

Seriál: Numerické metody a počítačové simulace

V minulém díle jsme se seznámili s numerickou derivací a se základy stochastické metody Monte Carlo, jmenovitě MC integrací. Nyní navážeme obecnějším výkladem numerické integrace, ale předtím se ještě podíváme na matematický pojem „náhodná procházka“. Náhodné procházky využijeme jednak v pokročilých metodách MC integrace, ale také například při simulaci Brownova pohybu.

Náhodné procházky

Na počátku studia fyziky se studenti obvykle seznámí s mechanikou a Newtonovými zákony. Pomocí nich se naučí spočítat, kterým směrem a jakou rychlostí se bude pohybovat hmotný bod v důsledku sil na něj působících. Tento přístup ovšem selhává, pokud budeme chtít předpovědět například trajektorii prachového zrnka ve vzduchu. Abychom našli polohu částice o byt jen několik nanosekund později, potřebovali bychom vyřešit statistice srážek – musíme totiž znát také aktuální polohu všech molekul, které by se s naší částicí mohly srazit. Tento výpočetně náročný přístup využívá v simulacích molekulová dynamika.

Snížíme-li své nároky a budeme chtít nalézt pouze střední vzdálenost nebo jinou statistickou veličinu, odemyká se nám možnost použít stochastické simulace. Základní myšlenkou je zde předpoklad, že okolní molekuly se pohybují náhodnými směry a rychlostmi (v případě klasického plynu určenými gaussovským rozdělením). Toto molekulární pozadí vyjmeme ze svých výpočtů a budeme se věnovat pouze pohybu těžké částice, který bude v důsledku náhodného hemžení na pozadí také náhodný. Pro pohyb těžké částice v důsledku chaotického pohybu molekul okolní tekutiny se ustálil název *Brownův pohyb*. Tento fyzikální pohyb můžeme aproximovat diskretním matematickým modelem zvaným *náhodná procházka*, který v každém časovém kroku posune částici náhodným směrem o pevně zvolenou vzdálenost. Náhodná procházka spadá do širší rodiny Markovových procesů, na které se nyní podíváme podrobněji.

Markovovy procesy

V prvním dílu seriálu jsme se seznámili s pojmem náhodná proměnná. Jednalo se o proměnnou, jejíž hodnota není předem známa (tj. není to konkrétní číslo), pouze známe pravděpodobnosti nabývání hodnot z určité množiny. Tyto pravděpodobnosti byly dány pravděpodobnostním rozdělením, např. náhodná proměnná „hod kostkou“ mohla nabývat hodnot 1 až 6 z diskretního rovnoměrného rozdělení. Pokud uspořádáme náhodné veličiny do řetězce, tj. přiřadíme náhodným veličinám indexy $1, 2, \dots, n$, nazýváme tento řetězec *náhodný proces*. Příkladem náhodného procesu je třeba série hodů mincí.¹ Realizací tohoto náhodného procesu je uspořádaná množina, jejíž každý prvek nabývá hodnoty „panna“ nebo „orel“ – například jsme vytvořili pomocí pěti hodů korunovou mincí realizaci náhodného procesu

{panna, panna, panna, orel, panna}.

¹Konkrétně se jedná o tzv. Bernoulliho proces s pravděpodobnostmi $p = 1/2$ a $1 - p = 1/2$.

V uvedeném procesu jsou prvky řetězce představovány navzájem nezávislými a shodně rozdělenými² náhodnými veličinami. Obecně však mohou být veličiny závislé a různě rozdělené. Uvažte třeba takovýto proces (A): Házím mincí, dokud nepadne panna, a pak házím kostkou, dokud nepadne šestka; pak házím mincí, dokud nepadne panna, a pak házím kostkou, dokud . . . Jiný příklad (B): Házím šestistěnnou kostkou a počítám oka tak dlouho, dokud nezískám součet větší nebo roven 20; poté vyměním šestistěnnou kostku za desetistěnnou a pokračuji v přičítání, dokud nedosáhnu součtu 40; poté se vrátím k šestistěnné kostce atd.

Nyní uvažujme speciální případ náhodného procesu, kdy jsme schopni určit stav v následujícím kroku bez znalosti historie procesu. Jinými slovy, pravděpodobnost realizace hodnoty x náhodné veličiny X_{i+1} dokážu jednoznačně určit na základě realizace y náhodné veličiny X_i . Této vlastnosti říkáme přiléhavě *bezpaměťovost* a příslušný náhodný proces se nazývá *Markovův*.³ Formálně tuto vlastnost (pro diskrétní náhodné veličiny) zapíšeme

$$P(X_{i+1} = x_{i+1} | X_1 = x_1, X_2 = x_2, \dots, X_i = x_i) = P(X_{i+1} = x_{i+1} | X_i = x_i) \quad (1)$$

a čteme: Pravděpodobnost, že náhodná veličina X_{i+1} nabývá hodnoty x_{i+1} za podmínky, že v prvním kroku nabyla veličina X_1 hodnoty x_1 , v druhém kroku nabyla veličina X_2 hodnoty x_2 atd., je stejná jako pravděpodobnost, že X_{i+1} nabude hodnoty x_{i+1} za podmínky, že v předešlém kroku nabyla X_i hodnoty x_i . Znalost řetězce na rozsahu indexů 1 až $i-1$ je tedy úplně zbytečná, můžeme ji zapomenout.

Série hodů mincí je triviálním případem bezpaměťového řetězce, kdy není potřeba znát ani stav v kroku i (toto platí pro každý řetězec navzájem nezávislých náhodných veličin). Proces A je také Markovův proces, další krok se řídí vždy současným stavem. Bepaměťový je i proces B – pro určení stavu v kroku B nám postačí znalost aktuálního součtu a typu kostky, kterou jsme naposled házeli. Vymyslet proces, který není Markovův, je snadné, například: Házím kostkou a počítám hodnoty, pokud padnou dvě šestky za sebou, přičtu k součtu v následujícím kroku bonus 10. Pro určení stavu v následujícím kroku je zjevně potřeba znát alespoň dva předchozí stavy, předpoklad (1) je porušen.

