jsou založeny na lineárním rekurentním vztahu

$$\xi_{i+1} = a \cdot \xi_i + b \pmod{M}. \quad (29)$$

Příkladem je generátor

$$\xi_{i+1} = 69069 \cdot \xi_i + 1 \pmod{2^{32}}.$$

### Generátory s posuvnými registry

Jsou svou podstatou předurčeny pro přímou hardwarovou realizaci. Existují jednoúčelové počítače obsahující integrovaný obvod, přímo generující pseudonáhodná čísla a takový obvod je konstruován podle algoritmu pro generátor s posuvným registrem

$$b_{i+1} = c_0 \cdot b_i + c_1 \cdot b_{i-1} + \dots + c_k \cdot b_{i-k} \pmod{2}. \quad (30)$$

Tento vztah je podobný vztahu pro lineární kongruenční generátor. Základní rozdíl je ve významu konstant:  $k \geq 1$ ,  $c_k$  nabývají hodnot 0 nebo 1 (nejméně dvě hodnoty musí být nenulové) a generátor pracuje místo s celými čísly pouze s bity  $b$ . Při praktickém použití

se pro zvýšení kvality generátoru volí konstanta  $k$ . Často se pracuje s lineárními generátory, kde pouze dvě hodnoty  $c$  jsou nenulové, například

$$b_{i+1} = b_{i-470} + b_{i-9688} \pmod{2},$$

kde vstupní posloupnost bitů musí mít délku 9689. Perioda takového generátoru může dosahovat hodnoty až  $2^{9689}-1$ . V literatuře [3] můžeme nalézt několik dalších typů generátorů s posuvnými registry.

Jsou snahy kvalitu generátorů pseudonáhodných čísel zvýšit, a to jak jejich periodu tak i náhodnost. Proto se dělá kombinace více generátorů. Jednou z možných cest je jedním generátorem naplnit množinu pseudonáhodných čísel a druhým generátorem z této množiny „losovat“ konkrétní číslo. První generátor nahradí stávající číslo novým a postup se opakuje. Takto se zvýší náhodnost generované posloupnosti, ale zpomaluje se výpočet.

#### **b) Kvazináhodná čísla**

Posloupnost těchto čísel se vůbec nepodobá náhodným číslům. Základní princip generátorů kvazináhodných čísel je založen na požadavku, co nejrovnoměrněji pokrýt interval  $(0,1)$ . Skutečná čísla pokrývají úsečku  $(0,1)$  rovnoměrně, ale s náhodnými fluktuacemi, které u kvazináhodných čísel nebudou.

Generátory kvazináhodných čísel jsou založeny na libovolném prvočíslu. Například na základě prvočísla 2 si ukažme několik prvních hodnot

$$1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, 1/16, \dots$$

Kvazináhodné číslo nelze použít tam, kde požadujeme náhodnost a nezávislost dvou po sobě jdoucích náhodných čísel (souřadnice náhodného bodu), neboť kvazináhodná čísla jsou v tomto ohledu přísně determinována. Jak již víme chyba metody Monte Carlo je  $1/\sqrt{N}$  (24). Tento vztah je platný při použití skutečně náhodných i pseudonáhodných čísel. Při použití kvazináhodných čísel se tento vztah zrychluje na téměř  $1/N$ .

#### **4.1.2 Transformace náhodných veličin**

Dalším krokem výpočetního schématu metody Monte Carlo je transformace náhodné veličiny  $\gamma$  s rovnoměrným rozdělením na intervalu  $(0,1)$ , kterou nám poskytuje generátor náhodných čísel. Existuje velké množství algoritmů, které si můžeme rozdělit na dvě skupiny:

a) algoritmy specializované pro konkrétní náhodné veličiny  $\xi$  (velký počet)



b) algoritmy obecné, jimiž lze vyřešit množství problémů

Máme k dispozici spojitou náhodnou veličinu  $\gamma$ , rovnoměrně rozdělenou v intervalu  $(0,1)$ . Parametry jejího rozdělení jsou

$$E\gamma = \frac{1}{2}$$

$$D\gamma = \frac{1}{12}.$$

a) Algoritmy specializované pro konkrétní náhodné veličiny  $\xi$  ( velký počet)

Při konkrétní realizaci na počítači získáme  $m$ - místná diskrétní čísla  $\gamma$ , která budou mít momenty

$$E\gamma' = \frac{1}{2} \cdot (1 - 2^{-m})$$

$$D\gamma' = \frac{1}{12} \cdot (1 - 2^{-2m}).$$

Při práci s dostatečně místnou aritmetikou můžeme rozdíly mezi  $\gamma$  a  $\gamma'$  zanedbat a výstup generátoru považovat buď za diskrétní, nebo za spojitou náhodnou veličinu.

b) Algoritmy obecné, jimiž lze vyřešit množství problémů

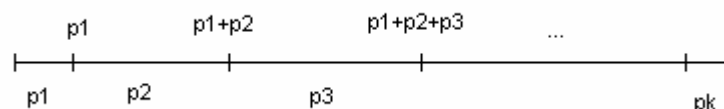
Z obecných algoritmů pro transformaci (rozehrávání) náhodných veličin si uvedeme následující čtyři:

1. Rozehrávání diskrétní náhodné veličiny

V tomto případě budeme mít diskrétní náhodnou veličinu  $\xi$  zadanou následující tabulkou

$$\xi = \begin{pmatrix} x_1 & x_2 & \dots & x_k \\ p_1 & p_2 & \dots & p_k \end{pmatrix}$$

V počítači si vytvoříme vektor o  $k$  složkách, poté nageneryjeme jednu hodnotu  $\gamma$  a určíme do kterého intervalu patří ( obr. 2 ).



**Obrázek 2:** Rozložení pravděpodobností  $p_i$  na intervalu  $\langle 0,1 \rangle$

Zjednodušeně budeme vyšetřovat podmínku

$$\gamma < \sum_{i=1}^j p_i .$$

První interval  $j$ , pro který bude tato podmínka splněna, určí příslušnou hodnotu  $\xi = x_j$ . Tento algoritmus lze použít pro transformaci libovolné náhodné veličiny. V případě, že počet možných hodnot veličiny  $\xi$ ,  $k$ , je příliš velký (trvalo by to dlouho), se používá rychlejší postup, například metoda půlení intervalu.

### 2. Rozehrávání spojité náhodné veličiny - metoda inverzní funkce

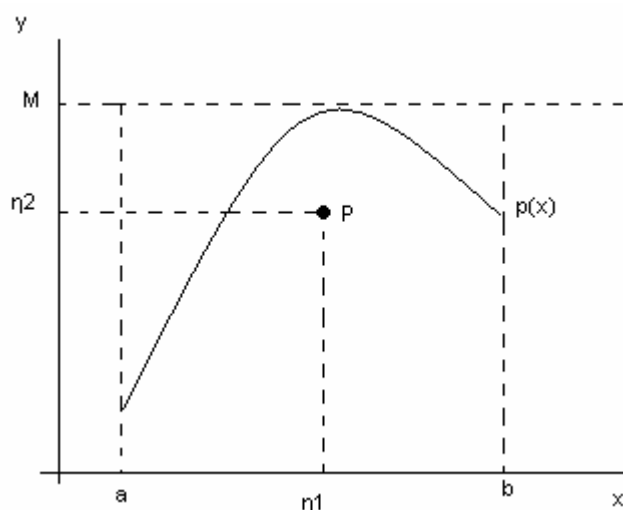
Pro spojité náhodné veličiny máme více algoritmů, nejjednodušší z nich je metoda inverzní funkce. Použijeme obvyklé značení pro hledanou spojitou náhodnou veličinu  $\xi$ , kde  $\langle a, b \rangle$  je obor hodnot a  $p(x)$  je hustota pravděpodobnosti. Pro nalezení spojité náhodné veličiny  $\xi$  na základě veličiny  $\gamma$  musíme vyřešit rovnici

$$\int_a^{\xi} p(x) dx = \gamma . \quad (31)$$

Výsledkem bude transformační vztah typu  $\xi = g(\gamma)$ . Muže se stát, že analytické řešení této rovnice nebude existovat. V takovém případě je inverzní funkce nepoužitelná a my zkusíme další metody.

### 3. Rozehrávání spojité náhodné veličiny- metoda výběru (von Neumanova)

Chceme-li tuto metodu použít, stačí nám jen předpoklad, že hustota pravděpodobnosti  $p(x)$  veličiny  $\xi$  je omezená na intervalu  $\langle a, b \rangle$ .



**Obrázek 3:** Metoda výběru pro generování náhodné veličiny  $\xi$

Postup metody je následující (obr. 3):

- Zvolíme konstantu  $M$ , tak aby platilo  $p(x) \leq M$ ,  $x \in \langle a, b \rangle$ .

- Nagenerujeme dvě hodnoty náhodné veličiny  $\gamma$ ,  $\gamma_1$  a  $\gamma_2$ , a vytvoříme čísla

$$\eta_1 = a + \gamma_1(b - a), \eta_2 = M\gamma_2.$$

- Bude-li bod  $P$  o souřadnicích  $(\eta_1, \eta_2)$  ležet pod křivkou  $y = p(x)$ , tj. bude-li  $\eta_2 < p(\eta_1)$ , zvolíme  $\xi = \eta_1$ . Nebude-li tato podmínka splněna nagenerujeme novou dvojici čísel a postup opakujeme.

Účinnost metody závisí na hodnotě  $M$ , kterou je nutno volit co nejblíže hodnotě

$$M = \sup p(x) \text{ pro } a \leq x \leq b.$$

#### 4. Metoda superpozice

Tuto metodu budeme používat převážně pro spojitou náhodnou veličinu, kde budeme hledat náhodnou veličinu  $\xi$  s distribuční funkcí

$$F(x) = \sum_{i=1}^m c_i \cdot F_i(x).$$

Kde  $F_i(x)$  jsou též distribuční funkce,  $c_1 + \dots + c_m = 1$  a všechna  $c_i > 0$ . Zavedeme diskrétní náhodnou veličinu  $\eta$  s rozdělením

$$\eta = \begin{pmatrix} 1 & 2 & \dots & m \\ c_1 & c_2 & \dots & c_m \end{pmatrix}$$

Hledanou veličinu  $\xi$  s distribuční funkcí  $F(x)$  najdeme tímto způsobem:

- nagenerujeme dvě nezávislé hodnoty  $\gamma_1$  a  $\gamma_2$  veličiny  $\gamma$
- rozehrajeme číslem  $\gamma_1$  hodnotu  $\eta = k$
- z rovnice  $F_k(\xi) = \gamma_2$  určíme veličinu  $\xi$

Za pomoci výše uvedených algoritmů můžeme rozehrávat jakoukoliv náhodnou veličinu, ale pro některé problémy máme specializované algoritmy, některé z nich si uvedeme:

##### 1. Modelování $n$ - rozměrného náhodného bodu

Jelikož souřadnice bodu jsou nezávislé, můžeme každou z nich modelovat zvlášť. Pokud máme modelovat z oblasti složitého tvaru je vhodné použít von Neumanovu metodu:

Zvolíme si jednoduchou oblast, která v sobě obsahuje zadanou oblast. Body v této větší oblasti testujeme, pokud padnou do původní oblasti, tak je ponecháme, jinak vyloučíme.

##### 2. Rozehrání Gaussovské náhodné veličiny $\zeta$

Požadujeme rozehrávat náhodnou veličinu  $\zeta$  s parametry

$$p(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{x^2}{2}\right), \quad x \in (-\infty, \infty).$$

Uvedeme si dva z mnoha existujících algoritmů:

a) Postup je založen na centrální limitní větě počtu pravděpodobnosti. Hodnotu Gaussovské náhodné veličiny  $\zeta$  dostaneme jako součet  $m$  hodnot rovnoměrného rozdělení  $\gamma$

$$\zeta' = \sum_{i=1}^m \gamma_i. \quad (32)$$

Zkonstruovaná veličina bude mít parametr  $\mu = \frac{m}{2}$ ,  $\sigma = \frac{1}{2}\sqrt{\frac{m}{3}}$ . Přejít na požadovanou normalizovanou veličinu  $\zeta$  pomocí vztahu

$$\zeta = \frac{\zeta' - \mu}{\sigma}.$$

Tento algoritmus je relativně rychlý a jeho přesnost je omezená. Velmi vysoká přesnost je dosažena při  $m=12$ .

b) Požadujeme-li zcela přesnou Gaussovskou veličinu, použijeme vztah

$$\zeta = \sqrt{-2 \ln \gamma_1} \cdot \cos(2\pi\gamma_2). \quad (33.1)$$

Kde  $\gamma_1$  a  $\gamma_2$  jsou dvě hodnoty rovnoměrně rozdělené náhodné veličiny  $\gamma$ .

