Počítačová cvičení

Petr Beremlijski, Marie Sadowská

Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB - Technická univerzita Ostrava

Cvičení 1 - Matlab - nástroj pro matematické modelování

Abychom se mohli věnovat numerickému řešení matematických úloh, potřebujeme vhodné prostředí, které nám to umožní. A tak jako fyzik či chemik mají svou laboratoř nebo patolog pitevnu, mají i numeričtí matematici svojí Maticovou laboratoř[1] - Matlab. Podrobně se tomuto pracovnímu prostředí a jeho příkazům věnuje přiložený Matlabovský slabikář[2]. My si v tomto textu uvedeme pouze stručný přehled matlabovských proměnných a příkazů, kterým se budeme věnovat.

Prostředí

help, demos, intro, who, whos, clear, size, length

Proměnné

- Skaláry
- Vektory
- Matice

Příkazy

- Skalární funkce - *sin, cos, tan, exp, log, abs, sqrt, round*
- Vektorové funkce a generování vektorů - *max, min, sort*
- Maticové funkce a generování matic - *det, rand, ones, zeros, eye*
- Skalární operace - $+,\ -,\ *,\ /,\ \hat{}$
- Maticové a vektorové operace - $+,\ -,\ *,\ '$ (transponování), $\backslash$ ($A\backslash v = x \Leftrightarrow Ax = v$)
  Operace "po prvcích" - $.*,\ .\hat{}\ ,\ ./$
- 2D grafika (vykreslení grafů funkcí jedné proměnné) - *plot, hold on, hold off, figure*
- 3D grafika (vykreslení grafů funkcí dvou proměnných) - *meshgrid, mesh, contour, hold on, hold off, figure*
- Řídící příkazy - *if* (podmíněný příkaz) *for, while* (příkazy cyklu se známým počtem opakování a podmínkou na začátku)

[1] MATrix LABoratory
[2] K. Sigmon - MATLAB Primer

- Relace a logické operace - <, >, <=, >=, ==, ~=, &, |, ~
- Skripty a funkce - *function*

Vše si vyzkoušíme při řešení následujících úloh.

Příklad 1 Kolik členů harmonické řady[3] musíte nejméně sečíst, aby tento částečný součet řady měl hodnotu alespoň 10 (15, 20)?

Příklad 2 Zkuste odhadnout s využitím Matlabu součet řady $\sum_{n=1}^{\infty} \left(\frac{1}{n} \cdot \frac{1}{n+1}\right)$.

Příklad 3 Sestrojte grafy následujících funkcí:

- $f(x) = x^2$
- $f(x) = \sqrt{1 - x^2}$
- $f(x) = x^2 \cdot \sin\left(\frac{1}{x^2}\right)$
- $f(x) = |x|$

Příklad 4 Sestrojte grafy následujících funkcí:

- $f(x, y) = x^2 + y^2$
- $f(x, y) = \sqrt{x^2 + y^2}$
- $f(x, y) = (x^2 + y^2) \cdot \sin\left(\frac{1}{x^2+y^2}\right)$
- $f(x, y) = \sqrt{|xy|}$

[3] Řadou (reálných čísel) rozumíme výraz $a_1 + a_2 + \cdots + a_n + \cdots = \sum_{n=1}^{\infty} a_n$, kde pro každé $n \in \mathbb{N}$ je $a_n \in \mathbb{R}$. Harmonickou nazýváme řadu $\sum_{n=1}^{\infty} \frac{1}{n}$.