Náhodná procházka v 1D

Náhodná procházka je náhodný proces daný součtem nezávislých a shodně rozdělených náhodných veličin. Nejjednodušším příkladem je opět házení mincí, na rozdíl od příkladu výše však není aktuální stav určen posledním hodem, ale součtem všech dosavadních hodů (přitom uvažujeme panna = -1 , orel = 1). Název náhodná procházka vychází z představy, kdy součtu realizací náhodných veličin přiřadíme význam vzdálenosti od počátku – máme tedy objekt, který se v každém kroku pohne náhodně doleva, nebo doprava, a takto se prochází po přímce.

Možná si vzpomenete na úlohu z prvního dílu seriálu, kdy Mirek s Lukášem hráli o FYKOSí trička, přičemž o vítězi trička se v každém kroku rozhodovalo na základě hodu mincí. Tehdy jsme jako „hru“ označili sekvenci obsahující počty triček jednoho z hráčů. „Hra“ bylo tedy označení pro realizaci náhodné procházky končící pro některého hráče nulou (stav, kdy jeden hráč měl všechna trička nebo žádné, představoval *absorpční bariéru* = konec hry).

Nadále budeme často zaměňovat pojmy náhodná procházka a realizace náhodné procházky, je však dobré mít na paměti, že se principiálně liší.

²V anglické literatuře se můžete setkat se zkratkou *iid* – independent and identically distributed.

³Občas se v učebnicích můžete setkat i s označením markovský či markovovský proces, v obou případech jde o podobnostní přídavné jméno odvozené od ruského matematika Andreje Andrejeviče Markova.

Oblíbenou učebnicovou úlohou je „opilcova procházka“: Opilec se o půlnoci vrací domů z hospody. Dům, kde na něj čeká rozčilená manželka, se nachází ve vzdálenosti a od hospody. Druhým směrem se ve vzdálenosti b od hospody nachází řeka. Opilec se s každým krokem posune o vzdálenost $+1$ (k řece), nebo -1 (k domovu), přičemž oba směry jsou stejně pravděpodobné. Zajímat nás budou následující otázky:

- Jaká je pravděpodobnost, že opilec dojde dřív domů než do řeky?
- Jaká je pravděpodobnost, že se opilec po n krocích nachází v bodě k ?
- Jaká je střední poloha opilce?
- Jaká je střední vzdálenost opilce od hospody?

Než začneme úlohu řešit, zdefinujeme si procházku formálněji. Každý krok je náhodná veličina S_i , která nabývá hodnoty $+1$ s pravděpodobností $0,5$ a hodnoty -1 s pravděpodobností také $0,5$. Součet n těchto náhodných veličin označíme

$$W_n = \sum_{i=1}^n S_i.$$

Posloupnost $\{W_n\}$ potom představuje náhodnou procházku, přesněji *jednoduchou* náhodnou procházku. Přitom bez újmy na obecnosti volíme počáteční polohu $W_0 = 0$.

Při studování náhodné procházky z pohledu Markovových řetězců nás zajímá pravděpodobnost přechodu mezi stavy. Prostorem stavů je v případě jednorozměrné náhodné procházky množina celých čísel \mathbb{Z} . $P_{i,i+1}$ značí pravděpodobnost přechodu ze stavu i do stavu $i+1$, tj. pravděpodobnost, že uděláme krok vpravo. Pro procházku, kde je krok na obě strany stejně pravděpodobný, píšeme $P_{i,i+1} = 1/2 = P_{i,i-1}$. Toto lze vyjádřit také pomocí matice přechodu

$$T = \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 \end{pmatrix},$$

kteřou jsme zde omezili na množinu $\{1, 2, 3, 4\}$, ale v obecnosti je nekonečná.

První otázku jsme již zodpověděli v prvním dílu seriálu, když jsme se ptali, s jakou pravděpodobností vyhraje Mírek nad Lukášem ve hře o trička. Pravděpodobnost, že opilec dojde dřív domů, je $b/(a+b)$, a podobně pravděpodobnost, že dřív spadne do řeky, je $a/(a+b)$. Důkaz naleznete ve vzorovém řešení seriálové úlohy.

Abychom vyřešili druhou otázku, potřebujeme si rozmyslet, kolika různými způsoby můžeme po n krocích dojít do bodu k . V prvním kroku můžeme jít nalevo nebo napravo a skončit v bodě -1 nebo 1 . V druhém kroku již máme více možností: budto se z bodů 1 , -1 vrátíme zpět do nuly (tj. po dvou krocích jsme se dokážeme dostat nuly dvěma různými způsoby), nebo z bodu 1 vykročíme do 2 , nebo z bodu -1 vykročíme do -2 . Takto můžeme pokračovat a vytvořit schéma, které pro řetězec délky 4 vypadá následovně:

$$\begin{array}{cccccc} & & & & & 1 \\ & & & & & 1 \\ & & & & 1 & 1 \\ & & & 1 & 2 & 1 & . \\ & & 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

Řádky udávají délku řetězce n , sloupce polohu k (nula uprostřed) a čísla v tabulce počty možností, jak se do bodu dostat. Konstruujeme tedy Pascalův trojúhelník, přičemž po sudém počtu kroků můžeme být vždy pouze ve vzdálenosti od počátku dané sudým číslem, podobně pro lichý počet kroků lichým číslem. Prvky v tomto Pascalově trojúhelníku lze vyjádřit pomocí kombinačního čísla

$$\binom{n}{(n-k)/2}.$$

Pravděpodobnost realizace jednoho konkrétního řetězce je $1/2^n$, hledaná pravděpodobnost, že po n krocích se nacházíme v bodě k , je tedy

$$2^{-n} \binom{n}{(n-k)/2}.$$

Nalézt střední polohu opilce je velice jednoduché.⁴ Platí

$$E(W_n) = E\left(\sum_{i=1}^n S_i\right) = \sum_{i=1}^n E(S_i) = 0,$$

kde jsme využili toho, že střední hodnota součtu nezávislých veličin je rovna součtu jejich středních hodnot. Tento výsledek není nijak překvapivý, neboť každý řetězec délky n končí v bodě k k sobě má symetrický řetězec okolo počátku končící v bodě $-k$.