### 3. Modelování gama rozdělení

Požadujeme rozehrát náhodnou veličinu  $\xi^n$  s parametry

$$p(x) = \frac{1}{(n-1)!} \cdot x^{(n-1)} \cdot \exp(-x), \quad x \in (0, \infty), \quad n \geq 1 \text{ celé.}$$

K transformaci použijeme vzorec

$$\xi^{(n)} = -\ln(\gamma_1 \cdot \gamma_2 \cdot \dots \cdot \gamma_n)$$

Pro generování gama  $n$ - tého řádu musíme použít  $n$  hodnot náhodné veličiny  $\gamma$ .

### 4. Rozehrávání náhodného bodu v kouli o poloměru R

Budeme pracovat ve sférických souřadnicích  $(r, \theta, \varphi)$ . Transformační vztahy získané s použitím tří hodnot náhodné veličiny  $\gamma$  jsou

$$\begin{aligned} r &= R\sqrt[3]{\gamma_1} \\ \cos \theta &= 2\gamma_2 - 1 \\ \varphi &= 2\pi\gamma_3. \end{aligned} \quad (33.2)$$

Pokud potřebujeme náhodný bod na povrchu koule, použijeme druhý a třetí vztah. V počítačové fyzice s jejich pomocí rozehráváme náhodný směr, ten získáme pokud náhodným bodem na povrchu koule vedeme polopřímku ze středu koule.

### 4.1.3 Metoda Monte Carlo a matematická statistika

Nejběžnější formulací úloh řešených metodou Monte Carlo je hledání jedné neznámé hodnoty  $a$ . V tomto případě vytvoříme náhodnou veličinu  $\xi$  tak, aby hledané číslo bylo prvním momentem této náhodné veličiny. Očekávanou hodnotou  $E\xi$ , i hledaná  $a$  odhadneme tak, že provedeme  $n$  nezávislých realizací náhodné veličiny  $\xi: \xi_1, \dots, \xi_n$ . Podle zákona velkých čísel (22) platí, že aritmetický průměr těchto realizací je  $\bar{\xi}$ . Předpokládáme-li, že veličina  $\xi$  má konečný rozptyl  $D\xi$ , tak poté dostaneme pro dosti velká  $n$  vztah

$$P\left\{\left|\bar{\xi} - a\right| < x \cdot \sqrt{\frac{D\xi}{n}}\right\} = 2\Phi(x) - 1.$$

Při použití tohoto vzorce pro odhad přesnosti simulace metodou Monte Carlo si nejprve stanovíme dovolené riziko  $q/100$  a pak z tabulky integrálu pravděpodobnosti:

x	0,675	1,212	1,645	1,96	2,58	3,0	3,29
$2\Phi(x) - 1$	0,50	0,80	0,90	0,95	0,99	0,997	0,999

najdeme hodnotu  $x$  vyhovující vztahu

$$2\Phi(x) - 1 = 1 - q/100.$$

Na základě „pravidla tří sigma“ lze zvolit  $1 - q/100 = 0,997$ , čemuž odpovídá  $x = 3,0$ . Tato hodnota určuje horní odhad chyby. V reálných případech se obvykle volí mírnější podmínka- tzv. pravděpodobná chyba  $1 - q/100 = 0,5$ , čemuž odpovídá  $x = 0,67$  a výsledný vztah pro chybu metody Monte Carlo bude mít tvar

$$v = 0,67 \cdot \sqrt{\frac{D\xi}{n}}. \quad (34)$$

Tento vzorec nám pomůže porozumět empiricky nalezenému vztahu (24). Chyba výpočtu metody Monte Carlo závisí nejen na počtu pokusů, ale i charakteristice studované náhodné veličině  $D\xi$ . To nám dává možnost pokusit se zvýšit přesnost výpočtu volbou modelu s menší disperzí  $D\xi$ .

Chybu metody Monte Carlo můžeme kontrolovat již v průběhu simulace a jakmile dosáhneme požadovanou přesnost, je možno simulaci ukončit.

## 4.2 Řešení numerických problémů pomocí metody Monte Carlo

Metoda Monte Carlo byla původně navržena pro řešení problémů z oblasti jaderné fyziky. Posléze se ukázala být velmi vhodná pro řešení úloh ze všech vědních disciplin. Její přirozené použití je tam, kde studovaný problém má statistický charakter, avšak metoda Monte Carlo se dobře uplatňuje v oblastech, kde se pro řešení úloh používají výhradně determinované algoritmy (např. Buffonova jehla).

Algoritmy metody Monte Carlo jsou rozpracovány pro výpočet jednoduchých i násobných integrálů, pro řešení soustav lineárních algebraických rovnic, pro operace s maticemi, atd. Můžeme říci, že ke každému algoritmu z klasické numerické matematiky lze najít ekvivalentní postup v rámci metody Monte Carlo (ne vždy výhodný).

V této části si uvedeme některé důležité příklady, kde je použití metody Monte Carlo proti algoritmům numerické matematiky vhodnější buď z hlediska rychlosti výpočtu, nebo z jednoduchosti algoritmu, případně z obou hledisek.

### 4.2.1 Výpočet určitých integrálů

#### Základní metody

Úkolem je vypočítat integrál z funkce  $f(x)$  na vlastním intervalu  $\langle a, b \rangle$

$$I = \int_a^b f(x) dx. \quad (35)$$

K dispozici máme dvě metody:

#### 1. Výpočet střední hodnoty funkce

Nejprve se pokusíme nalézt střední hodnotu integrované funkce  $f(x)$  na intervalu  $\langle a, b \rangle$ . Podaří-li se nám to, potom určíme hodnotu integrálu

$$I = \int_a^b f(x) dx = (b - a) \cdot \bar{f}.$$

K nalezení střední hodnoty  $\bar{f}$  použijeme metodu Monte Carlo. Vezmeme náhodnou veličinu  $\xi$  rovnoměrně rozdělenou na intervalu  $\langle a, b \rangle$  a zavedeme náhodnou veličinu  $\eta = f(\xi)$ , jejíž matematické očekávání  $E\eta$  je rovno průměrné hodnotě funkce  $f(x)$  na

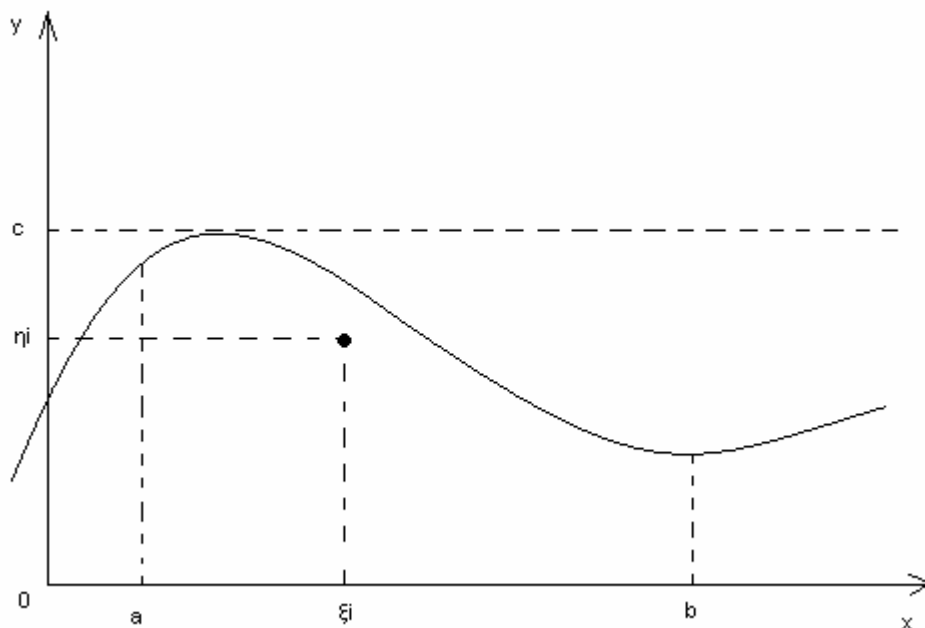
intervalu  $\langle a, b \rangle$ . Hodnotu  $E\eta$  odhadneme pomocí zákona velkých čísel (22) a dostaneme

$$I \doteq (b-a) \cdot \frac{1}{n} \cdot \sum_{i=1}^n f(\xi_i). \quad (36)$$

## 2. Geometrická metoda

Vycházíme ze geometrickému významu integrálu jako plochy pod křivkou  $y = f(x)$  v rozmezích  $a$  až  $b$  (obr.4). Předpokládáme, že funkce  $f(x)$  je na celém intervalu  $\langle a, b \rangle$  omezená. Vezmeme dvě náhodné veličiny:  $\xi \in \langle a, b \rangle$  a  $\eta \in \langle 0, c \rangle$  rovnoměrné rozdělené na příslušných intervalech. Takto připravíme  $n$  náhodných bodů  $(\eta, \xi)$  a pomocí podmínky  $\eta_i < f(\xi_i)$  budeme určovat, kolik z nich padne pod křivku. Jejich počet označíme  $n'$ . Poté vezmeme za integrál  $I$  odhad

$$I \doteq c(b-a) \cdot \frac{n'}{n}. \quad (37)$$



**Obrázek 4.:** Goniometrická metoda pro výpočet určitého integrálu funkce  $f(x)$

## Metody se zvýšenou účinností

Bez znalosti konkrétní funkce není možno rozhodnout, zda je výhodnější použít metodu střední hodnoty, nebo metodu goniometrickou. Avšak porovnání algoritmů založených na metodě Monte Carlo a klasických numerických algoritmů, vyznívá jednoznačně ve prospěch numerické matematiky. Proto je nutné pro praktické použití

obě výše uvedené metody zdokonalit. Jelikož konvergence metody Monte Carlo je velmi pomalá (úměrná  $1/\sqrt{n}$ ), budeme přesnost výpočtu zvyšovat ne zvyšováním počtu pokusů, ale zmenšováním rozptylu  $D\xi$  (34). Uvedme si zde několik příkladů těchto pokusů:

### 1. Nalezení hlavní části:

Rozdělíme integrál na dvě části a to:

- jednu dávající hlavní vklad do výsledku
- druhou jakou upřesňující korekci  $f(x) = q(x) + h(x)$

poté:

$$I = \int_a^b f(x)dx = \int_a^b q(x)dx + \int_a^b h(x)dx.$$

Dokážeme-li první integrál z funkce  $q(x)$  spočítat analyticky, budeme muset určit numericky jen hodnotu integrálu z funkce  $h(x)$ . Pokud však v prvním integrálu je obsažena větší část výsledku, projeví se chyba v přibližném určení hodnoty druhého integrálu na výsledku jen málo. Celkový výsledek poté můžeme získat přesněji, nebo zmenšit počet pokusů  $n$ . Tento postup lze použít jak s metodou střední hodnoty funkce, tak i s geometrickou metodou. Např. pro metodu střední hodnoty se vztah (36) modifikuje na

$$I \doteq \frac{b-a}{n} \sum_{i=1}^n [f(\xi_i) - q(\xi_i)] + \int_a^b q(x)dx.$$

Rozptyl nové náhodné veličiny, se kterou pracujeme,

$$\eta = (b-a)[f(\xi) - q(\xi)] + \int_a^b q(x)dx,$$

bude zmenšený

$$D\eta = (b-a) \int_a^b [f(x) - q(x)]^2 dx - \left[ I - \int_a^b q(x)dx \right]^2$$

a efektivita metody střední hodnoty funkce vzroste.

### 2. Metoda váženého výběru:

Při této modifikaci obou základních metod nebudeme volit náhodná čísla  $\xi$  v intervalu  $\langle a, b \rangle$  rozdělená rovnoměrně, ale v oblasti více přispívající k hodnotě integrálu  $I$  je budeme generovat s větší hustotou. Tento postup existuje ve dvou modifikacích.