Programy pro numerické výpočty, mezi něž patří i Matlab, umí vyřešit mnoho i velmi komplikovaných úloh v "rozumném" čase. Nejsou však všemocné a mají své limity. Vhodně to ilustruje tato úloha: Nalezněte nejmenší přirozené číslo n, pro které největší společný dělitel čísel $(n^{17}+9)$ a $((n+1)^{17}+9)$ není 1. Pokud byste chtěli tuto úlohu řešit počítačem, čekali byste opravdu dlouho, protože řešením této úlohy je

$$n = 8424432925592889329288197322308900672459420460792433^{4}.$$

My si "omezení" Matlabu uvědomíme díky následujícímu příkladu. V tomto příkladě budeme mít za úkol zjistit pravděpodobnost výskytu daného jevu. Nejjednodušší, ale ne nejméně pracnou, možností, jak vypočíst pravděpodobnost určitého jevu, je zjistit počet příznivých možností a podělit jej počtem všech možností.

Příklad 1 Mějme posloupnost přirozených čísel $1, 2, 3, \ldots, n$. Poté ji náhodně promíchejte. Jaká je pravděpodobnost, že ani jedno číslo nebude na své původní pozici?[5] Vyřešte pomocí Matlabu. Pro jak velké n jste schopni tuto pravděpodobnost zjistit?

Nápověda: Pro algoritmizaci úlohy použijte rekurzi[6].

[4]Odhadněte, jak dlouho budete čekat na nalezení řešení, pokud máte počítač, který zvládne miliardu operací za sekundu a předpokládáte, že nalezení největšího společného dělitele pro jedno konkrétní n se dá dosáhnout jednou operací.

[5]Všimněte si, jaké číslo získáte, pokud vypočtete převrácenou hodnotu zjištěné pravděpodobnosti.

[6]V programování rekurze představuje opakované vnořené volání stejné funkce (podprogramu), takovou funkci pak nazveme rekurzivní. Součástí rekurzivní funkce musí být ukončující podmínka, která určuje, kdy se má vnořování zastavit. Pro použití rekurze je zapotřebí, aby programovací jazyk umožňoval volání podprogramu ještě před ukončením jeho předchozího volání. Po každém kroku volání sebe sama dochází ke zjednodušení problému, pokud není splněna ukončující podmínka, provede se rekurzivní krok.

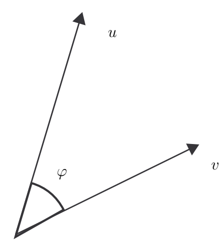
Podívejme se na dvě aplikace aritmetických vektorů a matic.

Vyhledávání v databázích

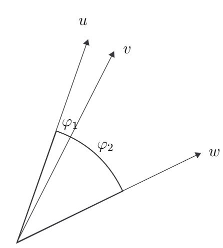
Nejprve si zavedeme pojem aritmetický vektor. **N-rozměrný aritmetický vektor** je uspořádaná n-tice čísel, jejíž prvky se nazývají složky. Tyto uspořádané n-tice budeme zapisovat do hranatých závorek do řádků nebo sloupců. Připomeňme si **skalární součin** dvou aritmetických vektorů o n složkách, který je definován vztahem

$$u \cdot v = ||u|| \, ||v|| \cos \varphi,$$

kde $||u|| = \sqrt{\sum_{i=1}^{n} u_i^2}$ a φ je úhel, který svírají vektory u a v.



O vektorech řekneme, že jsou si “blízké”, pokud úhel φ je blízký 0. Např. na následujícím obrázku je vektor u “blízký” v.



“Blízkosti” vektorů můžeme využít při vyhledávání v databázích. Ilustrujme si to na následujícím příkladě. Mějme kuchařku, která obsahuje n receptů, kde každý recept obsahuje některou z k surovin. Takovou databázi receptů můžeme popsat n vektory $r_1, r_2, \ldots, r_n$ o k složkách. Pokud chceme vybrat recept se surovinami, které nejlépe splňují náš požadovaný výběr, pak můžeme náš požadavek zapsat jako vektor p o k složkách a naším úkolem je najít vektor r_i z vektorů $r_1, r_2, \ldots, r_n$, který je “nejbližší” vektoru p (tj. platí, že $\frac{r_i \cdot p}{||r_i|| \, ||p||} = \cos \varphi_i$ je největší ze všech $\frac{r_l \cdot p}{||r_l|| \, ||p||} = \cos \varphi_l \; \forall l = 1, \ldots, n$).