Poslední otázka je poněkud komplikovanější. Hledáme střední hodnotu $E(|W_n|)$, což je určitě nenulová hodnota. Odhadnout⁵ chování tohoto výrazu lze pomocí odmocniny ze střední hodnoty kvadrátu (střední kvadratická hodnota, *root mean square*), kterou opět díky vlastnostem střední hodnoty zapíšeme jako

$$E(W_n^2) = \sum_{i=1}^n \sum_{j=1}^n E(S_i S_j).$$

Pro $i = j$ máme $E(S_i^2) = 1$, pro $i \neq j$ je střední hodnota nulová, protože jde o součin nezávislých veličin, celkově tedy

$$\sqrt{E(W_n^2)} = \sqrt{\sum_{i=1}^n 1} = \sqrt{n}.$$

Řešení uvedených čtyř otázek nám poskytlo základní vhled do chování náhodných procházek, ale zdaleka se nejednalo o vyčerpávající rozbor. Studium náhodných procházek má význam v algebře, numerických metodách, ekonomii a dalších oborech. Dodnes před námi stojí mnoho nevyřešených otázek, především ohledně chování vícedimenzionálních procházek, a na ty se nyní podíváme.

⁴Při velkém obsahu alkoholu v krvi bude tato střední poloha pravděpodobně horizontální.

⁵Je ovšem možné spočítat $E(|W_n|)$ i přesně. Můžete si to zkusit.

Náhodná procházka ve více dimenzích

Definice náhodné procházky ve více dimenzích přináší dodatečnou komplikaci. Zatímco v 1D existovaly pouze dva možné směry, kam vykročit, nyní jich je nekonečně mnoho. Abychom si situaci ulehčili, budeme předpokládat, že v každém kroku je možné pohnout se pouze ve směru nebo proti směru jedné z kartézských os. Takže ve dvou dimenzích budeme mít čtyři možnosti pohybu, ve třech osm, \dots , obecně 2^d směrů, kde d je dimenze v prostoru. Specificky ve dvou dimenzích (v těch se budeme pohybovat především) jsou možné polohy v prostoru reprezentovány uzly čtvercové sítě.

Náhodná veličina K_i , která představuje jeden krok, může ve 2D nabývat čtyř hodnot, které si označíme třeba l , r , u , d (*left*, *right*, *up*, *down*). Také se na tuto veličinu můžeme dívat jako vektor se dvěma složkami, který může nabývat hodnot $(1, 0)$, $(0, 1)$, $(-1, 0)$ a $(0, -1)$, každé s pravděpodobností jedna čtvrtina. Náhodnou procházku pak definujeme opět přes součet

$$W_n = \sum_{i=1}^n K_i,$$

přičemž proces náhodné procházky $\{W_n\}$ je nyní tvořen posloupností vektorových náhodných veličin.

Otázky, které jsme řešili pro případ 1D procházky, lze přeformulovat i pro vícedimenzionální případ, ale nalezení odpovědi je již komplikovanější. Omezíme se pouze na otázky c) a d). Lze si rozmyslet, že střední polohou bude opět počátek, a to v jakémkoli počtu dimenzí. V souboru všech možných (realizací) procházek najdeme ke každé procházce právě jednu další symetrickou okolo počátku a střední poloha této dvojice je nulová, tedy i střední hodnota pro celý soubor bude nulová. Střední vzdálenost opět pouze odhadneme pomocí střední kvadratické hodnoty

$$\sqrt{\mathbb{E}(W_n^2)} = \left(\sum_{i=1}^n \sum_{j=1}^n \mathbb{E}(K_i \cdot K_j) \right)^{1/2} = \left(\sum_{i=1}^n \mathbb{E}(K_i \cdot K_i) \right)^{1/2} = \sqrt{n}.$$

Dostali jsme stejný výsledek jako pro jednu dimenzi díky skutečnosti, že skalární součin dvou identických vektorů K_i je roven jedné. Vzdáleností se zde myslí eukleidovská vzdálenost

$$|(x_1, x_2, \dots, x_n)| = \left(\sum_{i=1}^n x_i^2 \right)^{1/2},$$

skalární součin pak udává čtverec vzdálenosti. Samotná střední vzdálenost je (v asymptotickém smyslu, tedy pro $n \rightarrow \infty$) ještě násobena faktorem, který se blíží určité malé hodnotě z intervalu $(0, 1)$.

Pro algebraiky je důležitou otázkou, zda se procházka někdy navrátí do svého počátku. Bez důkazu zde uvedme, že dvoudimenzionální procházka v limitě nekonečného počtu kroků alespoň jednou dosáhne každého bodu, a tedy i toho počátečního. Konvergence pravděpodobnosti návratu k hodnotě 1 je ale velmi pomalá, proto v oblasti simulací není příliš důležitá. Pro 1D procházky platí silnější tvrzení, že v limitě nekonečného počtu kroků projdeme každým bodem nekonečněkrát.

Jako motivaci pro studium náhodných procházek jsme na začátku uvedli Brownův pohyb. Avšak správný model Brownova pohybu by měl být v čase spojitý. Provedeme-li pro náhodnou

procházku s časovým krokem Δt limitu $\Delta t \rightarrow 0$, dostaneme tzv. Wienerův proces. Na počítači však není možné provést simulaci, která je v čase spojitá. Jelikož však naše definice náhodné procházky pevně svazuje časový krok Δt a délku kroku v prostoru Δl , můžeme se Wienerově procesu (a tedy i Brownově pohybu) přiblížit přeškálováním rozměru procházky. Jinými slovy, limitu $\Delta t \rightarrow 0$ nahradíme limitou $\Delta l \rightarrow 0$. Abychom toto chování ozřejmili, ukážeme si nyní, jak vypadají výsledky simulace náhodné procházky.