V té jednodušší rozdělíme integrační interval  $\langle a, b \rangle$  na několik dílčích intervalů a v každém budeme generovat jiný počet náhodných čísel  $\xi_i = n_1, n_2, \dots, n_{k+1}$ . Zvolíme-li  $n_1 + n_2 + \dots + n_{k+1} = n$  a použijeme-li pro integraci vztah

$$I = \int_a^{c_1} f(x)dx + \int_{c_1}^{c_2} f(x)dx + \dots + \int_{c_k}^b f(x)dx,$$

Dostaneme výsledek přesnější než při práci s  $n$  náhodnými čísly  $\xi$  na celkovém intervalu  $\langle a, b \rangle$  a to tím více, čím více se nebude integrovaná funkce na tomto intervalu měnit.

Při obecnější a přesnější variantě metody váženého výběru budeme náhodnou veličinou  $\xi$  generovat s hustotou pravděpodobnosti spojitě se měnící na celém intervalu  $\langle a, b \rangle$ .

Hledaný integrál upravíme do tvaru

$$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)} p(x)dx.$$

Zavedeme-li novou náhodnou veličinu  $\eta = f(\xi)/p(\xi)$ , bude pro ní platit:  $E\eta = I$ .

Jako odhad hledaného integrálu můžeme vzít vztah

$$I \doteq \frac{1}{n} \sum_{i=1}^n \frac{f(\xi_i)}{p(\xi_i)}.$$

Nyní vhodnou volbou hustoty pravděpodobnosti  $p(x)$  můžeme zmenšit rozptyl a tím i chybu metody. Rozptyl náhodné veličiny  $\eta$  bude

$$D\eta = \int_a^b \frac{f(x)^2}{p(x)} dx - I^2.$$

### 3. Metoda symetrizace integrované funkce:

Obecně platí, že rozptyl metody bude tím menší, čím se bude integrovaná funkce na intervalu  $\langle a, b \rangle$  pomaleji měnit- rozptyl konstanty je nula. Budeme se proto snažit integrovanou funkci upravit tak, aby se funkce na integračním intervalu měnila co nejméně.

Je-li funkce  $f(x)$  např. monotónně rostoucí nebo klesající, je jí vhodné symetrizovat podle vztahu

$$q(x) = \frac{1}{2} [f(x) + f(a+b-x)].$$

Integrál pak budeme moci přibližně spočítat např. metodou střední hodnoty (36) podle modifikovaného vztahu

$$I \doteq \frac{b-a}{2n} \sum_{i=1}^n [f(\xi_i) + f(a+b-\xi_i)],$$

Který bude určitě přesnější než původní vzorec.

### Vícerozměrné integrály

Odlišná situace nastane, pokud změním formulaci problému a místo integrování funkce jedné proměnné (35) budeme potřebovat počítat dvou a vícerozměrné integrály

$$I = \int_{a(c(x))}^{b(d(x))} \int f(x, y) dy dx. \quad (38)$$

Tato situace je ve fyzice běžná. Častěji než jednorozměrné případy řešíme případy dvou- a třírozměrné, někdy i vícerozměrné. Zatímco pro jednorozměrné integrály existuje větší množství metod, přechod k vícerozměrným integrálům vždy znamená značné zkomplikování a prodloužení doby výpočtu. Budeme-li při výpočtu jednorozměrného integrálu brát funkční hodnoty v  $n$  uzlových bodech, při přechodu k  $d$ -rozměrnému problému vzroste počet uzlových bodů na  $n^d$ .

Znamé metody numerické matematiky jsou pro vícerozměrné integrály nepoužitelné, používá se metoda Monte Carlo. Obě základní metody navržené pro jednorozměrnou integraci lze bez problému zobecnit na více rozměrů bez ztráty účinnosti.

Obecná formulace problému (35) bude

$$I = \int_G f(P) p(P) dP, \quad (39)$$

kde  $p(P)$  je zadaná hustota pravděpodobnosti v oblasti  $G$ , přes kterou integrujeme a  $P = (x, y, \dots)$  je bod z oblasti  $G$ .

Metoda střední hodnoty funkce spočívá v tom, že vezmeme náhodné body  $P_1, P_2, \dots$  s hustotou  $p(P)$  a zavedeme náhodnou veličinu  $\eta = f(P)$ , jejíž matematická naděje je rovna  $I$

$$\bar{\eta} = \frac{1}{n} \sum_{i=1}^n \eta_i = \frac{1}{n} \sum_{i=1}^n f(P_i). \quad (40)$$

Analogicky můžeme zobecnit i goniometrickou metodu. A to tak, že si vytvoříme novou  $d+1$  rozměrnou oblast  $G \times (0, c)$  a v ní budeme generovat náhodné body  $Q$ . Budeme vyšetřovat kolik je z těchto  $n$  bodů pod povrchem „plochy“  $z = f(P)$ . Jejich počet je  $n'$ , dostaneme

$$I \doteq cG \frac{n'}{n} \quad (41)$$

Ve více rozměrech lze též používat postupy pro snižování rozptylu. Proti jednorozměrnému případu však přibývá další pravidlo: Dokážeme-li integrovat přes některou proměnou analyticky, rozptyl se sníží.

#### 4.2.2 Řešení Laplaceovy rovnice

Další oblastí matematiky je řešení parciálních diferenciálních rovnic. Z těchto rovnic se ve fyzice vyskytují velmi často Laplaceova a Poissonova rovnice. V této části budeme demonstrovat použití metody Monte Carlo při řešení eliptických parciálních diferenciálních rovnic na řešení Laplaceovy rovnice s Dirichletovou úlohou.

##### Formulace úlohy

Budeme formulovat pro případ, kdy máme stanovit rozdělení elektrostatického potenciálu ve dvou rozměrech:

- vezměme si dvourozměrnou oblast  $G$  s hranicí  $\Gamma$
- máme za úkol nalézt rozložení potenciálu  $U(x, y)$  vyhovující uvnitř oblasti  $G$

Laplaceově rovnici  $\Delta U = 0$  a nabývající na hranici oblasti hodnoty  $U(\Gamma)$

- zavedeme čtvercovou síť s krokem  $h$  očíslovanou indexy  $i$  a  $j$ , místo spojitě rozložených bodů o souřadnicích  $(x, y)$  budeme pracovat pouze s uzly sítě o souřadnicích  $(i, j)$  (tzv. diskretizace prostorových souřadnic).

- místo se spojitým integrálem  $U(x, y)$  budeme pracovat s novou diskrétní funkcí  $u_{ij}$ .

- nejprve nalezneme ekvivalenty parciálních derivací  $\partial U / \partial x$ ,  $\partial U / \partial y$ ,  $\partial^2 U / \partial x^2$ , atd. a z nich sestavíme diferenční ekvivalent celé rovnice. Pro Laplaceovu rovnici dostaneme obvyklý čtyřbodový vztah

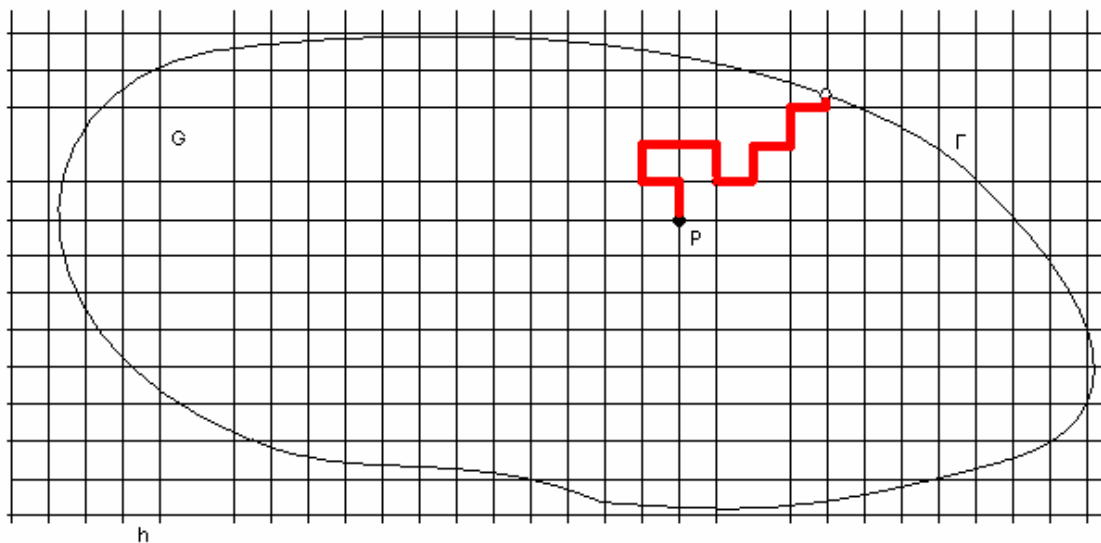
$$u_{ij} = \frac{1}{4} [u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}]. \quad (42)$$

- tento vztah musí platit současně ve všech vnitřních bodech oblasti  $G$  a při řešení se musí využít i hodnoty diskretizované funkce  $u$  na hranicích oblasti  $\Gamma$ , kde se pro uzlové body hranice prostě položí  $U(\Gamma) = u(\Gamma)$ .

Vidíme, že formulace úlohy je shodná s přístupem numerické matematiky. Rozdíl bude v řešení vzniklé soustavy rovnic (42) – místo numerických algoritmů (např. Gaussova eliminace) se využívají stochastické metody Monte Carlo. Uvedenou formulaci problému můžeme modifikovat podle konkrétních podmínek (čtvercovou síť nahradit obdélníkovou, atd.)

### Algoritmus blouzení v pravoúhlé síti

Jednoduchý algoritmus pro řešení Laplaceovy rovnice je založený na představě náhodného blouzení ve vytvořené síti – viz obr.5. Hledáme pravděpodobnost toho, že po vypuštění z bodu  $P$  dorazíme po náhodné trajektorii do některého hraničního bodu, kde již zůstaneme. S každým hraničním bodem je spojena nějaká veličina, kterou při tomto blouzení získáme. Celý pokus pak  $n$ -krát zopakujeme a startovnímu bodu  $P$  přiřadíme průměrnou hodnotu z takto získaných veličin.



**Obrázek 5 :** Síť pro řešení dvourozměrné Laplaceovy rovnice metodou Monte Carlo.

Algoritmus blouzení v pravoúhlé síti

Při náhodném blouzení musíme procházet řadou náhodných bodů a v každém z nich máme stejnou pravděpodobnost, že budeme pokračovat v jednom ze čtyř směrů.

Pokud tuto podmínku vyjádříme v pravděpodobnostech, dostaneme výraz ekvivalentní vztahu (41), jenž odpovídá numerickému řešení právě Laplaceovy rovnice.

Velkou výhodou popsaného algoritmu je, že proti numerickému řešení Laplaceovy rovnice, kdy musí vztah (42) současně platit ve všech vnitřních bodech oblasti  $G$ , zde získáme řešení v jednom bodě  $P$ . Toto řešení je dobré pro nalezení řešení Laplaceovy rovnice v jediném bodě, nebo malém počtu bodů oblasti  $G$ . Potřebujeme-li nalézt řešení Laplaceovy rovnice současně ve všech vnitřních bodech oblasti  $G$ , mnohem efektivnější postup nám nabízí klasická numerická matematika.

Nevýhodou algoritmu bloudění v pravoúhlé síti je právě tato síť. Používáme je ke dvěma účelům, které jsou vůči sobě v protikladu:

- Síť nám slouží k tomu, abychom v jejích uzlech určovali hledané hodnoty potenciálu. Síť by měla být co nejhustší pro dosažení dostatečné přesnosti.

- Síť slouží též k tomu, abychom s její pomocí realizovali co nejnáhodnější průchod. Hustá síť náhodnost nezvýší, naopak prodlouží dobu výpočtu.

Existují algoritmy, které obě funkce sítě od sebe oddělují a celý výpočet tím zefektivňují. Tyto algoritmy (např. algoritmus bloudění s náhodným krokem) jsou o mnohem složitější a jsou doporučovány jen zkušeným uživatelům. Algoritmy bloudění s náhodným krokem jsou vysvětleny v literatuře [4].

### Účinnost metody Monte Carlo při řešení Laplaceovy rovnice

Použití metody Monte Carlo na řešení Laplaceovy rovnice přináší sice jednoduchý algoritmus, ale jeho efektivita není příliš vysoká. Přejít z dvourozměrné do třírozměrné úlohy znamená značné prodloužení výpočtu. Typicky minimální rozměry sítí při praktickém použití numerického řešení Laplaceovy rovnice např. v elektrotechnice jsou řádu  $100 \times 100$  ( $10^4$ ). Přejít na třírozměrnou síť znamená řešit řádově  $10^6$  algebraických rovnic typu (42). V algoritmech metody Monte Carlo stoupá výpočetní náročnost problému s dimenzí pomaleji. Je-li  $\Gamma$  doba potřebná k přechodu z jednoho bodu náhodná trajektorie do dalšího bodu a  $v_i$  je celkový počet kroků jedné trajektorie při  $i$ -tém pokusu, potom celková doba výpočtu

$$T = \tau(v_1 + v_2 + \dots + v_n) = nEv.$$

Počet pokusů  $n$  potřebných k dosažení zadané přesnosti najdeme v zákoně velkých čísel (22). Celková doba výpočtu  $T$  závisí na lineárních rozměrech oblasti  $G$  kvadraticky a při přechodu do třetí dimenze se tento odhad změní jen málo.

