

POČÍTAČOVÁ CVIČENÍ

ŠKOLA MATEMATICKÉHO MODELOVÁNÍ

2022

Petr Beremlijski, Michal Béréš,
Simona Béréšová, Rajko Čosić,
Marie Sadowská

Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB - Technická univerzita Ostrava
2022

Předmluva

Milí studenti,
moc rád Vás jménem Katedry aplikované matematiky a Jednoty českých matematiků a fyziků vítám na již 18. ročníku našeho semináře ŠKOMAM (ŠKOla MATematického Modelování).

Jako každý rok se Vám pokusíme v těch několika dnech ukázat, že **matematika je** nejenom velice **užitečná** a že nemálo jejích aplikací lze zvládnou již pomocí středoškolských znalostí a důvtipu, ale že je taky nesmírně **zábavná a inspirující**. Snad si všimnete, že se svou prací bavíme a že je nám spolu s našimi studenty (a proto již kolegy) na katedře dobře. Doufáme, že aspoň některé z Vás toto setkání a možnost si leccos vyzkoušet přiměje k úvahám (a k následnému rozhodnutí) přihlásit se ke studiu našeho programu **Výpočetní a aplikovaná matematika**. Byli bychom moc rádi.

Milí studenti, přeju Vám (i nám), ať si dny strávené na ŠKOMAMu pořádně užijeme.

V Orlové 24.1.2022

Jirka Bouchala

Úvod

Tento text je určen pro účastníky semináře Škola matematického modelování (<http://skomam.vsb.cz>) a slouží jako pomůcka k úlohám, které řeší studenti v průběhu tohoto semináře na počítačových cvičeních. Tento seminář, pro který používáme zkratku ŠKOMAM, organizuje Katedra aplikované matematiky (<http://am.vsb.cz>) Fakulty elektrotechniky a informatiky Vysoké školy báňské – Technické univerzity Ostrava jednou ročně již od roku 2005. V tomto roce probíhá 18. ročník tohoto semináře.

Pro počítačová cvičení budeme používat programovací jazyk Python, který navrhnul Guido van Rossum na konci 80. let. Python je (podobně jako jazyk R či jazyk Octave) volně přístupný (<https://www.python.org/>) a nyní patří celosvětově mezi nejoblíbenější a nejpoužívanější programovací jazyky. Jeho další velkou výhodou je možnost instalace na většinu běžných platforem jako je MS Windows, Unix, macOS nebo Android. Jedním z důvodů, proč jsme se pro Python rozhodli, je jeho jednoduchost pro učení i pro začátečníky. Podrobný popis tohoto jazyka je k dispozici v [1]. Pro snadnou práci během online cvičení budeme s programy v Pythonu pracovat ve webovém rozhraní Jupyter Notebook (<https://jupyter.org/>). Ke každému počítačovému cvičení budeme mít jeden nebo více těchto Jupyter Notebooků, které jsou uloženy na úložišti Binder (<https://mybinder.org/>). Díky tomu si účastníci semináře nemusejí nic instalovat a vše je pro ně přichystáno.



Tento seminář je pořádán s finanční podporou Fakulty elektrotechniky a informatiky (<http://www.fei.vsb.cz>), statutárního města Ostravy a projektu Math Exercises for You 2 (<http://math4u.vsb.cz>) podpořeného programem Erasmus+. Nad akcí převzala záštitu Jednota českých matematiků a fyziků (<http://jcmf.vsb.cz>).

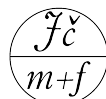
VŠB TECHNICKÁ | FAKULTA
UNIVERZITA | ELEKTROTECHNIKY
OSTRAVA | A INFORMATIKY

OSTRAVA!!!

Spolufinancováno
z programu Evropské unie
Erasmus+



MATH 4 U



Abychom se mohli věnovat pokročilejším matematickým úlohám, potřebujeme vhodné prostředí, které nám jejich řešení umožní. K tomuto účelu použijeme skriptovací programovací jazyk Python, konkrétně jazyk Python 3. Podrobně se o něm můžete dozvědět např. v textu „Učíme se programovat v jazyce Python 3“ (viz [1]).

V tomto cvičení se seznámíme se základním přehledem datových struktur a příkazů Pythonu 3. Protože práci s proměnnými, cykly a podmínkami, základními funkcemi a knihovnami, najdete podrobně popsánu v Jupyter Notebook „cv1_uvod“, který naleznete na adrese

<https://mybinder.org/v2/gh/Beremi/SKOMAM/main>

a také jsme připravili video na YouTube provádějící tímto cvičením, které najdete v playlistu

<https://youtube.com/playlist?list=PLVrbfTRvGpyU2u-4iSNoGh8tprKIo9hvx>,

uvedeme si v tomto textu jen úkoly k procvičení, které jsou součástí úvodního cvičení.

Úkol 1.1 Napište funkci, která pro zadané rozměry a hustotu kvádrů vrátí jeho objem, povrch a hmotnost. ▲

Úkol 1.2 Běžná cihla má rozměry 29 cm × 14 cm × 6,5 cm a hustotu $1,9 \frac{\text{g}}{\text{cm}^3}$. Jaká je její hmotnost? ▲

Úkol 1.3 Napište funkci, která pro zadaný poloměr a výšku válce vrátí jeho objem a povrch. Dále vypočtete objem a povrch válce o poloměru 10 cm a výšce 5 cm. ▲

Úkol 1.4 Napište funkci, která pro zadané koeficienty a, b, c a hodnoty $d_0 < d_1$ vykreslí graf funkce $f(x) = ax^2 + bx + c$ na intervalu (d_0, d_1) . ▲

Úkol 1.5 Napište funkci, která pro zadané koeficienty a, b, c vypíše kořeny kvadratické funkce $ax^2 + bx + c$. ▲

Úkol 1.6 Využijte předchozí úkoly a vytvořte funkci, která pro zadané koeficienty a, b, c a hodnoty $d_0 < d_1$ vykreslí graf funkce $f(x) = ax^2 + bx + c$ na intervalu (d_0, d_1) a zároveň znázorní kořeny funkce f . ▲

CVIČENÍ 2: NUMERICKÝ VÝPOČET URČITÉHO INTEGRÁLU A MONTE CARLO

Funkci $F: \mathbb{R} \rightarrow \mathbb{R}$ nazveme **primitivní funkcí** k funkci $f: \mathbb{R} \rightarrow \mathbb{R}$ na otevřeném intervalu $I \subset \mathbb{R}$, pokud pro každé $x \in I$ platí

$$F'(x) = f(x).$$

Lze ukázat, že pokud je f spojitá na otevřeném intervalu I , pak existuje primitivní funkce F k f na tomto intervalu.