Příklad 1 Popište následující recepty pomocí vektorů jako v předchozím textu:

- Buchty - ingredience: 100 g Hery, 100 g cukru moučka, 2 vejce, 500 g hladké mouky, 1/4 l mléka, 30 g droždí
- Čínské placičky - ingredience: 3 silnější kuřecí řízky, 3 vejce, 1 drobně nakrájená cibule, 4 lžíce Solamylu, sůl, 4 lžíce oleje, 1 lžíce sojové omáčky, 1 lžíce worcesterové omáčky
- Čočková polévka - ingredience: 1 lžíce olivového oleje, 1 mrkev, 1 cibule, 4 stroužky česneku, 50 g žitné mouky, 450 g čočky, sůl, 1 kostka zeleninového bujónu, podle chuti chilli nebo cayenský pepř, majoránka, 10g másla, 2 lžíce plnotučné hořčice, 2 vejce, petrželka
- Domácí buchty - ingredience: 600 g polohrubé mouky, 1/4 l mléka, 2 vejce, 10 lžic oleje, 40 g cukru, 1 prášek do pečiva, 1 kostička droždí (42 g), špetka soli
- Evíkova mňamka - ingredience: 4 kuřecí řízky, 8 plátků anglické slaniny, sýr podle chuti (eidam, blaťácké zlato, hermelín, niva..., ale vždy jen jeden druh), 1 cibule, sůl, pepř, 50 g másla, 1 lžíce sojové omáčky
- Chléb - ingredience: 1,5 lžíce octa, 3 lžíce olivového oleje, 10 g cukru, 3 lžičky soli, 360 g hladké mouky pšeničné, 140 g žitné mouky, 75 g celozrnné mouky žitné, 75 g celozrnné mouky pšeničné, lžíce kmínu, 15 g sušeného droždí
- Chlebíčková pomazánka - ingredience: 100 g brambor, 1 cibule, 1 lžíce tatarské omáčky, sůl, pepř
- Kokosová hrníčková bábovka - ingredience: 200 g hladké mouky, 10 lžic kokosu, 100 g cukru krupice, 1/4 l mléka, 6 lžic oleje, 1 prášek do pečiva, 1 vanilkový cukr, 3 vejce
- Kuřecí kousky v sýrovém těstíčku - ingredience: 4 kuřecích prsíček, sůl
Těstíčko: 1 vejce, 0,05 l bílého vína, 80 g hladké mouky, strouhaný sýr
Dresink: 100 g bílého jogurtu, 5 lžic tatarky, mletý bílý pepř, sůl, 2 stroužky česneku
- Pařížské kostky - ingredience: 440 g polohrubé mouky, 220 g cukru, 7 lžic oleje, 1/4 l vlažného mléka, 2-3 lžíce kakaa, 1 prášek do pečiva, 3 celá vejce
- Perník - ingredience: 1/2 kg polohrubé mouky, 350 g cukru krystal, 1/2 l mléka, 10 lžic oleje, 2 celá vejce, 4 lžíce rozředěných povidel, 2-3 lžíce kakaa, 1 lžička jedlé sody, 1 prášek do pečiva, špetka soli, 1 lžička mleté skořice, lze přidat najemno nastrouhaná 2-3 jablka
- Pizza těsto - ingredience: 0,5 kg hladké mouky, 1 lžíce olivového oleje, 1 lžíce soli, 15 g droždí
- Plněné kuře - ingredience: 1 kuře, 1 velká cibule, 3 plátky anglické slaniny, 2 lžíce oleje, sůl
- Plněný cop - ingredience: 500 g polohrubé mouky, 50 g moučkového cukru, 3 žloutky, 100 g rozpuštěného másla, špetka soli, 0,2 l mléka, 40 g droždí
Náplň: 5 lžic strouhaných lískových oříšků, 100 g cukru, 3 vejce, špetka skořice, vejce na potření, sekané oříšky na posypání
- Tatranské pracny - ingredience: 300 g hladké mouky, 250 g Hery, 100 g moučkového cukru, 1 vejce, 1 lžíce kakaa, 1 lžička skořice, 6 lžic mletých ořechů, oříšků nebo mandlí (může se i namíchat)
- Tradiční italské lasagne - ingredience: Boloňská omáčka: 1 lžíce olivového oleje, 1 střední cibule, 1 stroužek česneku, 500 g mletého hovězího (kuřecího, sójového masa, jaké kdo má rád), 250 g oloupaných celých či nakrájených rajčat v plechovce, 2 lžíce rajčatového protlaku, 4 lžíce červeného vína, sůl, čerstvě namletý pepř
Sýrová omáčka: 50 g másla, 50 g mouky, 0,6 l mléka, sýr na strouhání (Čedar) 12 listů vaječných těstovin - lasagní
- Vepřová kýta - ingredience: 6 řízků z kýty, 1 lžička soli, 1 lžička pepře, 1/2 lžíce solamylu, 1 lžíce worchestrové omáčky, 2 stroužky česneku, 1/2 lžičky majoránky, 2 vejce, 1 lžíce raj. protlaku, 1 lžíce oleje