Výhodou algoritmů založených na metodě Monte Carlo je, že není nutné počítat současně rozdělení potenciálu ve všech bodech sítě, ale jen ve vybraných bodech. Mezi další výhodou patří malé nároky na paměť počítače.

### 4.2.3 Další problémy

Lze konstatovat, že stochastické algoritmy lze nalézt téměř pro každou oblast numerických výpočtů a otázkou pouze je, kdy je jejich použití výhodné. Metoda Monte Carlo se ve fyzice používá mimo jiné i při interpolaci funkcí mnoha proměnných a při řešení soustav lineárních algebraických rovnic. Při interpolacích funkcí mnoha proměnných se výhodnost metody Monte Carlo projevuje tím více, čím více proměnných interpolovaná funkce obsahuje. Při řešení soustav lineárních algebraických rovnic se používá schopnost soustředit se na řešení jen omezeného problému (hledání jednoho konkrétního kořene). Běžné metody nacházejí současně všechny kořeny, kdežto metoda Monte Carlo dokáže hledat pouze jeden kořen, čímž se doba výpočtu redukuje na veličinu úměrnou  $m$ .

Mezi velmi efektivní aplikace metody Monte Carlo patří *interpolace funkcí mnoha proměnných*. Mějme funkci  $m$  proměnných,  $f(x_1, x_2, \dots, x_m)$ , a mějme zadány její hodnoty pouze ve vrcholech jednotkové  $m$ -rozměrné krychle. Úkolem je určit hodnotu funkce v nějakém bodě o souřadnicích  $(x_1, \dots, x_m)$  uvnitř krychle. Každou souřadnici budeme interpolovat lineárně a poté interpolační vztahy pro jednotlivé rozměry  $m$  budou

$m=1$

$$f(x_1) = (1 - x_1) \cdot f(0) + x_1 \cdot f(1)$$

$m=2$

$$f(x_1, x_2) = (1 - x_1) \cdot (1 - x_2) \cdot f(0) + x_1 \cdot (1 - x_2) \cdot f(1,0) + (1 - x_1) \cdot x_2 \cdot f(0,1) + x_1 \cdot x_2 \cdot f(1,1).$$

S růstem počtu proměnných  $m$  počet členů v interpolačním vztahu prudce narůstá ( $m=30$  dosahuje  $10^9$ ).

O mnoho jednoduší řešení těžké úlohy je pomocí metody Monte Carlo. Vytvoříme  $n$  bodů o náhodných souřadnicích  $(\xi_1, \dots, \xi_m)$  a z nich určíme hodnotu interpolované funkce  $f$

$$f(x_1, \dots, x_m) \doteq \frac{1}{n} \sum_{i=1}^n f(\xi_1^{(i)}, \dots, \xi_m^{(i)})$$

Každá z veličin  $\xi_j^{(i)}$  ( $j = 1, \dots, m; i = 1, \dots, n$ ) může nabývat hodnot 0 nebo 1 s pravděpodobnostmi

$$P\{\xi_j = 0\} = 1 - x_j$$

$$P\{\xi_j = 1\} = x_j,$$

Proto každý bod  $(\xi_1, \dots, \xi_m)$  představuje některý vrchol jednotkové krychle. Hodnoty  $\xi_j$  určíme pomocí náhodné veličiny  $\gamma$  rovnoměrně rozdělené v intervalu  $(0,1)$  podle předpisu

$$\xi_j = 0 \text{ pro } \gamma \geq x_j$$

$$\xi_j = 1 \text{ pro } \gamma < x_j.$$

Dalším problémem je *řešení soustav lineárních algebraických rovnic*, za určitých podmínek můžeme uplatnit metodu Monte Carlo.

Zformulujeme si úlohu: Vezměme si soustavu  $m$  lineárních algebraických rovnic s neznámými  $z_1, \dots, z_m$

$$\sum_{j=1}^m a_{ij} z_j = b_i \quad i = 1, \dots, m. \quad (43)$$

Tuto soustavu lze zapsat ve vektorově formě a to jako

$$A\vec{z} = \vec{b},$$

kde  $A$  je čtvercová matice  $m \times m$ ,  $\vec{z}$  a  $\vec{b}$  jsou vektory o  $m$  složkách. Přepsáním do „iteračního“ tvaru dostaneme

$$\vec{z} = C\vec{z} + \vec{b}.$$

kde matice  $C$  je rovna  $E - A$ ,  $E$  je jednotková matice. Řešení má tvar

$$\vec{z} = C^{-1}\vec{b}. \quad (44)$$

Inverzní matice  $A^{-1}$  můžeme vyjádřit řadou

$$A^{-1} = E + C + C^2 + \dots$$

Řada konverguje tehdy, jsou-li čísla matice  $C$  v absolutní hodnotě menší než 1.

Dosadíme-li rozvoj matice  $A^{-1}$  do hledaného řešení, dostaneme

$$\vec{z} = \vec{b} + C\vec{b} + C^2\vec{b} + \dots$$

Toto řešení můžeme složit z dílčích iterací

$$\begin{aligned}\bar{z}^{(1)} &= \bar{b} \\ \bar{z}^{(2)} &= C\bar{z}^{(1)} + b = C\bar{b} + \bar{b} \\ \bar{z}^{(3)} &= C\bar{z}^{(2)} + b = C^2\bar{b} + \bar{b} \\ &\dots\end{aligned}$$

Metoda jednoduchých iterací konverguje právě tehdy, konverguje-li řada  $\bar{z}^{(1)}, \bar{z}^{(2)}, \bar{z}^{(3)}, \dots$ . Pro  $i$ -tou složku vektoru,  $z_i$ , dostaneme vztah

$$z_i = b_i + \sum_j c_{ij} b_j + \sum_{j_1} \sum_{j_2} c_{ij_1} c_{j_1 j_2} b_{j_2} + \sum_{j_1} \sum_{j_2} \sum_{j_3} c_{ij_1} c_{j_1 j_2} c_{j_2 j_3} b_{j_3} + \dots \quad (45)$$

Řešení téhož problému pomocí metody Monte Carlo. Vztah (45) nám dává návod jak vypočítat jednu složku řešení rovnice (43). Jednotlivé členy výrazu (45) můžeme určit pomocí náhodných veličin. Naším cílem je nalézt  $E\eta_i = z_i$ . Budeme-li postup  $n$ -krát opakovat, dostaneme jednotlivé realizace veličiny  $\eta_i : \eta_i^{(1)} : \eta_i^{(2)} : \dots : \eta_i^{(n)}$ .

Veličinu  $\eta_i$  zkonstruujeme následujícím způsobem: Každý prvek matice  $C$  zapíšeme jako součin dvou prvků

$$c_{ij} = F_{ij} P_{ij}$$

Kde  $P_{ij} \in \langle 0,1 \rangle$ ; pro  $c_{ij} = 0$  položíme  $P_{ij} = 0$ . Analogicky upravíme i prvky vektoru pravé strany  $b$

$$b_i = f_i p_i,$$

$p_i \in (0,1)$ . Volbou prvků  $P$  a  $p$  provedeme tak, aby platilo

$$p_i + \sum_{j=1}^m P_{ij} = 1 \quad i = 1, \dots, m.$$

Na tyto prvky můžeme pohlížet jako na soubor pravděpodobností nezávislých událostí. Rovnici (44) s jejich pomocí přepíšeme do tvaru

$$z_i = f_i p_i + \sum_j F_{ij} f_j P_{ij} p_j + \dots + \sum_{j_1} \dots \sum_{j_r} F_{ij_1} \dots F_{j_{r-1} j_r} f_{j_r} P_{ij_1} \dots P_{j_{r-1} j_r} p_{j_r} + \dots \quad (46)$$

Pro  $i = 1$  nyní rozdělíme interval  $\langle 0,1 \rangle$  na  $m+1$  částí, jejichž délky budou

$$P_{i1}, P_{i2}, \dots, P_{im}, p_i.$$

Totéž můžeme provést i pro ostatní hodnoty  $i = 2, \dots, m$  a získat tak  $m$  způsobů rozdělení intervalu  $\langle 0,1 \rangle$ . Pak nageneryjeme náhodné veličiny nezávisle a rovnoměrně rozdělené na intervalu  $(0,1)$ :  $\gamma_0, \gamma_1, \gamma_2, \dots$ . Nejprve vezmeme  $i$ -té rozdělení intervalu  $\langle 0,1 \rangle$  a zjistíme, kam padne veličina  $\gamma_0$ . Padne-li do poslední,  $m+1$  části délky  $p_i$ ,



položíme realizaci náhodné veličiny  $\eta_i$  rovnou  $f_i$ . Padne-li číslo  $\gamma_0$  do  $j$ -té části rozdělení (o délce  $P_{ij_1}$ ), vezmeme další,  $j_1$ -té rozdělení intervalu  $\langle 0,1 \rangle$

$$P_{j_{11}}, P_{j_{12}}, \dots, P_{j_{1m}}, P_{j_1}$$

a zjistíme, do které části tohoto nového rozdělení padne číslo  $\gamma_1$ .

### 4.3 Použití metody Monte Carlo ve fyzice

Metoda Monte Carlo může najít své místo v matematice a jejím prostřednictvím i ve fyzice. Hlavní fyzikální použití metody Monte Carlo je ovšem při provádění počítačových experimentů. Typické použití metody Monte Carlo je při stochastickém částicovém modelování. Problematice stochastického částicového modelování ve fyzice se budeme věnovat v této části.

Při řešení problémů vyjádřených pomocí postupů numerické matematiky, ať již klasických nebo jejich stochastických ekvivalentů, jsme schopni popsat konkrétní algoritmus řešení studovaného problému, tento algoritmus naprogramovat a pro konkrétní data vyřešit. Při počítačových experimentech ve fyzice stojíme před podstatně složitějším úkolem- fyzikálních problémů je mimořádně veliký počet a teoreticky každý z nich vyžaduje individuální přístup.

Pokud se vrátíme k obecnému schématu řešení problémů pomocí metody Monte Carlo (sekce 3.1):

1. Analýza problému a stochastická formulace modelu pomocí náhodné veličiny  $\xi$
2. Generování náhodné veličiny  $\gamma$  rovnoměrně rozdělené na jednotkovém intervalu
3. Transformace náhodné veličiny  $\gamma$  na hledanou náhodnou veličinu  $\xi$
4. Statistické vyhodnocení opakováním bodů 2 a 3 vzniklého souboru realizací náhodné veličiny  $\xi : \xi_1, \xi_2, \dots, \xi_n$ ,

těžiště počítačového experimentu stochastickým modelováním je v kroku 1.

Při analýze studovaného fyzikálního problému je třeba nalézt vhodný zjednodušený model a určit, která náhodná veličina jej nejlépe popisuje. Do tohoto kroku též patří určení toho, kde najdeme hledaný výsledek. Momenty obvykle vyhodnocujeme tehdy, když hledáme fyzikální řešení ve formě jediné hodnoty, zatímco např. úhlová nebo energetická rozdělení částic nejčastěji převádíme na rozdělení

pravděpodobností náhodných veličin. Tato analýza je velmi důležitá, neboť určujeme techniku statistického vyhodnocení výsledků v bodě 4.

Vlastní tvůrčí činnost leží v bodě 1, který je nejobtížnější. Vše ostatní jsou již standardní postupy, které jdou naučit (sekce 2 a 4.1).

### 4.3.1 Transportní problém

Nejrozšířenějším problémem částicového modelování ve fyzice je transportní problém, při kterém studujeme průchod částic hmotným prostředím. Původní formulace pochází z oblasti jaderné fyziky, kde byl studován transport neutronů v atomovém reaktoru. Lze tak také řešit např.

- transport elektronů v polovodičích
- průchod urychlených elektronů v elektronovém mikroskopu tenkou vrstvou kovu
- pohyb elektronů a iontů v plazmatu (horkém)
- průchod záření látkou, ... .