Pomocí primitivní funkce zadefinujeme určitý integrál funkce f od a do b , kde $a, b \in \mathbb{R}$ a $a < b$. Jsou-li funkce f a F spojité na uzavřeném intervalu $\langle a, b \rangle$ a je-li F primitivní k f na (a, b) , definujeme **určitý integrál funkce f od a do b** jako

$$\int_a^b f(x) dx := F(b) - F(a). \quad (1)$$

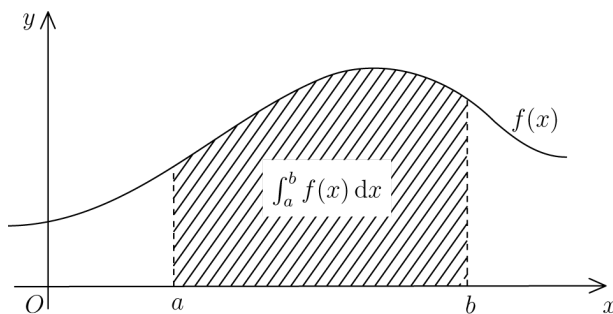
Vztah (1) se často nazývá **Newtonův-Leibnizův vzorec**^{1,2}. Pro zjednodušení zápisu se pro rozdíl hodnot $F(b)$ a $F(a)$ zavádí toto značení:

$$[F(x)]_a^b \stackrel{\text{ozn.}}{=} F(b) - F(a).$$

Popišme si nyní geometrický význam určitého integrálu. Nejprve předpokládejme, že je funkce f spojitá a nezáporná na intervalu $\langle a, b \rangle$. Pak integrál

$$\int_a^b f(x) dx$$

je roven obsahu rovinného obrazce ohraničeného osou x , grafem funkce f a přímkami $x = a$ a $x = b$, viz obr. 1. Nyní rozšířme naše úvahy na spojitě funkce, které mohou být v intervalu



Obrázek 1: Význam určitého integrálu

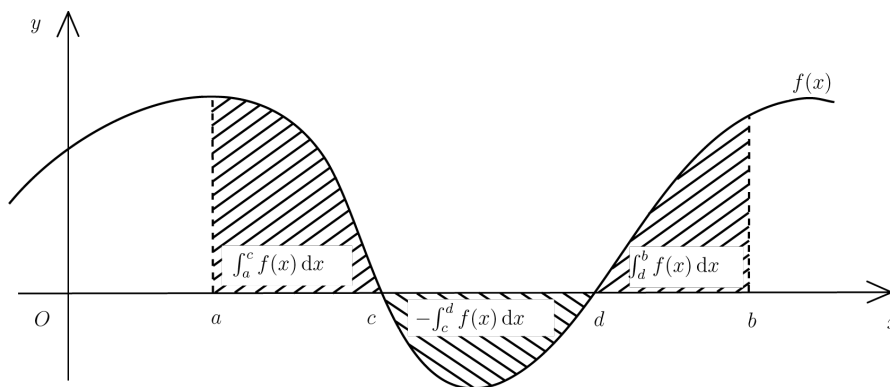
$\langle a, b \rangle$ záporné. Nechť je např. funkce f záporná na intervalu $(c, d) \subset \langle a, b \rangle$, viz obr. 2. Potom také

$$\int_c^d f(x) dx < 0.$$

¹Isaac Newton (1643–1727) – významný anglický fyzik, matematik a astronom

²Gottfried Wilhelm Leibniz (1646–1716) – významný německý matematik

Chceme-li pak pomocí tohoto integrálu vypočítat obsah plochy ohraničené osou x , grafem funkce f a přímkami $x = c$ a $x = d$, musíme na této části vzít integrál s opačným znaménkem. Bude-li nás dále zajímat obsah plochy, kterou ohraničuje osa x , graf funkce f a přímky $x = a$ a $x = b$, a bude-li funkce f protínat v intervalu $\langle a, b \rangle$ osu x , musíme tyto průsečíky najít a rozdělit interval $\langle a, b \rangle$ na intervaly, na nichž má f totéž znaménko. Na těchto intervalech spočteme integrály funkce f . Výsledný obsah plochy pak dostaneme jako součet všech vypočtených integrálů, přičemž integrály na těch intervalech, kde je f nekladná, musíme uvažovat se záporným znaménkem (viz obr. 2).



Obrázek 2: Obsah rovinného obrazce a určitý integrál

Přibližný výpočet určitého integrálu

V případě, že primitivní funkci nelze vyjádřit elementárními nebo tabelovanými funkcemi, Newtonův-Leibnizův vzorec (1) nemůžeme použít. Někdy zase může být hledání primitivní funkce příliš složité či časově náročné. V těchto případech často přistupujeme k tzv. numerickému výpočtu daného určitého integrálu, který nám dá přibližnou hodnotu integrálu s danou přesností. V následujícím textu si přiblížíme dvě základní metody přibližného výpočtu určitého integrálu, a to obdélníkové a lichoběžníkové pravidlo.