Zkuste na základě “blízkosti” vyhledat, které recepty obsahují ingredience nejlépe odpovídající těmto třem požadavkům:

- 100 gramů cukru, 20 gramů droždí, 100 gramů Hery, 2 lžíce kakaa, 500 gramů mouky a skořice

- sůl, sýr a worchester
- 2 stroužky česneku, 1 lžíce hořčice, 3 kuřecí řízky, majoránka, pepř, 2 plátky slaniny a sůl

Použití matic při zápisu geometrických zobrazení

Začneme zavedením pojmu matice. Nechť jsou dány prvky $a_{11}, a_{12}, \ldots, a_{mn}$ z dané množiny $\mathcal{F}$. **Matice typu** (m, n) je obdélníková tabulka

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix},$$

která má $m \cdot n$ prvků a_{ij} uspořádaných do m řádků r_i^A a n sloupců s_j^A, takže

$$A = \begin{bmatrix} r_1^A \\ \vdots \\ r_m^A \end{bmatrix} = \left[s_1^A, \ldots, s_n^A\right],$$

$$r_i^A = [a_{i1}, \ldots, a_{in}], \; s_j^A = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix}.$$

Stručně píšeme též $A = [a_{ij}]$.
Nyní si zavedeme operaci násobení matic. K tomu ale potřebujeme nejdříve definovat násobení matice a vektoru. Součinem matice $A = [a_{ij}]$ typu (m, n) a sloupcového vektoru $x = [x_i]$ o n složkách nazýváme vektor y o m složkách definovaný předpisem

$$y = Ax = x_1 s_1^A + \cdots + x_n s_n^A.$$

Rozepsáním definice po složkách dostaneme

$$[y]_i = [Ax]_i = a_{i1}x_1 + \cdots + a_{in}x_n = r_i^A x.$$

Teď můžeme přejít k násobení matic. Jestliže A je matice typu (m, p) a B je matice typu (p, n), pak **součin matic** A **a** B je matice AB typu (m, n) definovaná předpisem

$$AB = \left[As_1^B, \ldots, As_n^B\right].$$

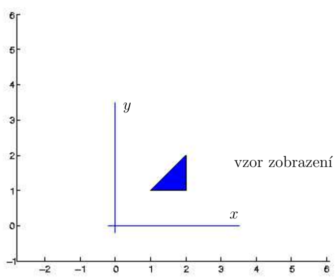
Rozepíšeme-li si definici násobení matic po složkách, dostaneme