Algoritmy pro všechny tyto problémy jsou společné a zaleží pouze na fyzikální interpretaci vstupních dat a výsledcích modelování.

### Základní schéma

Pokud bychom řešili transport částic látkou jinými prostředky než metodou Monte Carlo, museli bychom detailně znát celý průběh transportu a ten co nejlépe napodobit výpočetními prostředky – spojitým modelováním nebo deterministickým čističovým modelováním. Metoda Monte Carlo je z tohoto hlediska mnohem bližší reálnému experimentu. Před začátkem modelování je třeba určit podstatné fyzikální procesy, ke kterým při transportu dochází, a jejich rozdělovací zákony. Při vlastní simulaci postupně rozehráváme jednotlivé hodnoty těchto náhodných faktorů a započtením jejich vlivu určíme konkrétní realizaci studovaného náhodného procesu.

Tímto postupem můžeme modelovat historii částic a následným zobecněním velkého počtu těchto historií určit i makroskopické charakteristiky procesu. Uvedené metodě se říká *metoda počítačového experimentu*.

Při počítačovém experimentu můžeme postupovat dvěma cestami. První nejprve analyzuje celý studovaný fyzikální jev a snažíme se vybudovat model co nejvěrnější, ale přitom dostatečně jednoduchý pro následující výpočet. Takovým modelům se říká *přirozené* a budeme se jimi zabývat v této části. Alternativnímu postupu pomocí tzv. *umělých modelů* se budeme věnovat v části 4.3.2.

## Data

Při počítačovém modelování musíme spolu s výběrem používaných algoritmů věnovat pozornost i *datovým strukturám* (tj. kam budou data v průběhu výpočtu ukládána). Toto vyplývá ze zásad strukturovaného programování, které je uvedeno v literatuře[7]. Datová struktura vhodná pro počítačové studium transportu částic je uvedena na obrázku 6.

	1	2	3	4	5	...										
x																
y																
z																
v <sub>x</sub>																
v <sub>y</sub>																
v <sub>z</sub>																
typ																
...																

**Obrázek 6.:** Datové struktury pro čističové modelování – přirozené modely.

Všechny údaje příslušející jedné částici jsou ukládány do jednoho sloupce datové struktury, tj. počet sloupců odpovídá celkovému počtu částic. Ve skutečnosti je situace poněkud složitější. Při částicovém modelování rozehráváme dva typy algoritmů. Jedním z nich je, že při simulaci budeme vypouštět částice postupně a teprve když urazí celou svoji trajektorii, tak budeme sledovat pohyb další částice. Toto je tzv. *jednočásticová metoda*. Jí by odpovídala datová struktura na obr. 6 pouze s jedním sloupcem. Tato metoda se použije tehdy, když je rozhodující pouze interakce částic s prostředím a vzájemnou interakci můžeme zanedbat.

Pokud ale bude vedle interakce s prostředím přítomna i vzájemná interakce částic, musíme současně sledovat chování celého souboru částic. Pak se bude jednat o *metodu mnohočásticovou*.

Nyní si datovou strukturu uvedenou na obr.6 prohlédneme detailně. Pro každou částici do ní budeme ukládat několik skupin údajů:

- Prostorové souřadnice částice

Naše modely mohou být prostorově jednorozměrné nebo vícerozměrné. V našem obr.6 bude první skupina dat obsahovat 1-3 položky, např. kartézské souřadnice  $x, y, z$ . Tato skupina nemůže být prázdná, neboť musí obsahovat minimálně jednu prostorovou souřadnici, která při studiu průchodu částice látkou představuje jakousi řídicí osu (již odpovídá hlavní cyklus v programu).

- Rychlost částic

Abychom mohli sledovat průběh transportu částic je potřeba mít k dispozici rychlost každé částice  $\vec{v}$ . V závislosti na rozměrnosti modelu se může jednat např. o složky rychlosti  $v_x, v_y, v_z$ , případně vyjádření v jiné soustavě, ale také např. o absolutní hodnotu celkové rychlosti  $v$  spolu se složkou  $v_x$ , atd. V některých aplikacích je vhodnější pracovat s kinetickou energií částice ( $E_x, \dots$ )

- Další údaje

Při řešení problému většinou výše uvedené dvě skupiny stačí, pokud ale budeme mít v našem souboru více druhů částic (typické např. pro fyziku plazmatu nebo jadernou fyziku), budou nás zajímat ještě další charakteristiky částice – její hmotnost, náboj, atd. Tím by ale datový soubor narůstal na neúměrnou velikost. Proto je vhodné data optimalizovat, tak že zavedeme jen jeden další integrální parametr „typ částice“ a přiřazením konkrétních hmotnosti atd. se provede v programu jen jednou.

### **Pracovní oblast**

Při částicovém modelování musíme nejdříve určit oblast v níž budeme studovaný proces simulovat. Při modelování transportu metodou Monte Carlo budeme mít tuto oblast tvořenou třemi částmi – viz obr.6:

- zdrojem částic
- vlastní pracovní oblastí v níž se transport odehrává
- cílovou oblastí



**Obrázek 7.:** Znázornění pracovní oblasti při modelování částic

Vlastní program simulující průchod částic hmotným prostředím (jednočásticovou metodou) pak vypadá tak, že ze zdroje vstoupí do pracovní oblasti jedna částice, kterou budeme detailně studovat. Nejprve určíme vzdálenost do bodu interakce s materiálem, pak určíme typ interakce a podle něj upravíme směr a rychlost dalšího pohybu částice nebo částic ( může se stát, že i původně jednočásticová metoda může přejít v metodu mnohočásticovou). Pak celý postup opakujeme a současně testujeme, zda trajektorie nějakým způsobem nebyla ukončena.

#### Zdroj částic

Představuje generátor částic, které vstupují do naší pracovní oblasti. Jeho charakteristiku budeme přebírat ze studovaného fyzikálního jevu (např. bodový zdroj se sférickou symetrií). Z hlediska datové struktury znázorněné na obr. 7 provádí zdroj naplnění příslušných sloupců dat pro částice vstupující ze zdroje počátečními hodnotami, tj. hodnota proměnné  $x$  se nastaví na nulu, zadají se případné hodnoty dalších prostorových souřadnic  $y$  a  $z$ , a zadají se počáteční rychlosti nebo energie těchto částic.

#### Pracovní oblast

Pracovní oblast modelu je ta část prostoru, kde dochází k interakcím procházejících částic s látkovým prostředím. Interakce jsou vždy v modelu přítomné, kdyby nebyly, mohli bychom řešit problém analyticky a nepotřebovali bychom používat techniky počítačového modelování. Charakteristiku prostředí, v němž transport probíhá, musíme převzít z vnějšku modelu, neboť to jsou základní experimentální údaje. Z literatury nebo

od kolegů-experimentátorů musíme získat informaci o počtu rozptylových procesů, jejich intenzitě a přesný popis jejich vlivu na procházející částice. Tyto údaje shrneme do jedné makroskopické veličiny – výsledné střední volné dráhy (nebo celkového účinného průřezu). Na jejím základě vytvoříme tzv. náhodnou volnou dráhu, po jejímž uražení se částice srazí a trajektorie znázorněná obr. 7 se zalomí. Pak musíme z existujících rozptylových procesů vybrat typ konkrétní interakce, podle její charakteristiky změnit rychlost částice  $\vec{v}$ , postup zopakovat a takto vytvořit celou trajektorii. Z hlediska datové struktury se při pohybu v pracovní oblasti budou měnit všechny údaje charakterizující částici. Pokud dojde k tzv. štěpení trajektorie, tj. pokud se při určité srážce vytvoří další částice, zavedeme do datového souboru další částici nebo částice.

### Cíl

Trajektorii můžeme skončit třemi způsoby – standardním vstupem do *cílové oblasti*, ukončení trajektorie záchytem a případně i návratem na částice do zdrojové oblasti. Ve všech případech (třech) zpracování této informace závisí opět na studovaném jevu a na vytvořeném modelu. Obvyklé je, že pouze zapisujeme počet částic, které pronikly do cílové oblasti a srovnáváme je s počtem částic, které pronikly do cílové oblasti a s počtem částic původně zdrojem vypuštěných. Je ovšem možné provádět i detailní analýzu částic vyletujících z pracovní oblasti a zaznamenávat jejich rychlosti  $\vec{v}_i$ , což pak využijeme např. při modelování úhlového nebo energetického rozdělení studovaných částic.

V modelu transportního jevu mluvíme o třech oblastech, které důsledně oddělujeme jak v popisu modelu tak i v programu. Někdy však nelze tyto tři oblasti prostorově odlišit – např. při modelování procesů v nízkoteplotním plazmatu, přesto však při vytváření modelu a jeho realizaci na počítači budeme tyto tři oblasti oddělovat. Z tohoto hlediska se jedná ne o prostorové, ale o časové oddělení – částice jsou nejprve generovány, pak dochází k transportu a nakonec se výsledky zpracují.

### Rozptylové procesy

Předpokládejme, že v látce v níž probíhá transport částic je přítomno celkem  $k$  typů rozptylových procesů. Tyto procesy jsou charakterizovány středními volnými drahami  $\lambda_1, \lambda_2, \dots, \lambda_k$ , případně účinnými průřezy jednotlivých typů interakcí  $S_1, S_2, \dots, S_k$ .

Střední volná dráha  $\lambda_i$  je průměrná vzdálenost mezi srážkami  $i$ -tého typu a udává se v metrech. Účinný průřez můžeme zavést dvěma způsoby, jako makroskopický účinný průřez

$$S_i = \frac{1}{N \cdot \lambda_i},$$

kde  $N = 3,21 \cdot 10^{22}$  molekul nebo atomů v  $m^3$  při  $T = 300K$ . Tento účinný průřez je udáván v  $m^2$ . Druhou možností je mikroskopický účinný průřez

$$S_i = \frac{1}{\lambda_i}$$

udávaný v jednotkách  $m^{-1}$ . Pokud dostaneme experimentální hodnoty, tak rozměr účinného průřezu a jeho číselná velikost nám řekne, která definice byla při měření použita. Z dílčích interakcí vytvoříme celkový účinný průřez  $S$  nebo celkovou střední volnou dráhou  $\lambda$ . Složení dílčích interakcí provedeme podle vztahů

$$S = \sum_{i=1}^k S_i,$$

nebo

$$\frac{1}{\lambda} = \sum_{i=1}^k \frac{1}{\lambda_i}.$$

Dílčí i celkové střední volné dráhy mohou být konstantami většinou však závisejí na poloze nebo na energii částice. Totéž platí pro účinné průřezy.

Základní typy rozptylových procesů jsou následující:

#### 1) Pružný rozptyl

Při pružném rozptylu se při interakci nemění celková energie interagujících částic. Ve fyzice pevných látek je energie částice  $E$  před i po srážce stejná. Směr pohybu částice se ovšem změní, a proto se změní i „složky“ energie  $E_x, E_y$  a  $E_z$ . Naproti tomu např. ve fyzice plazmatu, i při srážce lehkého elektronu s těžkým iontem se při přesném výpočtu musí určitá malá ztráta energie elektronu započítat. Směr částice po interakci je buď zadán simulovanými experimentálními podmínkami, častěji se však předpokládá úhlově izotropní rozptyl, kde se směrové kosiny rozehrávají ze vztahů (33.2).

#### 2) Nepružný rozptyl

Při nepružném rozptylu vždy dochází ke ztrátě energie  $\Delta E$ . Tato ztráta energie může být konstantní (např. při excitaci atomu do vyššího stavu je  $\Delta E$  rovno rozdílu

energetických hladin atomu) , může to být ale i náhodná veličina ( při ionizaci). Směr částic po interakci bývá většinou zadán z experimentu, někdy ale při nedostatku experimentálních dat se opět uchylujeme ke vztahům (33.2).

### 3) Záchyt

Záchyt fyzikálně odpovídá různým mechanismům, např. pohlcení neutronu. V modelu se to ošetřuje stejně – trajektorie prostě v daném bodě končí a ze zdroje je vypuštěna další částice. Model lze doplnit i o popis fyzikálních procesů vyvolaných záchytem částic a uvolněním energie nebo náboje, kterou nese – např. po zabrzdění rychlého elektronu v látce vzniká elektromagnetické záření.

### 4) Štěpení

Štěpení znamená proces, při kterém z jedné primární studované částice vznikají dvě nebo i více sekundárních – neutronů v jaderné fyzice, elektronů v plazmatu, apod. Programovací problém se štěpení spojený je takový, že pokud se často proces opakuje z původní jedné částice vzniká celá lavina. Toto se velmi obtížně programuje.