Obdélníkové pravidlo

Rozdělme nejprve interval $\langle a, b \rangle$ na n stejných dílků o délce

$$h = \frac{b - a}{n}. \quad (2)$$

Krajní body i -tého dílku postupně označme x_{i-1} a x_i ; platí tedy

$$x_0 = a < x_1 < \dots < x_{n-1} < x_n = b. \quad (3)$$

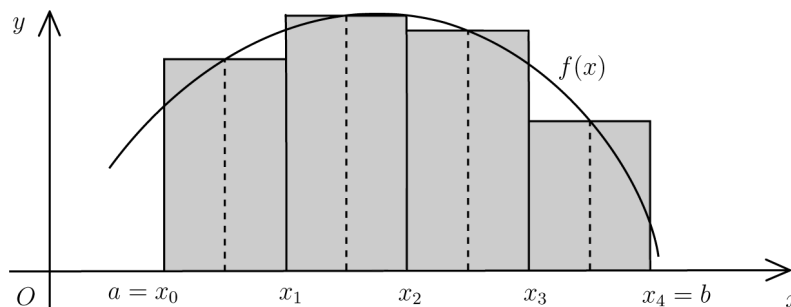
Vypočteme si dále středy jednotlivých dílků:

$$c_i = \frac{x_{i-1} + x_i}{2}, \quad i = 1, 2, \dots, n.$$

Na i -tém dílku pak funkci f nahradíme konstantní funkcí o hodnotě $f(c_i)$ a hledaný integrál budeme aproximovat takto:

$$\int_a^b f(x) dx \approx hf(c_1) + hf(c_2) + \dots + hf(c_n) = h \sum_{i=1}^n f(c_i).$$

Například integrál funkce f od a do b z obr. 3 tak nahrazujeme součtem obsahů příslušných obdélníků.



Obrázek 3: Aproximace obdélníky ($n = 4$)

Lze ukázat, že pokud existuje spojitá f'' na $\langle a, b \rangle$, potom pro chybu aproximace platí

$$\left| \int_a^b f(x) dx - h \sum_{i=1}^n f(c_i) \right| \leq \frac{(b-a)^3}{24n^2} \max_{x \in \langle a, b \rangle} |f''(x)|. \quad (4)$$

Lichoběžníkové pravidlo

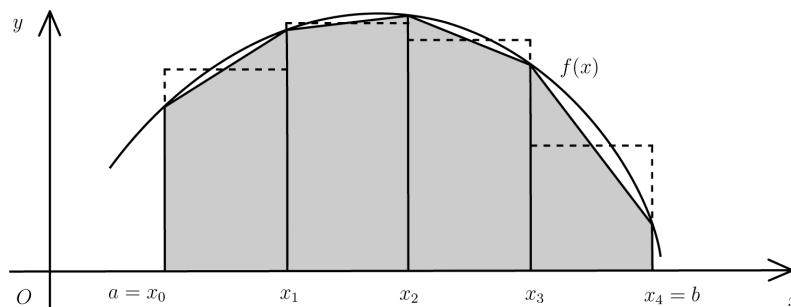
Interval $\langle a, b \rangle$ rozdělíme stejně jako u obdélníkového pravidla na dílky shodné délky, viz (2) a (3). Hledaný integrál budeme aproximovat takto:

$$\begin{aligned} \int_a^b f(x) dx &\approx h \frac{f(x_0) + f(x_1)}{2} + h \frac{f(x_1) + f(x_2)}{2} + \dots + h \frac{f(x_{n-1}) + f(x_n)}{2} = \\ &= \frac{h}{2} (f(x_0) + 2f(x_1) + \dots + 2f(x_{n-1}) + f(x_n)) = \frac{h}{2} \left(f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i) \right). \end{aligned}$$

Například integrál funkce f od a do b z obr. 4 tak nahrazujeme součtem obsahů příslušných lichoběžníků.

Je možné ukázat, že pokud existuje spojitá f'' na $\langle a, b \rangle$, potom pro chybu aproximace platí

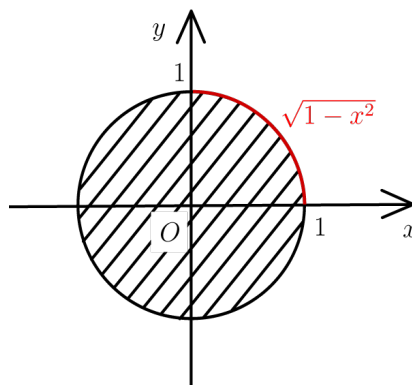
$$\left| \int_a^b f(x) dx - \frac{h}{2} \left(f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i) \right) \right| \leq \frac{(b-a)^3}{12n^2} \max_{x \in \langle a, b \rangle} |f''(x)|.$$



Obrázek 4: Aproximace lichoběžníky ($n = 4$)

Všimněme si, že u lichoběžníkového pravidla máme odhad chyby aproximace horší než u obdélníkového pravidla, přestože u obdélníkového pravidla užíváme nahrazení konstantními funkcemi a u lichoběžníkového pravidla používáme nahrazení lineárními funkcemi.³ Z uvedených vzorců pro odhad chyby aproximace lze určit, jaký počet dílků zaručí požadovanou přesnost aproximace. Jelikož ovšem odhady obsahují druhé derivace, jejichž hodnoty nemusí být lehké odhadnout, může být výpočet počtu potřebných dílků náročný a výsledek pesimistický.

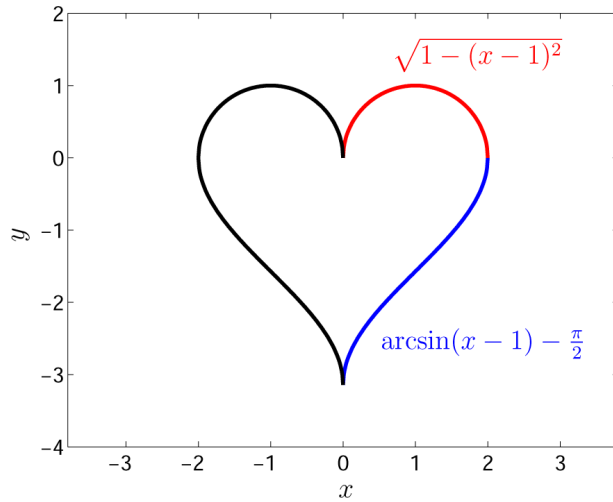
Úkol 2.1 Pomocí obdélníkového pravidla spočtete přibližně obsah jednotkového kruhu a výsledky porovnejte se skutečným obsahem daného kruhu.



Obrázek 5: Jednotkový kruh

Úkol 2.2 Pomocí obdélníkového pravidla spočtete přibližně obsah srdce na obr. 6 a výsledky porovnejte se skutečným obsahem srdce.