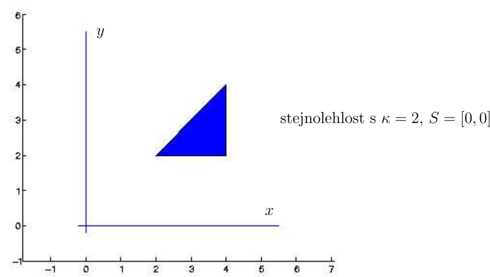
$$[AB]_{ij} = a_{i1}b_{1j} + \cdots + a_{ip}b_{pj} = r_i^A \cdot s_j^B.$$

Než použijeme výše zavedenou operaci, připomeneme si některé pojmy z oblasti geometrických zobrazení. **Geometrickým zobrazením** nazveme zobrazení, které každému bodu A útvaru U přiřazuje právě jeden bod A' útvaru U'. Bod A je tzv. vzor a bod A' se označuje jako obraz. My se budeme zabývat třemi zobrazeními, a to stejnolehlostí, rotací a posunutím.

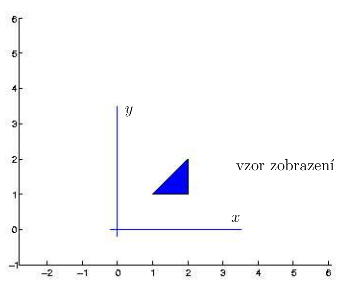
Pro připomenutí:

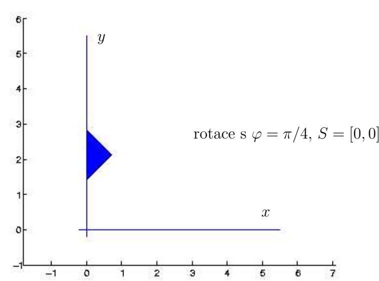
Stejnolehlost:
Mějme v rovině či prostoru ρ bod S. Geometrické zobrazení, při němž obrazem bodu S je bod S a obrazem každého $A \in \rho,\ A \neq S$, je takový bod $A' \in \rho$, že pro vektor SA' platí $SA' = \kappa SA$, kde $\kappa \in \mathbb{R}\setminus\{0, 1\}$ je pevně zvolené, se nazývá **stejnolehlostí** (homotetií). Bod S nazýváme středem stejnolehosti a κ koeficientem stejnolehlosti. Stejnolehlost s $\kappa = -1$ je středovou souměrností.



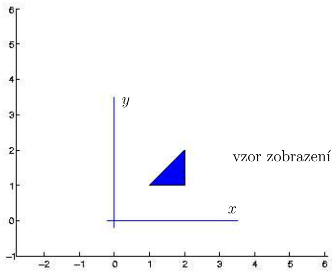


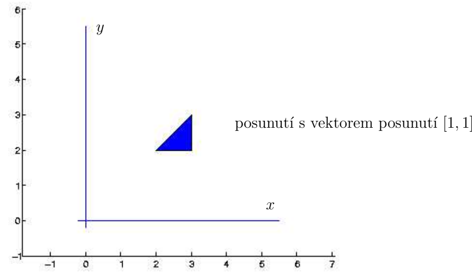
Rotace:
Otočení (**rotace**) v rovině je geometrické zobrazení, které je charakterizováno tím, že spojnice všech bodů s pevně zvoleným bodem S, tzv. středem otočení, se změní o stejný úhel φ, tzv. úhel otočení, a vzdálenost bodů od středu otočení zůstává nezměněna.





Posunutí:
Posunutí (translace) je geometrické zobrazení, které je charakterizováno tím, že všechny body transformované množiny bodů změní své kartézské souřadnice o stejnou hodnotu, tj.





ke každému bodu přičteme stejný vektor posunutí.