Po uražení náhodné volné dráhy dojde určitě k interakci, je však třeba rozhodnout, která interakce to bude. Při tom se využijí hodnoty středních volných drah dílčích interakcí  $\lambda_i$ . Pro pravděpodobnost výskytu  $i$ -té interakce použijeme jeden ze vztahů

$$p_i = \frac{\lambda}{\lambda_i}$$
$$p_i = \frac{S_i}{S}$$

Pokud tyto pravděpodobnosti známe, použijeme standardní postup na rozehrávání diskretní náhodné veličiny ( zobrazený na obr.2).

### **Náhodná volná dráha**

Základním úkolem při řešení transportního problému ve fyzice je rozehrávání náhodné volné dráhy. V této souvislosti musíme operovat se dvěma pojmy: *náhodná volná dráha*  $\xi$  je vzdálenost, kterou urazí částice mezi dvěma po sobě jdoucími interakcemi, a jedná se o náhodnou veličinu. Naproti tomu *střední volná dráha*  $\lambda$  je obvyklé (nenáhodné) číslo udávající průměrnou vzdálenost mezi interakcemi a charakterizuje tak prostředí, v



němž transport probíhá. Střední volná dráha představuje první moment náhodné veličiny  $\xi$  a je mezi nimi proto obvyklý vztah (22)

$$\lambda = E\xi = \frac{1}{n} \cdot \sum_{i=1}^n \xi_i .$$

Zákon velkých čísel (22) nám umožňuje přibližně řešit tuto implikaci pro obecnou náhodnou veličinu  $\xi$ :

$$\xi_1, \xi_2, \dots, \xi_n \Rightarrow E\xi ,$$

naproti tomu opačná implikace, kterou potřebujeme vyřešit pro generování náhodných volných drah:

$$\lambda = E\xi \Rightarrow \xi_1, \xi_2, \dots, \xi_n$$

není na obecné úrovni řešitelná.

Pokud využijeme fyzikální vlastnosti rozptylu částic v látce, lze vztah pro generování náhodných volných drah odvodit. Musí však být splněn předpoklad, že střední volná dráha je konstantní. Za tohoto dosti silného předpokladu můžeme použít pro generování jednotlivých realizací náhodných volných drah  $\xi_i$  jednoduchý vztah

$$\xi_i = -\lambda \cdot \ln \gamma_i , \quad (47)$$

kde  $\gamma$  je rovnoměrně rozdělená náhodná veličina v intervalu  $(0,1)$ . Pokud ale není předpoklad  $\lambda = konst.$  splněn, nemůžeme vztah (46) použít.

Vztah pro rozehrávání náhodné volné dráhy  $\xi$  můžeme odvodit tímto postupem: Předpokládejme, že částice se pohybuje z bodu  $x = 0$  podél osy  $x$  a po cestě se může srážet. Zavedeme parametr  $S$  – celkový účinný průřez interakce částice s prostředím. Rozdělovací funkce náhodné volné dráhy  $\xi$ ,  $F(x)$ , bude

$$F(x) = P\{\xi < x\}.$$

Pravděpodobnost, že částice bude mít první srážku v intervalu  $\langle x, x + \Delta x \rangle$ , můžeme určit dvěma způsoby. Za prvé pro  $\Delta x$  dosti malé s využitím distribuční funkce dostaneme

$$F(x + \Delta x) - F(x) .$$

Za druhé můžeme tutéž pravděpodobnost vyjádřit pomocí účinného průřezu  $S$

$$[1 - F(x)] \cdot S \cdot \Delta x ,$$

kde  $1 - F(x)$  je pravděpodobnost, že částice doletí bez srážky do místa  $x$  a  $S \cdot \Delta x$  je pravděpodobnost srážky v intervalu  $\langle x, x + \Delta x \rangle$ .

Vytvoříme-li z obou těchto vztahů pro pravděpodobnost srážky rovnici a povedeme-li limitní přechod  $\Delta x \rightarrow 0$ , dostaneme pro rozdělovací funkci  $F(x)$  výraz

$$F(x) = 1 - \exp\left[-\int_0^x S(s)ds\right].$$

Odpovídající hustota pravděpodobnosti rozdělení náhodných volných drah  $\xi$  bude

$$p(x) = S(x) \cdot \exp\left[-\int_0^x S(s)ds\right]. \quad (48)$$

Náhodná volná dráha je tedy spojitou náhodnou veličinou s oborem hodnot  $x \in \langle 0, \infty \rangle$  a hustotou pravděpodobnosti (48). Při znalosti celkového účinného průřezu  $S(x)$  můžeme tyto údaje využít k rozehrávání konkrétních hodnot náhodných volných drah. Problém je ale ve tvaru závislosti (48) – jedná se o integrál v exponenciále. Pokud bude možno tento vztah spočítat analyticky, lze připravit explicitní vzorec typu (47) a zařadit jej do programu. Jediný problém je, že to lze udělat jen při znalosti konkrétních rozptylových procesů v našem modelu. Pokud ovšem nebude integrál (48) řešitelný analyticky, v modelování metodou Monte Carlo jej nemůžeme použít ( numerické řešení by bylo velmi pomalé).

V prostředí, kde celkový účinný průřez  $S$  bude konstantní, dostaneme zjednodušené výrazy pro rozdělovací funkci a hustotu pravděpodobnosti

$$\begin{aligned} F(x) &= 1 - \exp(-Sx) \\ p(x) &= S \cdot \exp(-Sx). \end{aligned}$$

#### 4.3.2 Metoda nulové srážky

Tato metoda byla navržena v roce 1972 [5] a s její pomocí můžeme i v nehomogenních prostředcích používat pro rozehrávání náhodné volné dráhy jednoduchý vzorec (47)

$$\xi_i = -\lambda \cdot \ln \gamma_i$$

místo podstatně složitějšího a výrazně pomalejšího vztahu (47).

Původně mělo studované prostředí  $k$  typů srážkových procesů charakterizovaných účinnými průřezy  $S_1, \dots, S_k$ . Z nich byl vytvořený celkový účinný průřez  $S$  a celková střední volná dráha  $\lambda$ . Tyto veličiny ale závisely na poloze částic a proto nebylo možno využít pro rozehrávání náhodné volné dráhy vztah (47). Proto v nové formulaci modelu budeme pracovat s prostředím obsahujícím  $k+1$  typů srážkových procesů charakterizovaných účinnými průřezy  $S_1, \dots, S_k, S_{k+1}$ . Srážkový

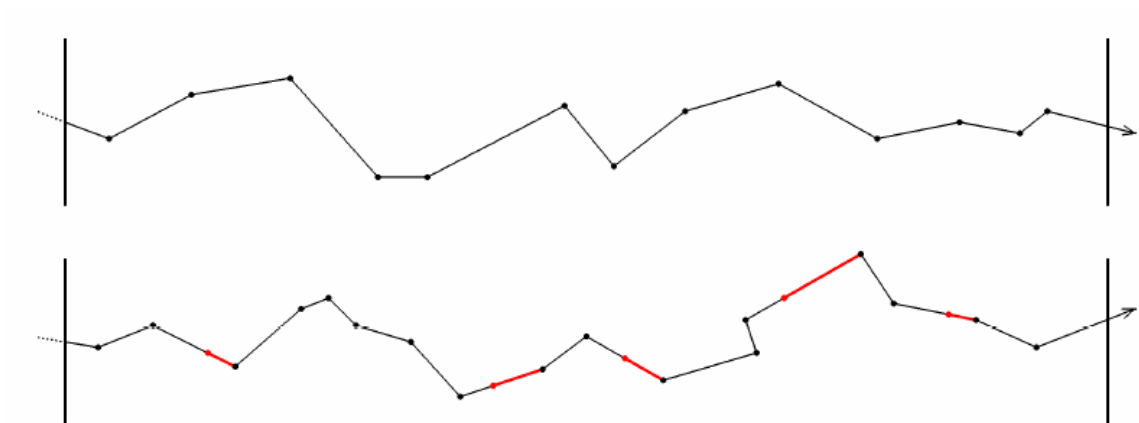
proces jsme si vymysleli, proto si můžeme navrhnout i jeho závislost na poloze. Uděláme to tak, aby nový celkový účinný průřez

$$S' = S_1 + \dots + S_k + S_{k+1}$$

byl konstantní. Pak i upravená střední volná dráha  $\lambda'$  bude konstantní a pro rozehrávání náhodné volné dráhy  $\xi'_i$  můžeme používat jednoduchý vztah

$$\xi'_i = -\lambda' \cdot \ln \gamma_i.$$

Tomuto novému umělému procesu, který jsme do modelu zavedli, se říká *nulová srážka*, protože nebude měnit ani energii interagující částice, ani směr jejího pohybu – další část trajektorie bude prodloužením předchozího úseku. Tento předpoklad plně postačí k tomu, aby kompenzoval vliv zavedeného umělého obratu, což si můžeme demonstrovat na obr.8.



**Obrázek 8.:** Modelování transportního procesu částic pracovní oblastí: nahoře – přirozený algoritmus, dole – simulace s užitím metody nulové srážky.

V případě přirozeného procesu je trajektorie tvořena náhodnými volnými drahami  $\xi_i$  rozehrávanými na základě střední volné dráhy  $\lambda$ . V případě umělého procesu pracujeme s náhodnými volnými drahami  $\xi'_i$  rozehrávanými pomocí střední volné dráhy  $\lambda'$ . Červeně znázorněny na obr.8 jsou náhodně vkládány úseky odpovídající mechanismu nulové srážky, které budou opět náhodně prodlužovat ostatní úseky. Obě trajektorie na obr.8 budou statisticky stejné a z metody nulové srážky se projeví jen výrazná úspora času modelování.

## 5. Rozehrávání veličin

V této sekci si uvedeme praktické rozehrávání rychlostí s maxwellovským a nemaxellovským rozdělením rychlostí. Zde využijeme poznatky, které jsme si již v dřívějších sekcích připomněli.

### 5.1 Rozehrávání rychlostí s maxwellovským rozdělením rychlostí

V této kapitole se budu zabývat řešením problému rozehrávání rychlostí částic ideálního plynu. Z teorie tekutin vyplývá, že v ustáleném stavu, kdy na částice nepůsobí žádná vnější síla, mají částice maxwellovské rozdělení rychlostí. Rozdělovací funkce má tedy v případech, kdy můžeme uvažovat tři stupně volnosti každé částice tvar

$$f(\varepsilon_e) = \frac{2}{\sqrt{\pi}} \frac{\sqrt{\varepsilon_e}}{k_B T_e^{1.5}} \cdot \exp\left(-\frac{\varepsilon_e}{k_B T_e}\right)$$

kde

Rozdělovací funkce vyjádřená jako funkce rychlosti se dá psát ve tvaru

$$f(v) = A \cdot v^2 \exp\left(-\frac{m_e v^2}{2k_B T_e}\right)$$

kde konstantu A lze pro třírozměrný pohyb elektronů zapsat ve tvaru

$$A = \sqrt{\frac{2}{\pi}} \left(\frac{m_e}{k_B T_e}\right)^{\frac{3}{2}} \quad (49)$$

Jednou z možností, jak nagenarovat soubor rychlostí částic, je využití metody inverzní funkce, jak byla popsána v jedné z předchozích kapitol. Za tímto účelem byl v programovém balíku MATLAB vytvořen tento zdrojový kód, který jsme si barevně rozlišili, pro následné vysvětlení:

```
m_e=9.109534e-31;
T_e=23200;
k_B=1.380662e-23;
q=1.6021892e-19;
pocet=1000000;

konstanta=m_e/(k_B*T_e);
vMax=5*sqrt(2*k_B*T_e/m_e);
v_Maxwell=linspace(0+eps, vMax, 1000);

f= @(v)(2/sqrt(2*pi))*(konstanta.^1.5)*(v.^2).*exp(-
konstanta.*(v.^2)/2);
```

```

for i=1:1:1000
    F(i) = quad(f,0,v_Maxwell(i));
end

v=interp1(F, v_Maxwell, rand(1,pocet), 'spline');
Theta=acos(2*rand(1,pocet)-1);
fi=2*pi*rand(1,pocet);
v_x=v.*cos(fi).*sin(Theta);
v_y=v.*sin(fi).*sin(Theta);
v_z=v.*cos(Theta);

```

V následujících několika řádcích si projdeme vytvořený program a vysvětlíme si postupně co který řádek znamená:

```

-   konstanta=m_e/(k_B*T_e);
    vMax=5*sqrt(2*k_B*T_e/m_e);
    v_Maxwell=linspace(0+eps, vMax, 1000)

```

Tato část slouží pro nadefinování konstant, ve které jsou napočítány hodnoty pomocných proměnných, které umožňují efektivnější provedení výpočtu.

```

-   f= @(v) (2/sqrt(2*pi))*(konstanta.^1.5)*(v.^2).*exp(-
    konstanta.*(v.^2)/2);

```

Zde je definována MATLABovská funkce *f*, která odpovídá rozdělovací funkci rychlostí elektronů (viz obrázek 9).

```

-   for i=1:1:1000
        F(i) = quad(f,0,v_Maxwell(i));
    end

```

Zde jsou numerickou integrací za využití MATLABovské funkce *quad* napočítány hodnoty distribuční funkce *F* – viz obrázek 10.

```

-   v=interp1(F, v_Maxwell, rand(1,pocet), 'spline');

```

Po ukončení cyklu jsou interpolací (MATLABovská funkce *interp1*) a za využití hodnot funkce *rand*, která generuje náhodná čísla s rovnoměrným rozdělením na intervalu  $\langle 0,1 \rangle$ , nalezeny hodnoty rychlostí, které odpovídají nagenované hodnotě distribuční funkce *F*. Je patrné, že při využití MATLABovské „maticové syntaxe“ odpadá psaní komplikovaných cyklů, jako by tomu muselo být například v programovacím jazyku C či Fortran, a celý příkaz lze přehledně zadat na jediném řádku.