³Aproximační vlastnosti obdélníkového a lichoběžníkového pravidla si lze detailněji prostudovat např. v [2].



Obrázek 6: Matematické srdce

Úkol 2.3 Aproximujte hodnotu integrálu

$$\int_0^1 x^2 dx$$

pomocí obdélníkového pravidla tak, aby chyba aproximace byla nejvýše 10^{-4} . Určete počet dílků dělení intervalu $\langle 0, 1 \rangle$, který zaručí dosažení požadované přesnosti.

Monte Carlo

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. Pro odhad přesnosti metody Monte Carlo platí následující tvrzení. S pravděpodobností 75 procent nemá chyba aproximace hodnotu větší než $1/\sqrt{n}$. Tzn. například pro 10^4 pokusů nám s pravděpodobností 75 procent chyba nepřekročí hodnotu 0,01.

My si ukážeme tři způsoby, jak pomocí Monte Carla spočítat určitý integrál. První dvě metody jsou si typově podobné a spočívají v určení poměru obsahů neznámé plochy a plochy, jejíž obsah známe.

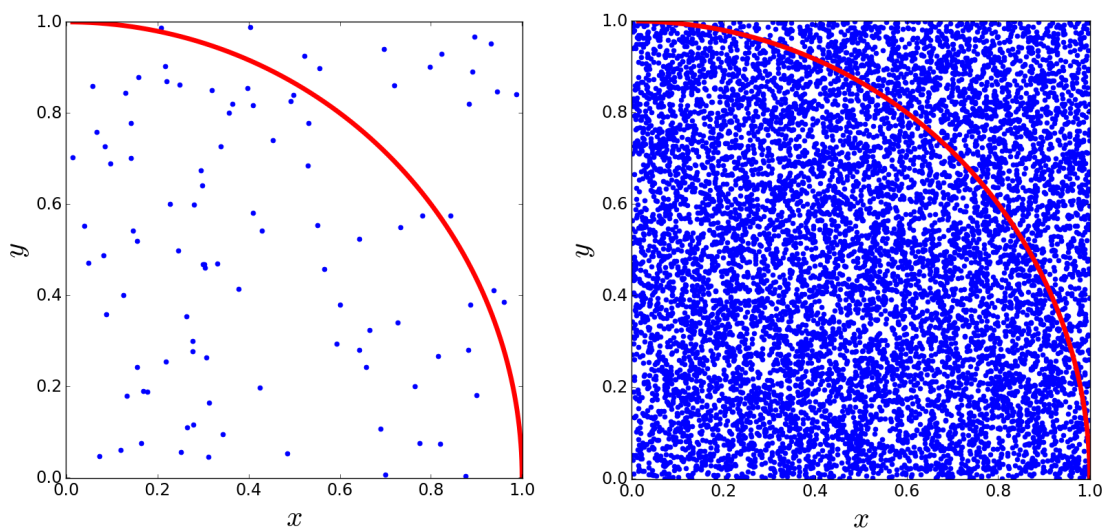
Náhodné souřadnice

Nejjednodušší způsob, jak pomocí Monte Carla odhadnout určitý integrál, je pomocí náhodného generování souřadnic v rovině. Princip si můžeme ukázat na problému určení integrálu

$$\int_0^1 \sqrt{1-x^2} dx, \quad (5)$$

který odpovídá spočítání obsahu plochy vymezené grafem funkce $\sqrt{1-x^2}$, osou x a osou y .

Grafem integrované funkce je na intervalu $\langle 0, 1 \rangle$ čtvrtina kružnice (viz obr. 7). Označme písmenem S plochu mezi tímto grafem, osou x a osou y . Nejprve vepíšeme plochu S do obrazce, jehož obsah známe, třeba do jednotkového čtverce. Metoda Monte Carlo funguje tak, že v jednotkovém čtverci náhodně generuje body. Pravděpodobnost, že vygenerovaný bod leží v S , je rovna poměru obsahu plochy S k obsahu plochy jednotkového čtverce, do kterého jsme ji uzavřeli.



Obrázek 7: Odhad obsahu plochy pod grafem funkce $\sqrt{1-x^2}$ (zakreslena červeně) na intervalu $\langle 0, 1 \rangle$ pomocí generování souřadnic v rovině. Pokrytí plochy jednotkového čtverce pomocí 100 (vlevo) a 10 000 (vpravo) náhodně generovaných bodů (zakresleny modře).

Úkol 2.4 Odhadněte číslo π pomocí metody Monte Carlo. Využijte k tomu integrál

$$\int_0^1 \sqrt{1-x^2} dx,$$

jehož odhad nám dá přibližnou hodnotu $\pi/4$. ▲

Tento přístup je intuitivní a kromě simulace v počítači si jej lze vyzkoušet i v praxi, například budete-li mít dlouhou chvíli na písčné pláži. Generování bodů není nic jiného, než házení kamínků (náhodnou silou a náhodným směrem) do jednotkového čtverce, ve kterém máme nakreslený graf funkce, jejíž integrál chceme odhadnout. Poměr počtu kamínků, které spadly pod graf funkce, k celkovému počtu kamínků ve čtverci nám dá odhad daného integrálu.