Pro simulaci jednoduché 2D procházky jsme použili následující kód psaný v Pythonu:

```
# nacteme matematickou knihovnu a-knihovnu s-generatory nahodnych cisel,
# numerickou knihovnu a-grafickou knihovnu
import math
import random
import numpy as np
from matplotlib import pyplot as plt

# zadame delku nahodne prochazky a-pocet opakovani
max_krok = 10000
max_pocet = 10000
pocet = 0
# definujeme pole, do ktereho budeme ukladat stredni vzdalenost od pocatku
vzd = np.zeros(max_krok+1,dtype=np.float)

while pocet < max_pocet:
    pocet += 1
    # definujeme pole, do kterych budeme ukladat trajektorii posledni prochazky
    xs = np.zeros(max_krok+1,dtype=np.float)
    ys = np.zeros(max_krok+1,dtype=np.float)
    # resetujeme aktualni pocet kroku
    krok = 0
    while krok < max_krok:
        krok += 1
        # generujeme dve nahodna cisla z-mnoziny {0,1}, ktera rozhodnou,
        # zda se budeme pohybovat po vertikale, nebo horizontale
        # a-jestli v-kladnem, nebo zapornem smeru
        hv = math.floor(random.random()+0.5)
        pm = math.floor(random.random()+0.5)
        # naplnime dalsi element pole trajektorii predchozi hodnotou
        xs[krok] = xs[krok-1]
        ys[krok] = ys[krok-1]
        # posuneme se o-jeden krok
        if hv == 0:
            if pm == 0:
                xs[krok] += -1
            else:
                xs[krok] += 1
        else:
            if pm == 0:
                ys[krok] += -1
            else:
                ys[krok] += 1
        # pricteme vzdalenost aktualni prochazky po aktualnim poctu kroku
        vzd[krok] += math.sqrt(xs[krok]**2. + ys[krok]**2.)

# pole vzdalenessi vydělíme počtem procházek, abychom dostali prumer
vzd = vzd/float(max_pocet)
```

Tento kód generuje `max_pocet` náhodných procházek o délce `max_krok`, přičemž vzdálenost procházky v každém kroku od počátku ukládá do pole `vzd`. Na konci každý prvek tohoto pole obsahuje součet vzdáleností po počtu kroků rovnému pozici prvku. Vydělením tohoto pole počtem všech procházek dostaneme závislost průměrné vzdálenosti na počtu kroků. Průměrování

velkého počtu procházek je důležité – zvolíme-li malou hodnotu `max_pocet`, může vzdálenost od počátku i klesat, pokud se v našem souboru zrovna vyskytne více procházek, které se vrací k počátku.

Pro uvedené hodnoty jsme zjistili, že průměrná vzdálenost po 10 000 krocích je 89,2. To docela dobře odpovídá asymptotické hodnotě⁶ 88,6. Spíše by nás však zajímalo, jak vypadá závislost střední vzdálenosti na počtu kroků a jak vůbec vypadá realizace náhodné procházky (trajektorie). Na konec kódu výše proto přidáme následující:

```
# fitujeme přímku v-log-log grafu na posledních (fit_body) bodech
fit_body = 1000
fit_range = np.linspace(max_krok-fit_body+1,max_krok,fit_body)
log_range = [math.log10(y) for y in fit_range]
log_vzd = [math.log10(y) for y in (vzd[max_krok-fit_body+1:])]
fit,cov = np.polyfit(log_range,log_vzd,1,cov=True)
print('Smernice fitu v-je {} se smerodatnou odchylkou {}'.format(fit[0],
    math.sqrt(cov[0,0])))

# nakreslime graf prumernych vzdalenosti a-trajektorie posledni prochazky
vzdplot = plt.plot(range(max_krok+1),vzd,'r',linewidth=2)
plt.show()
prochplot = plt.plot(xs,ys,linewidth=1)
plt.show()
```

Ze závislosti vzdálenosti na počtu kroků vezmeme posledních 1 000 bodů, provedeme logaritmickou transformaci a fitujeme daty polynom prvního řádu (přímku). Proměnná `fit` v sobě obsahuje koeficienty a , b lineárního fitu $f(x) = ax + b$ a proměnná `cov` je matice,⁷ na jejíž diagonále leží rozptyly odpovídající koeficientům a , b . Po dostatečném počtu kroků n by se měla vzdálenost od počátku chovat jako \sqrt{n} , směrnice přímky by tedy měla být rovna exponentu $1/2$. Výstupem jednoho běhu programu byl koeficient $a = 0,4890 \pm 0,0004$, což je poměrně blízko očekávané hodnotě 0.5 .⁸ Poslední řádku kódu vykreslují graf závislosti vzdálenosti na počtu kroků, abychom mohli asymptotické chování ověřit vizuálně (obr. 1), a také vykreslují trajektorii poslední vygenerované procházky. Příklady procházek o délkách 10 000 a 200 000 kroků jsou na obrázcích 2 a 3. V druhém případě již nedokážeme rozeznat jednotlivé kroky – trajektorie se blíží realizaci spojitého Brownova pohybu.