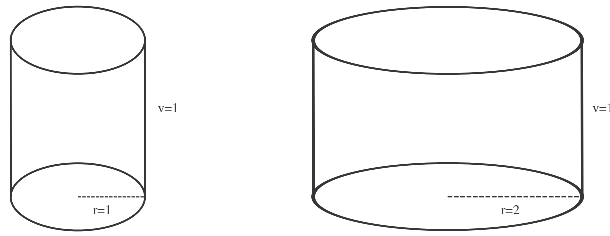
Nyní si ukážeme, jak lze pomocí elementárních maticových operací zapsat výše uvedená zobrazení. Zapišme souřadnice vzoru geometrického zobrazení, tj. bodu z $\mathbb{R}^2$ či $\mathbb{R}^3$, do sloupcového vektoru. Pokud vzorem zobrazení je více bodů, zapíšeme je jako sloupcové vektory do matice P. Obraz bodů ve stejnolehlosti s koeficientem κ a středem v počátku zapíšeme jako součin transformační matice T typu (n, n), kde n je rozměr prostoru, ve kterém zobrazujeme (2 nebo 3), a matice P. Matice T má všechny prvky na hlavní diagonále rovny koeficientu stejnolehlosti κ. Všechny další prvky jsou nulové. Obraz bodů v rotaci s úhlem otočení φ a středem otočení v počátku zapíšeme jako součin transformační matice R typu $(2, 2)$ a matice P, kde

$$R = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}.$$

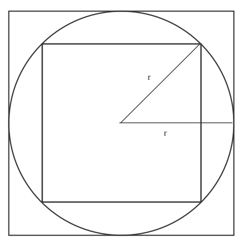
Posunutí s vektorem posunutí *pos* realizujeme tak, že ke každému sloupci matice P přičteme sloupcový vektor *pos*.

Příklad 2 Implementujte geometrická zobrazení - stejnolehlost, rotaci a posunutí. Nalezněte obraz trojúhelníku s vrcholy $[1, -1]$, $[2, 0]$ a $[1, 1]$ ve stejnolehlosti se středem v počátku a s koeficientem stejnolehlosti 2, v rotaci se středem otočení v počátku a úhlem otočení $\pi/2$ a v posunutí s vektorem posunutí $[1, -1]$. Zobrazte vzor a jednotlivé obrazy (použijte funkce *plot* a *patch*).

Hledejme obsah kruhu o poloměru r (představme si, že neznáme příslušný vzorec a nic nevíme o čísle π). Prvním nápadem by mohlo být zhotovení válcových nádob s různými poloměry podstav (např. s poloměry o délce 1 a 2 jednotek) a jednotkovou výškou.



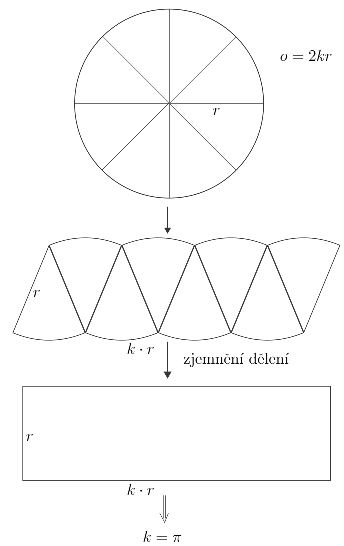
Objem vody, který se do takových nádob vejde, je roven obsahu podstavy válce, tj. obsahu kruhu s poloměrem r. Rychle si všimneme, že pokud zvětšíme poloměr podstavy dvakrát, zvětší se objem čtyřikrát a následně odvodíme, že obsah kruhu je přímo úměrný druhé mocnině poloměru. Také zjistíme, že druhou mocninu poloměru kruhu musíme vynásobit vhodnou konstantou, abychom dostali správnou hodnotu obsahu daného kruhu. Tuto konstantu označíme π. Existuje mnoho možností, jak tuto konstantu odhadnout. Snad nejjednodušší způsob, jak stanovit meze pro π, je vepsat do kruhu o poloměru r čtverec a stejnému kruhu opsat jiný čtverec.