Náhodná procházka

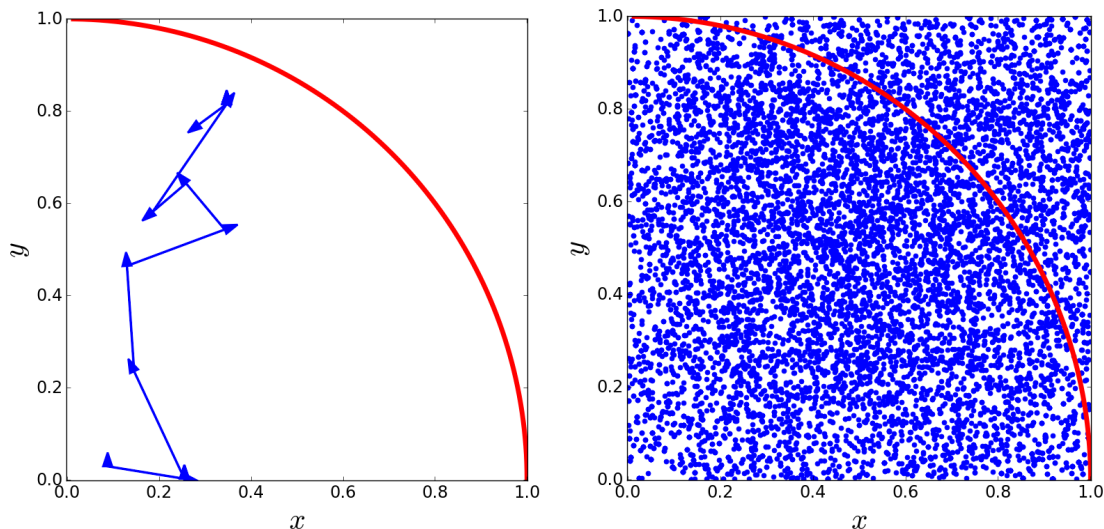
Zůstaňme na chvíli u představy simulace pomocí házení kamínků do písku a představme si, že se chce do simulace zapojit i malé dítě, které ještě nedohodí dost daleko. Pro jednoduchost uvažujme, že dítě může házet kamenem jen ve čtverci o délce strany $\delta < 1$. Pokud bychom pokračovali podle předchozích pravidel, nebyly by pozice ve čtverci generovány zcela náhodně. Je tedy třeba nějak upravit algoritmus.

Jedna z možností je, že se první kámen hodí náhodně do jednotkového čtverce a poté se bude konstruovat náhodná procházka podle následujících pravidel:

- každý hod začíná v místě dopadu předchozího hodu,
- hází se náhodně ve čtverci o délce strany δ , jehož střed je v místě, odkud se zrovna hází,
- pokud by nějaký hod mířil ven z jednotkového čtverce, kamínek se položí na předchozí kámen a pokračuje se ze stejného místa (jinými slovy – hod se započítá, ale pozice zůstane stejná).

Tento způsob nám zaručí, že po čase budou kameny ve čtverci rozmístěny opravdu náhodně. Pak už můžeme opět pomocí poměru počtu kamínků pod grafem k celkovému počtu kamínků ve čtverci odhadnout hledaný obsah. Schéma náhodné procházky je na obr. 8.

Tento způsob je mnohem složitější a méně přesný než předchozí, proto se využívá jen v případech, kdy není možné použít jiný přístup. Nejčastěji se využívá v případech, kdy nejsme schopni náhodně vybírat čísla s požadovanou hustotou pravděpodobnosti. V tomto textu se však s podobným případem nesetkáme – pokaždé volíme čísla s uniformní (rovnouměrnou) hustotou pravděpodobnosti.



Obrázek 8: Odhad obsahu plochy pod grafem funkce $\sqrt{1-x^2}$ (zakreslena červeně) na intervalu $\langle 0, 1 \rangle$ pomocí generování souřadnic náhodnou procházkou. Konstrukce náhodné procházky s $\delta = 0,25$ pomocí 15 (vlevo) a 10 000 (vpravo) bodů (zakresleny modře).

Úkol 2.5 Odhadněte číslo π pomocí metody Monte Carlo s generováním bodů pomocí náhodné procházky. Opět se nám bude hodit integrál