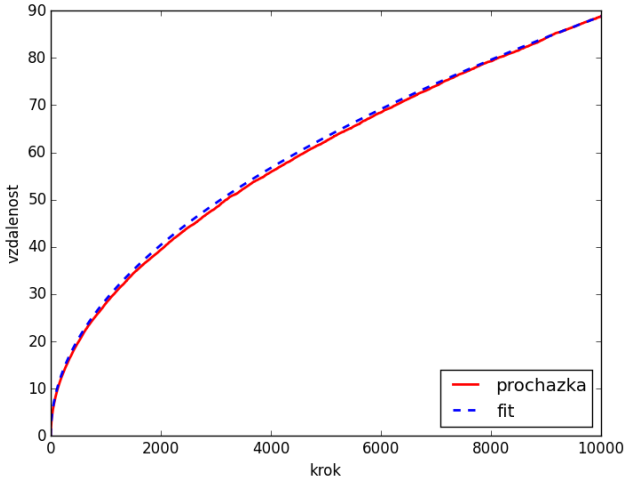
Kromě jednoduché procházky, které jsme se doteď věnovali, lze definovat ve 2D různé modifikace, které neměly v jedné dimenzi dobrý smysl. *Procházka bez návratu* v sobě zahrnuje dodatečný požadavek, že pokud nabývá K_i určité hodnoty, např. $(1, 0)$, nemůže K_{i+1} nabývat hodnoty opačné, $(-1, 0)$. *Procházka bez křížení* zakazuje, abychom do jakéhokoli bodu vstoupili výše než jednou. Na obrázku 2 bychom tedy viděli jednoznačný začátek a konec. Platí zajímavé⁹ tvrzení, že pravděpodobnost, že se náhodná procházka vrátí do počátečního bodu, roste s počtem kroků k jedné. Z toho nutně plyne, že každá procházka bez návratu jednou skončí. Nemá proto smysl studovat asymptotické chování střední vzdálenosti od počátku, ale můžeme například hledat střední délku procházky.

⁶Využili jsme znalosti multiplikatívního faktoru pro dimenzi dvě, který má hodnotu $\sqrt{\pi}/2$. V řešení seriálových úloh nepožadujeme dohledávat tyto faktory.

⁷Jedná se o kovarianční matici. Pokud se o kovarianci chcete dozvědět více, doporučujeme 3. díl seriálu z 30. ročníku FYKOSu.

⁸Pokud bychom chtěli zkusit provést výpočet pro delší procházky, trval by jeden běh programu již poměrně dlouho (minuty a více). Nezájímáme-li se o trajektorie procházek, můžeme si místo polí `xs` a `ys` ukládat pouze aktuální vzdálenost, čímž se běh programu výrazně zrychlí.

⁹Má využití například při maticovém zápisu integrálních rozvoji nebo při řešení Laplaceovy rovnice ve 2D.



Obr. 1: Vzdálenost 2D procházky od počátku v závislosti na počtu kroků. Graf vznikl průměrováním 10 000 procházek o délce 10 000 kroků. Mocninový fit byl proveden na posledních 1 000 bodech, v levé části grafu proto vidíme, že fit příliš nesedí – nevykazuje asymptotické chování.

Další možnou variací procházky je změna sítě, na které se pohybujeme. Řekněme, že použijeme například hexagonální mřížku. Potom z mikroskopického hlediska, tj. pro malý počet kroků, budeme pozorovat odlišnosti od případu s čtvercovou mřížkou. S rostoucím počtem kroků však tyto odlišnosti postupně mizí a procházka postupně konverguje k Brownovu pohybu. K tomuto chování dojde vždy, zachováme-li potřebnou symetrii ve volbě kroků.

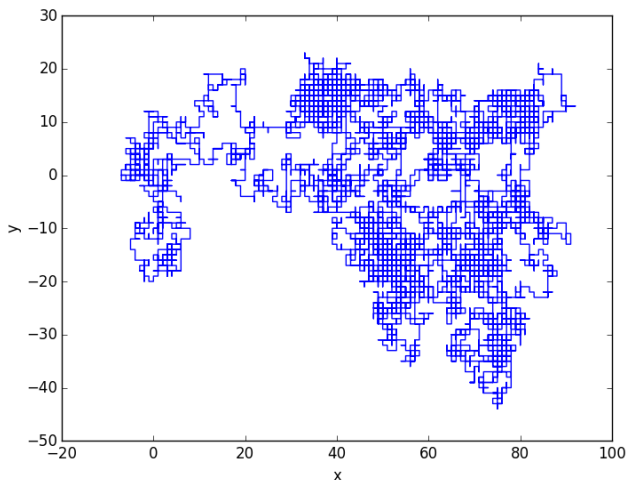
Náhodné procházky ve třech dimenzích¹⁰ a vyšších nejsou příliš dobře prozkoumané, alespoň z analytického hlediska. Jedním z analytických výsledků je, že s rostoucí dimenzí postupně klesá pravděpodobnost návratu k nule, zatímco v jedné i ve dvou dimenzích je rovna jedné, jak jsme zmiňovali výše. Pomocí počítačových simulací lze získat představu o asymptotickém chování délky procházky, pravděpodobnosti dosažení absorpční bariéry apod. To si vyzkoušíte v seriálové úloze. Ve velmi vysokých dimenzích nedokážeme simulacemi získat dobrou statistiku a nastupují zpět analytické metody.

Numerická integrace

Naším cílem je numericky zjistit hodnotu určitého integrálu

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

¹⁰Ti, kdo zažili éru Windows 95/98, si pravděpodobně vzpomenou na spořič obrazovky vykreslující spleť potrubí (Pipes screensaver). Jednalo se o 3D náhodnou procházku bez křížení s reflexivními okrajovými podmínkami vykreslovanou pomocí knihovny OpenGL. Problémem tohoto spořiče bylo, že mnohdy vytěžoval obrazovku i procesor více než běžná kancelářská práce.



Obr. 2: Trajektorie 2D procházky o délce 10 000 kroků. V grafu jsou dobře vidět jednotlivé pravoúhlé kroky, je tedy zřejmé, že se nejedná o spojitý proces.