Protože je snadné spočítat obsahy daných čtverců, zjistíme, že $\pi \in (2,4)$ (délka strany menšího čtverce je $\sqrt{2}r$, délka strany většího čtverce je $2r$). Mnohem rozumnější výsledek získáme, pokud budeme danému kruhu vepisovat n-úhelníky a počítat jejich obsahy. Tuto metodu nazýváme vyčerpávací (exhaustní) a pravděpodobně první ji použil Eudoxos[7].

Než se budeme věnovat odhadům čísla π, podívejme se krátce na výpočet obvodu kruhu. Jistě víme, že obvod kruhu je přímo úměrný dvojnásobku jeho poloměru, ale abychom dostali správnou hodnotu, je nutno $2r$ vynásobit vhodnou konstantou. Na následujícím obrázku provedeme přeuspořádání kruhu na útvar, který se pro zjemňující se dělení kruhu blíží obdélníku. Porovnáním obsahu kruhu a vzniklého obdélníku je názorně vidět, že tato konstanta je opět π.

[7] Eudoxos (410 nebo 408 př. n. l. – 355 nebo 347 př. n. l.) – řecký astronom, matematik a fyzik, student Platóna



Nyní si ukážeme několik způsobů, jak nalézt přibližnou hodnotu čísla π.

Buffonova metoda

Řešením tzv. Buffonova problému s jehlou[8] je aproximace čísla π. Úloha spočívá v opakovaném házení jehly o délce l na rovinu, na které máme vyznačenu síť rovnoběžek o vzdálenosti $s \geq l$. Jestliže jehlu hodíme n-krát a x-krát nám během těchto pokusů po dopadu zkříží některou z rovnoběžek, pak v případě, že $s = 2l$, číslo

$$\frac{n}{x}$$

aproximuje s danou přesností číslo π.

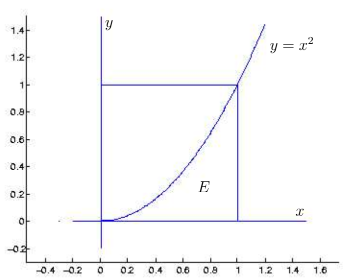
V roce 1975 pánové Perlman a Wichera publikovali tento výsledek týkající se přesnosti Buffonovy metody: S pravděpodobností 95 procent nemá chyba aproximace hodnotu větší než $5/\sqrt{n}$. Tzn. například pro 10000 pokusů nám s pravděpodobností 95 procent chyba nepřekročí hodnotu $0,05$.

Příklad 1 Implementujte Buffonovu metodu a použijte ji k aproximaci čísla π. Porovnejte vaši aproximaci se skutečnou hodnotou čísla π a určete chybu aproximace.

[8]G. L. Buffon (1707–1788) – francouzský přírodovědec

Monte Carlo je třída výpočetních algoritmů založená na provádění náhodných experimentů. Této metody se často používá pro simulaci fyzikálních a matematických systémů. Výsledkem provedení velkého množství experimentů je obvykle pravděpodobnost určitého jevu. Na základě získané pravděpodobnosti a známých vztahů pak spočítáme potřebné výsledky. Protože metoda vyžaduje generování velkého souboru náhodných dat, je vhodné pro její implementaci použití počítače. Metod Monte Carlo se používá v případě, kdy je příliš pracné nebo nemožné nalézt přesný výsledek jiným způsobem. Její výhodou je jednoduchá implementace, nevýhodou relativně malá přesnost. Metoda byla vytvořena skupinou fyziků pracujících na projektu jaderné pumy v Los Alamos, jméno metody bylo navrženo v roce 1940 von Neumannem[9].