$$\int_0^1 \sqrt{1-x^2} dx,$$

jehož odhad nám dá přibližnou hodnotu $\pi/4$. ▲

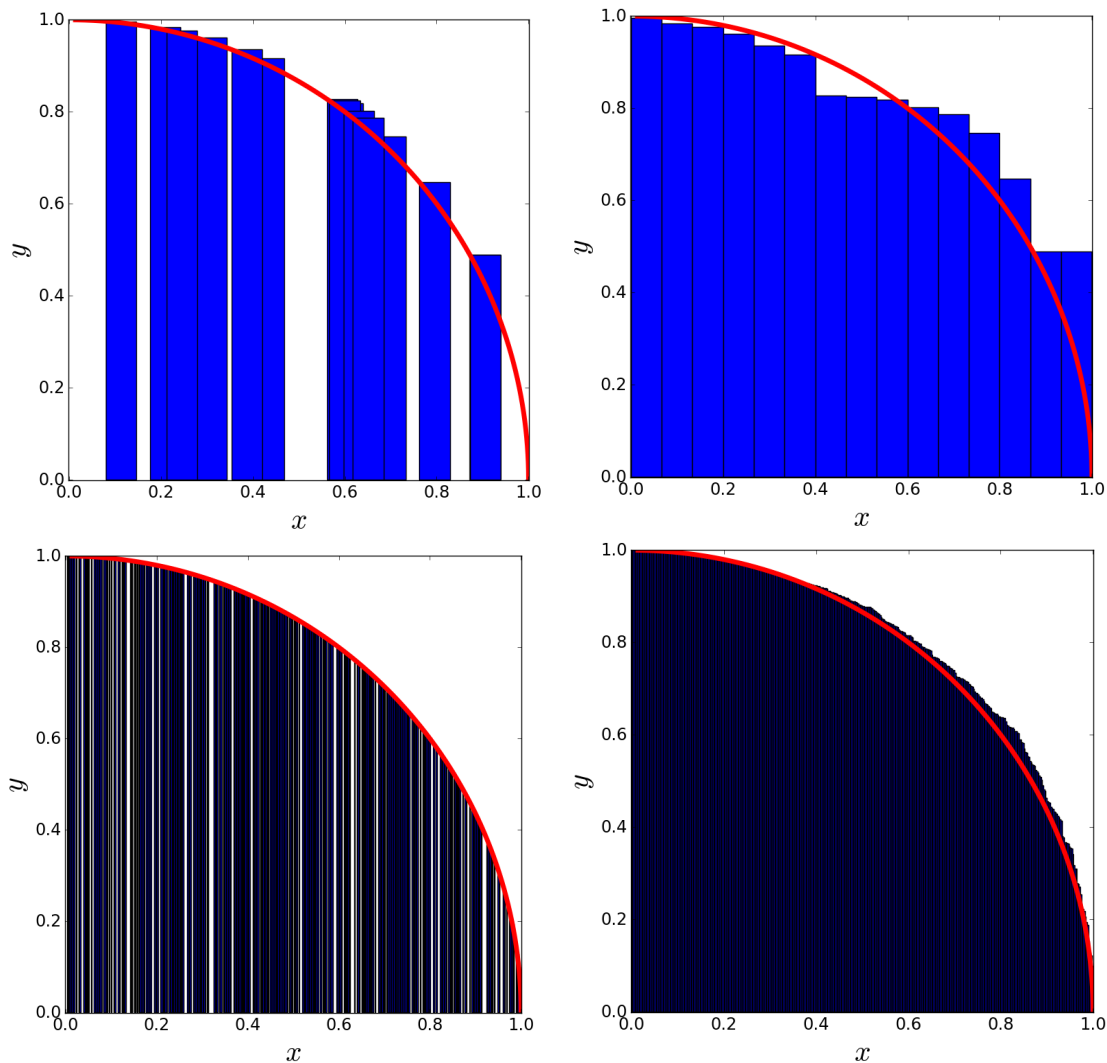
Crude Monte Carlo

Tato metoda je odvozena z definice určitého integrálu a využívá vzorec

$$\int_0^1 f(x) dx = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i), \quad (6)$$

kde body x_i , ve kterých se funkce vyčísluje, jsou generovány náhodně na intervalu $\langle 0, 1 \rangle$.

Na rozdíl od předchozích metod, které odhadovaly plochu pod grafem pomocí poměru bodů pod grafem k celkovému počtu bodů, tento přístup počítá odhad pomocí průměru funkčních hodnot v náhodných bodech. Nepotřebujeme tedy generovat body v rovině, ale pouze v jednom rozměru. Jak tato metoda funguje, můžete vidět na obr. 9.



Obrázek 9: Odhad obsahu plochy pod grafem funkce $\sqrt{1-x^2}$ na intervalu $\langle 0, 1 \rangle$ pomocí metody Crude Monte Carlo. Horní (modrá) řada odpovídá simulaci s $n = 15$ vyčísleními, spodní (černá) odpovídá simulaci s $n = 500$ vyčísleními. V levé části jsou vykresleny jednotlivé příspěvky tak, jak byly generovány v průběhu simulace. Vpravo jsou seřazeny tak, aby se nepřekrývaly pro snazší porovnání s odhadovanou plochou.

Úkol 2.6 Odhadněte číslo π pomocí metody Crude Monte Carlo. Využijeme integrál

$$\int_0^1 \sqrt{1-x^2} dx,$$

jehož odhad nám dá přibližnou hodnotu $\pi/4$. ▲

Vše najdete podrobně popsáno v Jupyter Notebook „cv2“ a „cv2.reseni“, který nalez-

nete na adrese

<https://mybinder.org/v2/gh/Beremi/SKOMAM/main>

a také jsme připravili video na YouTube provádějící tímto cvičením, které najdete v playlistu

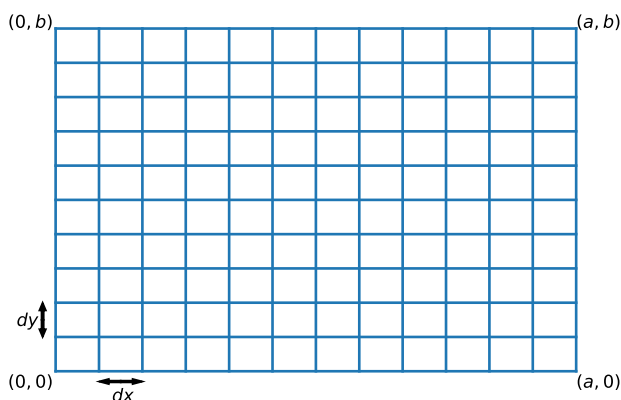
<https://youtube.com/playlist?list=PLVrbfTRvGpyU2u-4iSNoGh8tprKIo9hvx>.

CVIČENÍ 3: MODELOVÁNÍ PROCESU VEDENÍ TEPLA

Uvažujme oblast tvaru obdélníku o stranách a , b . V počátečním čase $t = 0$ má každý bod oblasti určitou teplotu. Úkolem tohoto cvičení je modelovat průběh vedení tepla v čase, tedy spočítat teplotu v oblasti po uplynutí určitého času. Teplotu budeme reprezentovat pomocí funkce

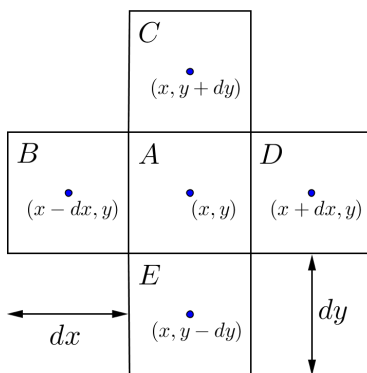
$$u : \mathbb{R}^3 \rightarrow \mathbb{R},$$

hodnota $u(x, y, t)$ udává teplotu v bodě o souřadnicích (x, y) v čase t .



Obrázek 10: Diskretizace obdélníkové oblasti

Nejprve provedeme tzv. diskretizaci - zadanou oblast rozdělíme na shodné obdélníkové podoblasti o stranách dx , dy , viz obr. 10. Pro jednoduchost nebudeme pracovat s hodnotami funkce u ve všech bodech zadané oblasti, ale teplotu v každé podoblasti budeme aproximovat pouze jednou hodnotou. Teplotu v podoblasti A (viz obr. 11) tedy reprezentujeme pomocí hodnoty v bodě (x, y) a podobně v sousedních podoblastech B , C , D , E .



Obrázek 11: Podoblasti B , C , D , E sousedící s podoblastí A

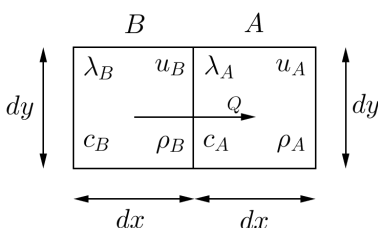
Aby bylo zadání kompletní, je potřeba specifikovat tzv. okrajové podmínky. Budeme předpokládat, že teplota na hranici oblasti se v čase nemění, tj. zůstává stejná jako v čase

$t = 0$. V řeči diskretizované úlohy budeme předpokládat, že teplota v podoblastech ležících na okraji oblasti zůstává stále stejná.