Tato znalost je zvláště užitečná, neboť jen pro relativně málo funkcí dokážeme úlohu vyřešit analyticky, tedy nalezením primitivní funkce

$$F(x) = \int f(x) dx$$

a spočtením určitého integrálu vztahem

$$\int_a^b f(x) dx = F(b) - F(a).$$

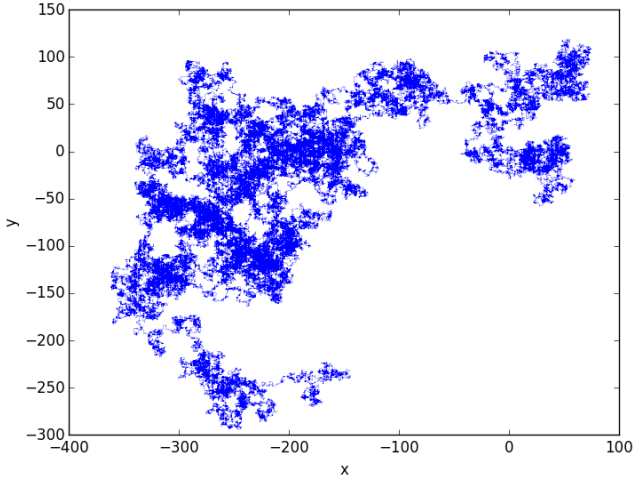
Vzpomeňme si nyní na poučku¹¹, že hodnota integrálu je rovna obsahu plochy pod grafem funkce $f(x)$ (s opačným znaménkem, pokud je funkce záporná). Celý problém se tedy redukuje na výpočet obsahu plochy. První, co nás napadne, je aproximovat plochu obdélníkem o stranách $b - a$ a $f(\frac{a+b}{2})$. Jde o jednu z variant *obdélníkového pravidla*, nicméně jak asi uhadneme, tato aproximace není pro většínu funkcí příliš přesná. Lepší rozhodně bude použít místo obdélníka lichoběžník. Aproximace integrálu je pak rovna

$$I = (b - a) \frac{f(a) + f(b)}{2}.$$

Pokud půjdeme v tomto duchu dále a aproximujeme plochu plochou pod parabolou, dostaneme jednoduché Simpsonovo pravidlo

$$I = \frac{b - a}{6} \left(f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right).$$

¹¹Ve skutečnosti jde o definici Riemannova integrálu.



Obr. 3: Trajektorie 2D procházky o délce 200 000 kroků. Diskrétní mřížka již není patrná, procházka se blíží spojitému procesu.

Dalo by se očekávat, že přesnějších metod analogicky dosáhneme aproximací plochou pod polynomem 3., 4., ... řádu. Tyto metody se ale nepoužívají kvůli neodstranitelným zaokrouhlovacím chybám.

Výše uvedené metody (vč. těch vyššího řádu) se nazývají *jednoduché Newtonovy-Cotesovy vzorce*. Zpravidla nám ale jednoduché Simpsonovo pravidlo nestačí, proto si představíme *složené Newtonovy-Cotesovy vzorce*. Myšlenka je jednoduchá. Rozdělme interval $\langle a, b \rangle$ na mnoho stejně velkých na sebe navazujících intervalů. Na každém z nich potom použijeme jednoduchý N-C vzorec, celkový integrál je pak součtem integrálů přes tyto malé intervaly. Čím je počet podintervalů větší, tím je výpočet samozřejmě přesnější. Složené lichoběžníkové, resp. Simpsonovo pravidlo pak má tvar

$$I_{\text{lichoběžník}} = h \left(\frac{f_0}{2} + \sum_{i=1}^{N-1} f_i + \frac{f_N}{2} \right),$$

$$I_{\text{Simpson}} = \frac{h}{6} (f_0 + 4f_{1/2} + 2f_1 + 4f_{3/2} + \dots + 4f_{N-1/2} + f_N),$$

kde $h = (b - a)/N$, $f_k = f(a + hk)$ a $f_{k/2} = f(a + hk/2)$. V praxi se pak používají právě tato dvě pravidla.

Představili jsme si základní metody výpočtu určitého integrálu v jedné dimenzi. V praxi se pak používají i jiné, pokročilejší metody,¹² pro jejich použití je ale nutné ovládat pokročilý matematický aparát, proto se jimi zde nebudeme zabývat. N-C vzorce lze zobecnit i pro vyšší dimenze, vzpomeňme si ale na minulý díl, kde jsme zjistili, že pro $d > 4$ je výhodnější použít

¹²Například Gaussovy kvadratury.

Monte Carlo integraci. Za určitých okolností, například pokud oblast, přes kterou integrujeme není pravouhlá, vyplatí se kvůli jednoduchosti použít MC integraci i v nižších dimenzích.

Představme si nyní, že chceme spočítat integrál z funkce, která má někde peak. Vzpomeňme si, že chyba (směrodatná odchylka) MC metody je $\sigma[I] = \sigma[f(x)]/\sqrt{N}$, kde N je počet vzorků a $\sigma[f(x)]$ rozptyl hodnot integrované funkce na daném intervalu. Pokud má ale funkce někde peak, pak tento rozptyl bude velký a v důsledku bude mít vypočtený integrál velkou chybu. To můžeme zachránit buď značným zvětšením počtu vzorků nebo zvolením jiného, než rovnoměrného rozdělení náhodných vzorků. Pokud totiž použijeme náhodné vzorkování s hustotou pravděpodobnosti $p(x)$, pak je hodnota integrálu rovna

$$I = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}.$$

a platí (viz obecné odvození v minulém díle seriálu)

$$\sigma[I] = \frac{1}{\sqrt{N}} \sqrt{\text{Var} \left[\frac{f(x)}{p(x)} \right]}.$$

Pokud bude $p(x)$ zvoleno tak, že $f(x)/p(x)$ bude skoro konstanta (nebo alespoň nebude obsahovat peaky) bude směrodatná odchylka hodnoty integrálu daleko menší. Abychom ale dokázali tohoto triku využít, musíme se naučit generovat náhodná čísla s libovolnou hustotou pravděpodobnosti.

Generování náhodných čísel s obecnou hustotou pravděpodobnosti

Zopakujme si nejprve některé základní pojmy týkající se spojitých náhodných reálných proměnných. *Hustotou pravděpodobnosti* nazýváme funkci $f(x)$ takovou, že

$$P[a \leq X \leq b] = \int_a^b f(x) dx.$$

Čteme: Pravděpodobnost, že náhodná proměnná X leží v intervalu $\langle a, b \rangle$ je rovna integrálu z hustoty pravděpodobnosti s mezemi a a b . Jinak řečeno, hustota pravděpodobnosti $f(x)$ vyjadřuje pravděpodobnost, že X leží v úzkém intervalu $\langle x, x + dx \rangle$. Je zřejmé, že $f(x)$ je všude nezáporná.