V matematice se Monte Carlo používá zejména pro výpočet určitých integrálů (zejména vícenásobných určitých integrálů), které je obtížné či nemožné vyčíslit analyticky nebo jinou vhodnou numerickou metodou. Např. obsah plochy ohraničené shora grafem funkce $y = x^2$ na intervalu $\langle 0, 1\rangle$ (tj. $\int_0^1 x^2 \,\mathrm{d}x$) je možné metodou Monte Carlo vypočíst následujícím způsobem. Nechť náš program generuje náhodně dvojice čísel $[x, y]$, přičemž každé z čísel x a y je vybráno nezávisle z intervalu $\langle 0, 1\rangle$. Tuto dvojici budeme chápat jako souřadnice bodu, který je náhodně zvolen ve čtverci $\langle 0, 1\rangle \times \langle 0, 1\rangle$. Pravděpodobnost toho, že bod leží uvnitř zadaného čtverce, je 1. Pravděpodobnost toho, že bod leží uvnitř podmnožiny E jednotkového čtverce (tj. čtverce, jehož strana má jednotkovou velikost), je rovna obsahu plochy E. Takže obsah plochy, která je podmnožinou jednotkového čtverce, můžeme odhadnout jako pravděpodobnost, že náhodně zvolený bod leží v této podmnožině.

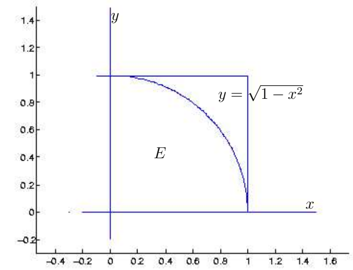


Příklad 2 Implementujte metodu Monte Carlo pro výpočet $\int_0^1 x^2 \,\mathrm{d}x$.

Pro odhad přesnoti metody Monte Carlo platí: S pravděpodobností 95 procent nemá chyba aproximace hodnotu větší než $1/\sqrt{n}$. Tzn. například pro 10000 pokusů nám s pravděpodobností 95 procent chyba nepřekročí hodnotu $0,01$.

Pokud chceme použít metodu Monte Carlo k aproximaci čísla π, vypočteme $\int_0^1 \sqrt{1 - x^2} \,\mathrm{d}x$. Snadno si uvědomíme, že tímto způsobem získáme aproximaci hodnoty $\pi/4$.

[9] John von Neumann (1903–1957) – maďarský matematik



Příklad 3 Implementujte metodu Monte Carlo pro výpočet $\int_0^1 \sqrt{1-x^2}\,\mathrm{d}x$ a použijte ji k aproximaci čísla π. Porovnejte vaši aproximaci se skutečnou hodnotou čísla π a určete chybu aproximace.

Řady

Poslední možností k aproximaci čísla π, kterou si v tomto přehledu ukážeme, je využití řady[10]. K této aproximaci použijeme Gregoryho řadu[11], která rozvíjí funkci arctg x:

$$\operatorname{arctg} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Tato řada má konečný součet pro $x \in \langle -1, 1 \rangle$ (říkáme, že konverguje), navíc platí, že čím je $|x|$ menší než 1, tím méně členů řady potřebujeme použít k nahrazení arctg x s "uspokojivou" přesností. První možností, jak aproximovat π, je tudíž nahradit arctg 1 Gregoryho řadou ($\operatorname{arctg} 1 = \pi/4$). Aproximaci s rychlejší konvergencí získáme, pokud použijeme rovnost $\operatorname{arctg} 1 = \operatorname{arctg}(1/2) + \operatorname{arctg}(1/3)$.

[10] Řadou (reálných čísel) rozumíme výraz $a_1 + a_2 + \cdots + a_n + \cdots = \sum_{n=1}^{\infty} a_n$, kde pro každé $n \in \mathbb{N}$ je $a_n \in \mathbb{R}$.

[11] James Gregory (1638 – 1675) – skotský matematik a astronom