Proces vedení tepla si zkusíme rozmyslet nejprve z fyzikálního pohledu. Poté si ukážeme jednu z možností, jak k tomuto problému přistupovat z pohledu matematiky.

Fyzikální odvození vedení tepla (1. způsob)

Budeme předpokládat, že zadaná oblast tvaru obdélníku má i třetí rozměr - tloušťku h . Ve skutečnosti se tedy jedná o kvádr $a \times b \times h$. Vyberme dvě libovolné sousedící podoblasti a označme je A a B , viz obr. 12. Předpokládejme, že každá z podoblastí má jinou teplotu. Z fyziky umíme určit množství tepla, které během určitého času předá teplejší podoblast té chladnější. Toto předané teplo způsobí snížení teploty v teplejší podoblasti a zvýšení teploty v chladnější podoblasti, postupně tedy dochází k vyrovnávání teplot. Při odvozování



Obrázek 12: Šíření tepla Q z podoblasti B do podoblasti A

budeme používat následující značení:

dx [m], dy [m], dt [s]	diskretizační kroky
u [K]	teplota
λ [$W m^{-1} K^{-1}$]	součinitel tepelné vodivosti
ρ [$kg m^{-3}$]	hustota
c [$J kg^{-1} K^{-1}$]	měrná tepelná kapacita
m [kg]	hmotnost

Označíme u_A , λ_A , ρ_A , c_A , m_A veličiny v podoblasti A , podobně pro podoblast B . Základní jednotkou teploty je Kelvin. My však budeme pracovat pouze s teplotními rozdíly, můžeme proto bez obav používat stupně Celsia (obě stupnice jsou navzájem pouze posunuty o hodnotu 273,15).

Vyjděme z toho, že známe teplotu v obou oblastech v čase t , tedy hodnoty $u_A(t)$ a $u_B(t)$. Naším úkolem bude určit teploty po uplynutí času dt , neboli hodnoty $u_A(t + dt)$ a $u_B(t + dt)$. Nejprve určíme množství tepla Q , které předá podoblast B podoblasti A za časový krok dt . Pro toto předané teplo platí

$$Q = q dy h dt, \quad (7)$$

kde q je hustota tepelného toku a $dy h$ je plocha společná podoblastem A a B . Teplotu v podoblastech reprezentujeme pomocí hodnot ve středu těchto podoblastí, teplo mezi

oběma středy tedy prostupuje vzdálenost $\frac{dx}{2}$ přes materiál podoblasti A a $\frac{dx}{2}$ přes materiál oblasti B . Hustotu tepelného toku q určíme pomocí vzorce

$$q = \frac{u_B(t) - u_A(t)}{\frac{dx}{2} \left(\frac{1}{\lambda_A} + \frac{1}{\lambda_B} \right)}. \quad (8)$$

Zbývá určit, o kolik stupňů se zvýší teplota podoblasti A po předání tepla Q z podoblasti B , tento teplotní přírůstek označíme Δu_{BA} . Hodnota Δu_{BA} závisí na tepelné kapacitě C_A podoblasti A , platí

$$\Delta u_{BA} = \frac{Q}{C_A}, \quad (9)$$

kde

$$C_A = m_A c_A = dx dy h \rho_A c_A. \quad (10)$$

Dosazením vztahů (7), (10) a (8) do vzorce (9) získáváme

$$\Delta u_{BA} = \frac{q dy h dt}{dx dy h \rho_A c_A} = \frac{u_B(t) - u_A(t)}{\frac{dx}{2} \left(\frac{1}{\lambda_A} + \frac{1}{\lambda_B} \right)} \frac{dy h dt}{dx dy h \rho_A c_A} = \frac{2(u_B(t) - u_A(t)) dt}{\left(\frac{1}{\lambda_A} + \frac{1}{\lambda_B} \right) \rho_A c_A dx^2}.$$

Můžeme si všimnout, že výsledný vzorec neobsahuje tloušťku h , ta se nám hodila pouze při odvozování.

Stejným způsobem bychom vyjádřili hodnoty Δu_{CA} , Δu_{DA} a Δu_{EA} , tedy změny teploty podoblasti A vlivem kontaktu s podoblastmi C , D a E (viz obr. 11). Teplotu podoblasti A v čase $t + dt$ pak vyjádříme jako součet

$$u_A(t + dt) \approx u_A(t) + \Delta u_{BA} + \Delta u_{CA} + \Delta u_{DA} + \Delta u_{EA}. \quad (11)$$

Vyjádřili jsme tedy teplotu v podoblasti A v čase $t + dt$ pomocí teplot v podoblastech A , B , C , D , E v čase t . Oblasti A až E můžeme ztotožnit s uzly (x, y) až $(x, y - dy)$, vzorec (11) tedy můžeme převést do podoby

$$u(x, y, t + dt) \approx u(x, y, t) + \frac{2}{\rho(x, y) c(x, y)} dt \left(\frac{u(x - dx, y, t) - u(x, y, t)}{\left(\frac{1}{\lambda(x, y)} + \frac{1}{\lambda(x - dx, y)} \right) dx^2} + \right. \quad (12)$$

$$\left. \frac{u(x, y + dy, t) - u(x, y, t)}{\left(\frac{1}{\lambda(x, y)} + \frac{1}{\lambda(x, y + dy)} \right) dy^2} + \frac{u(x + dx, y, t) - u(x, y, t)}{\left(\frac{1}{\lambda(x, y)} + \frac{1}{\lambda(x + dx, y)} \right) dx^2} + \frac{u(x, y - dy, t) - u(x, y, t)}{\left(\frac{1}{\lambda(x, y)} + \frac{1}{\lambda(x, y - dy)} \right) dy^2} \right).$$

Odvození pomocí konečných diferencí (2. způsob)

V předchozí sekci jsme popsali model vedení tepla pomocí známých fyzikálních zákonů. Matematicky lze proces vedení tepla v čase reprezentovat pomocí parciální diferenciální rovnice

$$\frac{\partial u}{\partial t} = \frac{\lambda}{\rho c} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (13)$$

více o parciálních derivacích lze najít v příloze D.