Druhým způsobem, jak popsat rozložení pravděpodobnosti, je *distribuční funkce* definovaná vztahem

$$F(x) = P[X \leq x].$$

Čteme: hodnota distribuční funkce v bodě x je rovna pravděpodobnosti, že náhodná proměnná X je menší nebo rovna x . Mezi distribuční funkcí a hustotou pravděpodobnosti platí vztah

$$F(x) = \int_{-\infty}^x f(t) dt.$$

Je zřejmé, že $F(x)$ je neklesající funkce a platí $F(-\infty) = 0$ a $F(+\infty) = 1$.

Inverze distribuční funkce

Prvním způsobem generování náhodných čísel s libovolnou distribucí je následující tvrzení. Mějme náhodnou proměnnou X s distribuční funkcí $F_X(x)$. Pak proměnná Y definovaná výrazem $Y = F_X(X)$ má rovnoměrné rozložení na intervalu $(0, 1)$. Důkaz je jednoduchý¹³, platí

$$F_Y(y) = P(Y \leq y) = P(F_X(X) \leq y) = P(X \leq F_X^{-1}(y)) = F_X(F_X^{-1}(y)) = y,$$

tedy $F_Y(y)$ je distribuční funkce rovnoměrného rozdělení. Tvrzení nám prakticky poslouží, pokud dokážeme najít inverzi kžžené distribuční funkce. Pak totiž dokážeme generovat náhodná čísla s distribuční funkcí $F(x)$ následujícím postupem.

1. Vygenerujeme náhodné číslo y s rovnoměrným rozdělením v intervalu $\langle 0, 1 \rangle$.
2. Kžžené náhodné číslo x s distribuční funkcí $F(x)$ získáme transformací $x = F^{-1}(y)$.

Problém je ale v tom, že musíme být schopni určit (ideálně analyticky) inverzi distribuční funkce, což zdaleka nedokážeme pro každou distribuci. Pak je nutné se poohlédnout po jiných metodách.

Boxova-Mullerova transformace

Jednou z aplikací inverze distribuční funkce je efektivní metoda generování náhodných čísel s normálním rozdělením.¹⁴ To má hustotu pravděpodobnosti

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

Je známo, že integrál hustoty pravděpodobnosti, resp. distribuční funkce nelze vyjádřit pomocí elementárních funkcí, ale jenom jako tzv. chybová funkce, která je tímto integrálem definována. S nalezením jednoduché inverze se pak můžeme rozloučit. Můžeme ale použít trik. Vezměme normální rozložení ve dvou dimenzích s navzájem nekorelovanými složkami

$$f(x, y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}},$$

neboli v polárních souřadnicích

$$f(\varrho, \vartheta) = \frac{1}{2\pi} e^{-\frac{\varrho^2}{2}}$$

a pokusme se spočítat distribuční funkci vzhledem k radiální souřadnici

$$\begin{aligned} F(r) &= \int_0^r \int_0^{2\pi} f(\varrho, \vartheta) \varrho \, d\vartheta \, d\varrho \\ &= \int_0^r \int_0^{2\pi} \frac{1}{2\pi} e^{-\frac{\varrho^2}{2}} \varrho \, d\vartheta \, d\varrho \\ &= \int_0^r e^{-\frac{\varrho^2}{2}} \varrho \, d\varrho \quad \left/ t = \frac{\varrho^2}{2}, dt = \varrho \, d\varrho \right. \\ &= \int_0^r e^{-t} dt = 1 - e^{-\frac{r^2}{2}}. \end{aligned}$$

¹³Převzato z https://en.wikipedia.org/wiki/Probability_integral_transform.

¹⁴Neboli Gaussovým rozdělením se středem $\mu = 0$ a rozptylem $\sigma^2 = 1$.

Nyní již můžeme distribuční funkci invertovat

$$F^{-1}(u_1) = \sqrt{-2 \ln(1 - u_1)}.$$

Pokud tento vztah použijeme ke generování náhodných čísel s distribucí $F(r)$, tedy u_1 bude mít rovnoměrné rozdělení v intervalu $\langle 0, 1 \rangle$, pak má u_1 a $1 - u_1$ stejné rozdělení, můžeme tedy psát

$$r = \sqrt{-2 \ln u_1}.$$

Tato distribuční funkce ale nepopisuje 1D normální rozdělení, ale rozdělení dané 2D Gaussovým „kopečkem“. Pokud ale tento kopeček řízeme podél libovolného radiálního směru, dostaneme 1D normální rozdělení. Proto si vygenerujeme ještě jednu náhodnou proměnnou s rovnoměrným rozdělením u_2 na intervalu $\langle 0, 1 \rangle$, která bude vyjadřovat úhel řezu. Pak proměnné x_1 a x_2 získané vztahy

$$\begin{aligned} x_1 &= \sqrt{-2 \ln u_1} \cos(2\pi u_2), \\ x_2 &= \sqrt{-2 \ln u_1} \sin(2\pi u_2) \end{aligned}$$

mají normální rozdělení a jsou statisticky nezávislé. Podařilo se nám tedy najít způsob, jak dvě rovnoměrně rozdělená náhodná čísla transformovat na dvě normálně rozdělená náhodná čísla. Této transformaci se říká *Boxova-Mullerova transformace*.

Pro úplnost dodejme, že pokud máme náhodnou proměnnou s normálním rozdělením x , pak náhodná proměnná y získaná transformací

$$y = \mu + \sigma x$$

má Gaussovo rozdělení se středem μ a rozptylem σ^2 .

von Neumannova metoda

Tato metoda dokáže transformovat dvě rovnoměrně rozdělená náhodná čísla na jedno číslo rozdělené dle obecné hustoty pravděpodobnosti $f(x)$. Nejprve je potřeba zvolit číslo M takové, aby $\forall x, f(x) < M$. Dále najdeme meze a, b tak, aby v intervalu $\langle a, b \rangle$ ležela všechna x , pro která $f(x) > 0$. Pak aplikujme následující algoritmus.