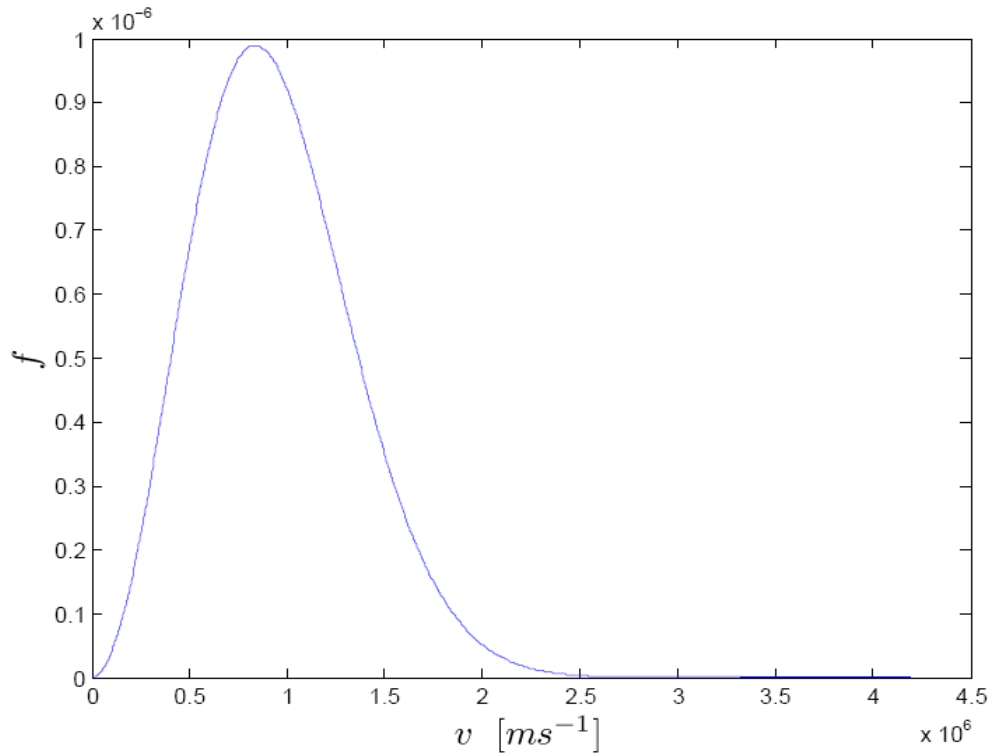
```

-   Theta=acos(2*rand(1,pocet)-1);
    fi=2*pi*rand(1,pocet);
    v_x=v.*cos(fi).*sin(Theta);
    v_y=v.*sin(fi).*sin(Theta);

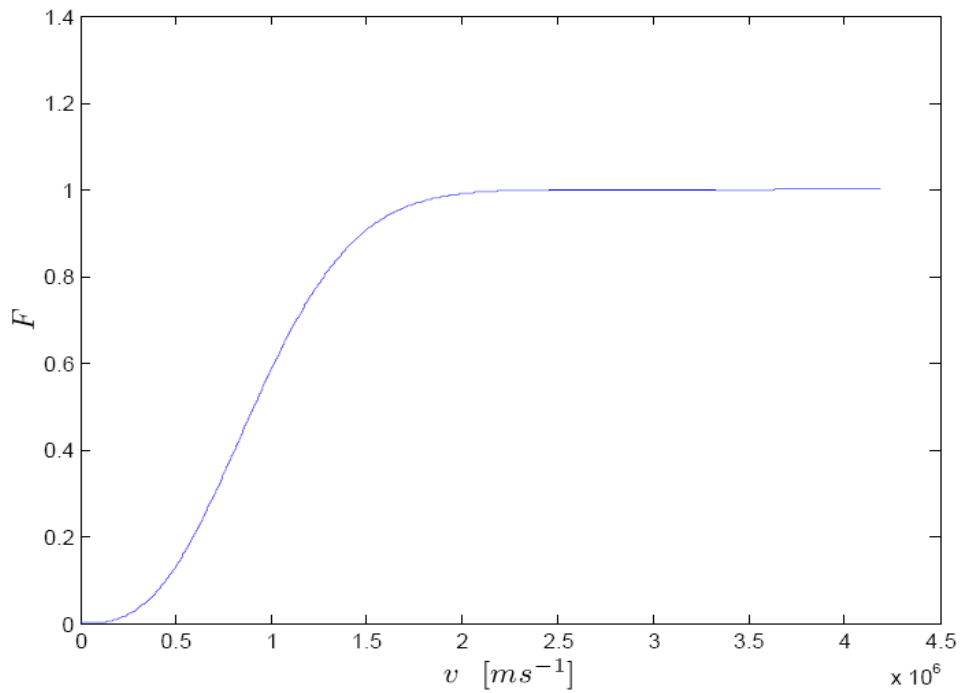
```

```
v_z=v.*cos(Theta);
```

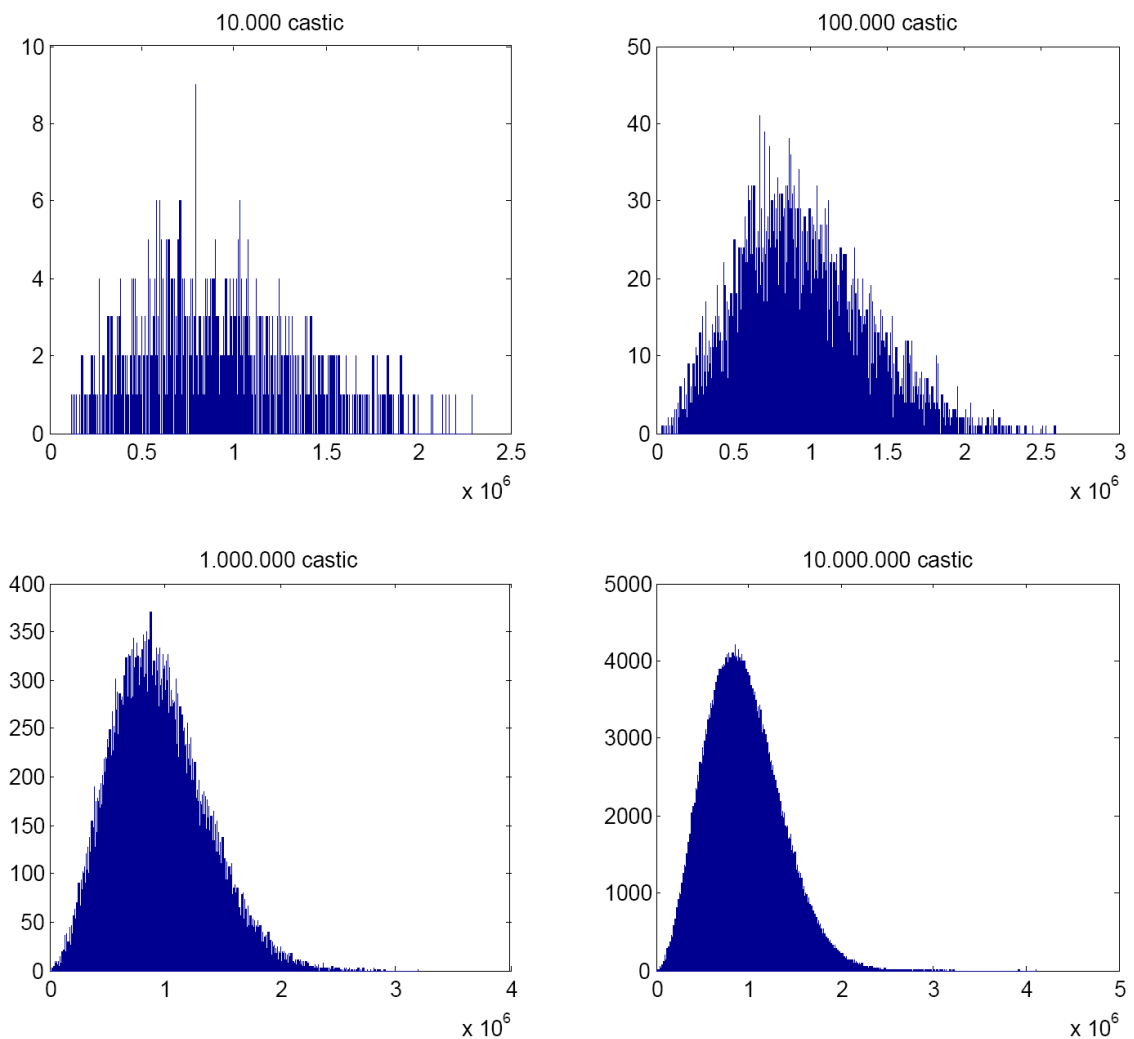
Tyto poslední řádky zápisu umožňují rozehrání jednotlivých složek rychlostí ve třech nezávislých směrech. Rozehrávání odpovídá generování bodů v jednotkové kouli



**Obrázek 9:** Hustota pravděpodobnosti  $f$  generovaná zdrojovým kódem.



**Obrázek 10:** Distribuční funkce  $F$  generovaná zdrojovým kódem.



**Obrázek 11:** Histogramy rychlostí částic – histogramy byly sestrojeny ze statistického souboru hodnot tvořeného 10.000, 100.000, milionem a 10 miliony částic.

Zdrojový kód jsem poté využil pro generování rychlostí elektronů o teplotě  $T = 23.200\text{ K}$  s maxwellovským rozdělením rychlostí. Tato úloha odpovídá například generování parametrů částic ve zdroji. Na obr. 11 je patrné, že při větším počtu generovaných částic jsou generovány i rychlosti částic z energetičtější části spektra a rozdělovací funkce vykazuje mnohem menší fluktuace.

Z pohledu časové náročnosti výpočtu se dá říci, že ho lze provést v závislosti na počtu generovaných hodnot za řádově jednotky sekund.

## 5.2 Rozehrávání rychlostí s nemaxwellovským rozdělením rychlostí

V této kapitole popíšeme možnost, jak lze nagenarovat rychlosti částic s obecně definovanou rozdělovací funkcí  $f$ . Například při studiu ionizovaných plynů, nebývá předpoklad maxwellovského rozdělení rychlostí často splněn, neboť rozdělovací funkce bývá často deformována například přítomností elektrického pole nebo neelastickými srážkami mezi částicemi (excitace, ionizace). Přesto může být úloha rozehrát rychlosti částic důležitá pro další provedení výpočtu, jako je tomu například ve třetí části algoritmu prezentovaném v [6].