1. Vygenerujeme náhodné číslo u_1 s rovnoměrným rozdělením v intervalu $\langle a, b \rangle$.
2. Vygenerujeme náhodné číslo u_2 s rovnoměrným rozdělením v intervalu $\langle 0, M \rangle$.
3. Pokud $u_2 < f(u_1)$, pak u_1 použijeme, jinak jej zahodíme.
4. Opakujeme od začátku.

Metoda funguje, protože pravděpodobnost, že bude hodnota u_1 přijata je rovna $f(u_1)/M$, je tedy úměrná hustotě pravděpodobnosti.

Z popisu algoritmu je vidět, že M musíme volit co možná nejmenší, jinak budeme velmi plynout náhodnými čísly. Tato metoda je ze stejného důvodu nevhodná pro hustoty pravděpodobnosti, které mají peak, či obecně velký rozptyl hodnot. Nicméně principiálně hustotu pravděpodobnosti takřka vždy známe, algoritmus jde tedy často použít, ač není optimální.

Metropolisův-Hastingsův algoritmus

Metropolisův-Hastingsův algoritmus je pozoruhodná aplikace náhodné procházky. Předpokládejme, že známe hustotu pravděpodobnosti $f(x)$ náhodné veličiny, jejíž hodnoty chceme generovat. Zvolme si výchozí hodnotu x_0 a tzv. proposal density $g(x|y)$, neboli hustotu pravděpodobnosti návrhu přeskočení z bodu y do bodu x . Tato distribuce ale musí být symetrická ve smyslu, že musí platit $g(x|y) = g(y|x)$. Více si o ní povíme za chvíli. Nyní již můžeme generovat náhodná čísla dle následujícího algoritmu.

1. Vygenerujeme náhodného kandidáta pro novou polohu x' dle hustoty pravděpodobnosti $g(x'|x_i)$. (x_i je aktuální poloha)
2. Spočítáme koeficient přijetí $\alpha = f(x')/f(x_i)$.
3. Pokud $\alpha > 1$, pak novou polohu přijmeme ($x_{i+1} = x'$). Jinak vygenerujeme rovnoměrně rozdělené náhodné číslo u v intervalu $\langle 0, 1 \rangle$. Pokud $\alpha > u$, pak také krok přijmeme, jinak zůstaneme na místě ($x_{i+1} = x_i$).
4. Hodnotu x_{i+1} využijeme jako náhodné číslo s rozdělením $f(x)$. Iterujeme pro další hodnoty.

Algoritmus si tedy můžeme představit jako chození po horách (funkci $f(x)$). Jsme vášniví horolezci a čím jsme výš, tím jsme tam radši (nadmořská výška zároveň odpovídá hustotě pravděpodobnosti našeho výskytu). Odněkud vyjdeme a chceme udělat krok s náhodně zvolenou délkou (a směrem, neboť x je obecně vektor) dle distribuce $g(x'|x_i)$. Pokud bychom krokem vyšplhali výš, tak neváháme a jdeme tam. Pokud bychom sešli níž, tak se nám tam sice moc nechce, ale sem tam slézt musíme, tak si ještě hodíme kostkou (vygenerujeme náhodné číslo mezi nulou a naší aktuální nadmořskou výškou). Pokud je toto číslo menší než nadmořská výška, kam míříme, tak nám štěstí nepřálo a slezeme. Jinak zůstaneme kde jsme. Pak uděláme další náhodný krok ...

Vraťme se nyní k volbě distribuce $g(x|y)$. Jak už jsme zmínili, může to být libovolná distribuce splňující $g(x|y) = g(y|x)$ (ve zobecněné verzi algoritmu ani to nemusí platit), musíme si ale dát pozor na to, aby navrhované kroky ležely většinou v oblasti, kde je $f(x)$ nenulová. V opačném případě vzroste poměr zamítnutých kroků a spomalí se výpočet. Samozřejmě pak musíme volit distribuci $g(x|y)$ tak, abychom byli schopni rychle a spolehlivě generovat náhodné veličiny s daným rozdělením. Tento samozřejmý fakt je ale zpravidla tím limitujícím, v praxi se tedy nejčastěji používá buď rovnoměrné rozdělení centované v aktuální poloze y , nebo Gaussovo rozdělení opět se středem v aktuální poloze y .

Hned vidíme nedostatek tohoto algoritmu, a to je autokorelace. Pokud totiž nebudeme dělat zcela náhodně velké skoky přes celou oblast, budou sousední hodnoty vždy relativně blízko sebe, tedy korelované. Nicméně pro účely MC integrace, pokud použijeme dostatek hodnot, nám tato vlastnost příliš nevádí. Velkou výhodou tohoto algoritmu je pak to, že lze jednoduše a efektivně použít i ve vyšších dimenzích.

Správná volba distribuce délky kroku má zásadní důsledky. Představme si, že je $f(x)$ tvořena dvěma kopci a hlubokým údolím mezi nimi. Pak při malých krocích vyšplháme na jeden z vrcholů, ale už se nejspíš nedostaneme na ten druhý (museli bychom mít nehorázné štěstí a sejít do údolí). Příliš dlouhé kroky jsou také špatné, neboť bychom se příliš často strefovali

do nižších partií, než kde jsme, a klesal by poměr nově přijatých kroků.

Fyzikální korespondenční seminář je organizován studenty MFF UK. Je zastřešen Oddělením pro vnější vztahy a propagaci MFF UK a podporován Ústavem teoretické fyziky MFF UK, jeho zaměstnanci a Jednotou českých matematiků a fyziků.

Toto dílo je šířeno pod licencí Creative Commons Attribution-Share Alike 3.0 Unported. Pro zobrazení kopie této licence navštivte <http://creativecommons.org/licenses/by-sa/3.0/>.