V tomto případě je rozdělovací funkce definována v jednotlivých bodech (je zadána tabulkou). Pro ukázkou rozehrajme náhodnou funkci  $\sin(x)/2$  na intervalu  $\langle 0, \pi \rangle$ . Předpokládejme, že ji máme definovanu ve 100 ekvidistantně rozložených uzlových bodech. Zdrojový kód pak může vypadat například takto (opět barevně rozlišíme):

```
xMin=0;
xMax=pi;
yMin=0;
yMax=1;

x=linspace(xMin, xMax, 100);
ROZDELENI_v=sin(x)/2;

pocet=100000;

xRand = (xMax - xMin)*rand(1, pocet);
yRand = (yMax - yMin)*rand(1, pocet);

pom=interp1(x, ROZDELENI_v, xRand, 'spline');
index = find( yRand < pom);

X=xRand(index);

hist(X, 100)
```

V následujících několika řádcích si projdeme vytvořený program a vysvětlíme si postupně co který řádek znamená:

```
- xMin=0;
  xMax=pi;
  yMin=0;
  yMax=1;
```

Tyto čtyři řádky definují minimální a maximální hodnoty, které budou generovány na ose x a y.

```
- x=linspace(xMin, xMax, 100);
  ROZDELENI_v=sin(x)/2;
  pocet=100000;
```



```
xRand = (xMax - xMin)*rand(1, pocet);  
yRand = (yMax - yMin)*rand(1, pocet);
```

První dva řádky (z těchto čtyř), v nichž nejprve generujeme uzlové body a posléze určíme v nich tabelované hodnoty. Pokud není daná funkce dána předpisem, můžeme data importovat například pomocí tabulky. Definiční řádek `pocet=10000` určuje počet bodů, které budeme generovat v obdélníku. Matice `xRand` a `yRand` pak obsahují nagenеровané souřadnice.

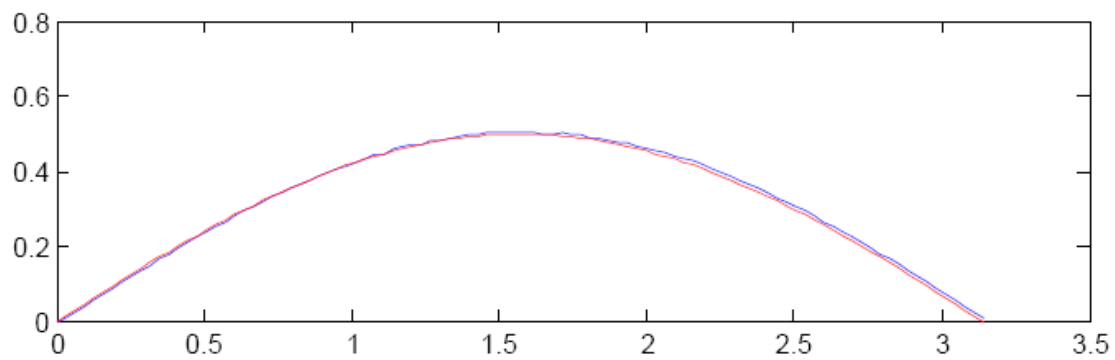
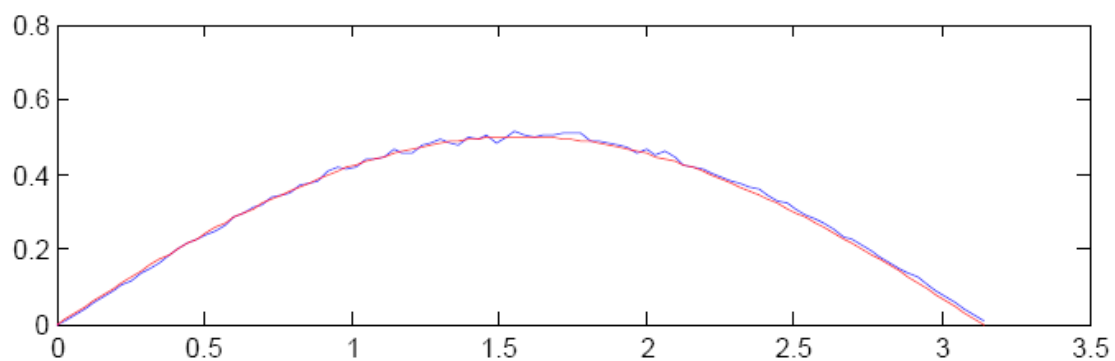
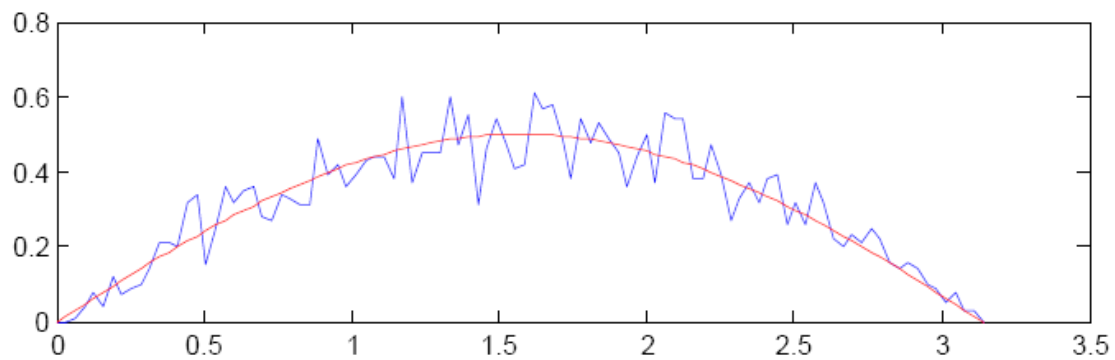
```
- pom=interp1(x, ROZDELENI_v, xRand, 'spline');  
  index = find( yRand < pom);
```

V těchto dvou řádcích si pak pomocí interpolace kubickým splajnem určíme matici `pom`, v níž jsou určeny aproximativní hodnoty tabelované funkce v náhodných bodech `xRand`. Indexy prvků, které leží v červené oblasti jsou obsaženy v matici `index`.

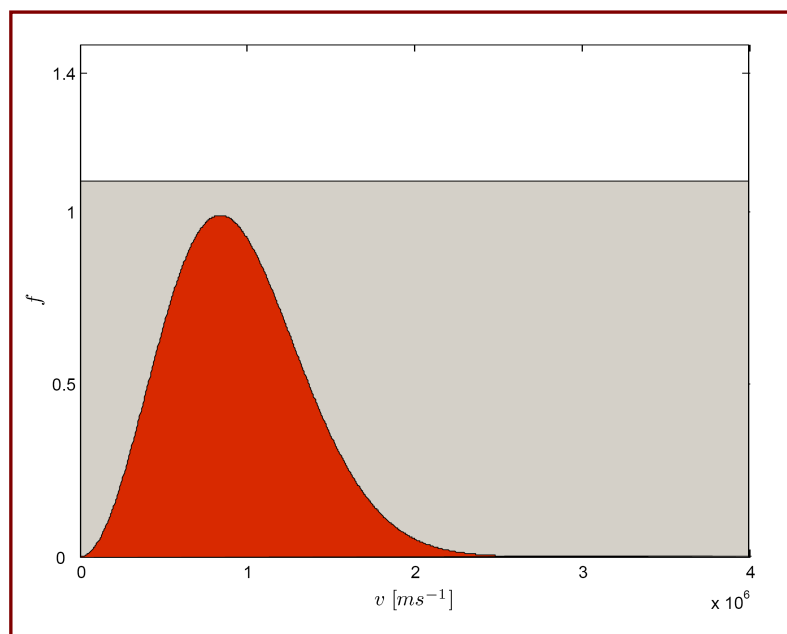
```
- X=xRand(index);
```

```
  hist(X, 100)
```

Zde se již pouze předávají hodnoty `xRand`, které mají rozdělovací funkci  $\sin(x)/2$ , jak je patrné z histogramů na obrázku 12. Opět platí, že se zvyšujícím se počtem nagenеровaných hodnot se zmenšuje počet fluktuací v těchto histogramech.



**Obrázek 12:** Rozehrávání neznámé funkce: horní graf 10.000 hodnot, prostřední 1.000.000 a dolní 10 milionů hodnot.



**Obrázek 13:** K vysvětlení principu metody.

Princip metody je takový (viz obrázek 13), že v obdélníku daných rozměrů generujeme bod, který je definován jako uspořádaná dvojice hodnot  $[x, y]$ . Hodnoty  $x$  generuje v rozsahu definičního oboru na ose  $x$ , hodnoty  $y$  v rozsahu od 0 do maxima tabelované funkce  $f$ . Nagenerovaný bod pak leží v šedém obdélníku. Pokud navíc leží i pod křivkou, tzn. v červené oblasti, bereme jeho  $x$ -ovou souřadnici jako hodnotu generované funkce. Hodnoty s větší pravděpodobností jsou tímto způsobem generovány tak často, jak to odpovídá dané pravděpodobnosti. Na obrázku 12 je histogram s rozdělovací funkcí  $\sin(x)/2$ .

## **6. Závěr**

Při sestavování mé bakalářské práce jsem čerpal z mnoha dostupných materiálů, které mám uvedeny v použité literatuře. Čerpání z těchto materiálů bylo velmi obtížné z důvodu cizího jazyka, ve kterém tyto materiály byly napsány, tj. většinou v angličtině. Další materiály, které jsem použil, jsem našel na internetových stránkách, v nápovědě programu MATLAB, nebo ve skriptech pro výuku tohoto programu od pana doc. RNDr. Josefa Blažka, CSc.[9].

Práce s programem MATLAB je velmi zajímavá a v některých částech podobná programování. S programem jsem se učil pracovat od nejjednodušších příkazů, kterými jsem např. vypočítal jednoduchý vzorec, nebo sestrojil graf až po příkazy, které jsem použil ve své práci. Se zvládnutím programu MATLAB mi pomáhal můj konzultant pan Mgr. Petr Bartoš Ph.D.

Tato bakalářská práce mi ukázala, jak těžké a hlavně časově náročné je vytvořit „vlastní publikaci“. A to i přesto, že jsem čerpal z materiálů již dříve sepsaných a nemusel jsem objevovat a vymýšlet žádné zákony. Když pomínu časovou náročnost, byla práce obtížná především v tom, že když jsem část práce udělal, a poté sem se k ní vrátil, nezdála se mi formulace textu příliš vhodná a mnohokrát jsem předělával to, co jsem již dříve napsal.

Mnou vybraná práce mi ukázala, jaké úsilí vyžaduje sepsání odborného materiálu, ale přinesla mi spoustu pozitivních poznatků jako např. práci s programem MATLAB.

## **Anotace:**

Tato práce obsahuje celkem 62 stran, na kterých jsou vysvětleny základní pojmy z teorie pravděpodobnosti a matematické statistiky. Po této kapitole následuje vysvětlení počítačového modelování a modelování Monte Carlo, ve kterém si ukážeme základní techniky a řešení problému pomocí této metody. Na závěr práce nalezneme ukázkou s maxwellovským a nemaxwellovským rozdělením rychlostí, které je namodelováno v programu MATLAB.

## **Abstract:**

This project contains total of 62 pages, in which are explained basic terms of probability theory and mathematical statistic. This part is followed by explanation of computer modeling and Monte Carlo modeling, in which are shown basic techniques and solutions of the problem by this method. In conclusion of this project is shown an example on maxwell's and nonmaxwell's allocation of speeds, which has been programmed in program MATLAB.

## **7. Literatura**

- [1] Metropolis N., Ulam S.: *The Monte Carlo Method*, J. Amer. Statist. Assoc. **44** (1949), No 247, 335.
- [2] Buffon G.L.L.: *Essai d'arithmétique morale*, Suppl. , à l'Histoire Naturelle, Vol. 4, Imp. Royale, Paris 1777.
- [3] Ripley B.D.: *Stochastic Simulation*, John Wiley and Sons, New York 1987.
- [4] Hrach, R.: *Počítačová fyzika I*, PF UJEP Ústí nad Labem 2003
- [5] Skullerud H.R.: *Null collision method*, Techn. Report EIP 72-1, Norwegian Inst. of Technology, Trondheim 1972.
- [6] Bartoš, P.: *Hybridní modelování ve fyzice plazmatu*, MFF UK PRAZE, PRAHA, 2007
- [7] Wirth N.: *Algorithms + Data Structures = Programs*.  
Přepřacované vydání - Wirth N.: *Algorithms and Data Structures*, Prentice Hall, USA 1986.
- [8][http://www.eamos.cz/amos/kat\\_fyz/modules/external/index.php?kod\\_kurzu=kat\\_fyz\\_1272](http://www.eamos.cz/amos/kat_fyz/modules/external/index.php?kod_kurzu=kat_fyz_1272)