Opět využijeme diskretizaci z obr. 10. Pro jednoduchost budeme předpokládat, že hodnoty veličin λ , c , q jsou shodné ve všech podoblastech, uvažujeme tedy homogenní materiál. Parciální derivace, které se vyskytují v rovnici (13), aproximujeme pomocí tzv. diferencí. Pro aproximaci derivace podle proměnné x volíme krok dx , pro proměnnou y krok dy a pro proměnnou t krok dt . Zjednodušeně můžeme říct, že „zanedbáme“ limitu v definici D.1 (viz str. 23):

$$\begin{aligned} \frac{\partial u}{\partial t}(x, y, t) &\approx \frac{u(x, y, t + dt) - u(x, y, t)}{dt}, \\ \frac{\partial u}{\partial x}(x, y, t) &\approx \frac{u(x + dx, y, t) - u(x, y, t)}{dx}, \\ \frac{\partial u}{\partial y}(x, y, t) &\approx \frac{u(x, y + dy, t) - u(x, y, t)}{dy}. \end{aligned}$$

Dále vyjádříme aproximaci druhé parciální derivací podle proměnné x

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x, y, t) &\approx \frac{\frac{\partial u}{\partial x}(x, y, t) - \frac{\partial u}{\partial x}(x - dx, y, t)}{dx} \approx \frac{\frac{u(x+dx, y, t) - u(x, y, t)}{dx} - \frac{u(x, y, t) - u(x-dx, y, t)}{dx}}{dx} = \\ &= \frac{u(x + dx, y, t) - u(x, y, t)}{dx^2} + \frac{u(x - dx, y, t) - u(x, y, t)}{dx^2} \end{aligned}$$

a analogicky podle proměnné y

$$\frac{\partial^2 u}{\partial y^2}(x, y, t) \approx \frac{u(x, y + dy, t) - u(x, y, t)}{dy^2} + \frac{u(x, y - dy, t) - u(x, y, t)}{dy^2}.$$

Dosazením do rovnice (13) získáváme přibližnou rovnost

$$\begin{aligned} \frac{u(x, y, t + dt) - u(x, y, t)}{dt} &\approx \frac{\lambda}{\rho c} \left(\frac{u(x + dx, y, t) - u(x, y, t)}{dx^2} + \right. \\ &\left. + \frac{u(x - dx, y, t) - u(x, y, t)}{dx^2} + \frac{u(x, y + dy, t) - u(x, y, t)}{dy^2} + \frac{u(x, y - dy, t) - u(x, y, t)}{dy^2} \right), \end{aligned}$$

z níž můžeme vyjádřit

$$u(x, y, t + dt) \approx u(x, y, t) + \frac{\lambda}{\rho c} dt \left(\frac{u(x - dx, y, t) - u(x, y, t)}{dx^2} + \right. \quad (14)$$

$$+ \frac{u(x+dx, y, t) - u(x, y, t)}{dx^2} + \frac{u(x, y-dy, t) - u(x, y, t)}{dy^2} + \frac{u(x, y+dy, t) - u(x, y, t)}{dy^2} \Bigg).$$

Vyjádřili jsme tedy teplotu v bodě (x, y) v čase $t + dt$ pomocí teplot v bodech (x, y) , $(x + dx, y)$, $(x - dx, y)$, $(x, y + dy)$ a $(x, y - dy)$ v čase t . Jinými slovy - teplotu v podoblasti A v čase $t + dt$ jsme opět vyjádřili pomocí teplot v podoblastech A, B, C, D, E v čase t . Můžeme si povšimnout, že pokud bychom při fyzikálním odvozování v předchozí sekci uvažovali homogenní materiál (tedy $\lambda_A = \lambda_B = \lambda$, $\rho_A = \rho$ a $c_A = c$), vzorec (12) by přešel do podoby (14).

Implementace

Oba výše uvedené přístupy vedou na stejný postup implementace. Vstupem programu jsou tři matice stejné velikosti získané diskretizací zadané oblasti:

- matice teplot v počátečním čase $t = 0$,
- matice součinitelů tepelné vodivosti λ ,
- matice hodnot ρc (hustota krát měrná tepelná kapacita),

dále velikosti diskretizačních kroků dx , dy , dt a celkový počet časových kroků N .

Stěžejní částí zdrojového kódu je implementace jednoho časového kroku. Matici teplot v čase $t = 0$ označíme U_0 . Matici teplot v čase $t = dt$ (označme U_1) určíme následovně:

1. Nastavíme $U_1 = U_0$.
2. V cyklu projdeme všechny prvky matice U_1 kromě okrajových.

Každý prvek odpovídá jedné podoblasti (označíme A). V matici U_0 najdeme hodnoty teploty ve čtyřech sousedních podoblastech v předchozím časovém kroce. Pomocí nich určíme hodnoty Δu_{BA} , Δu_{CA} , Δu_{DA} , Δu_{EA} a přičteme je k danému prvku matice U_1 .

Těchto časových kroků provedeme v cyklu celkem N . Výstupem je tedy matice teplot v síti po uplynutí N časových kroků. Případně je možné ukládat tuto matici jako obrázek vždy po uplynutí určitého počtu časových kroků a z výsledných obrázků vytvořit animaci.

Při implementaci začneme nejprve zjednodušenou situací, kdy budeme předpokládat homogenní materiál. V tomto případě tedy nemusíme zadávat λ a ρc v podobě matic, jedná se pouze o konstanty, viz následující zdrojový kód v jazyce Python. Další rozšíření tohoto základního kódu naleznete na adrese

<https://mybinder.org/v2/gh/Beremi/SKOMAM/main>

v podobě Jupyter Notebooku `cv3_teplo`. Také jsme připravili video na YouTube provádějící tímto cvičením, které najdete v playlistu

<https://youtube.com/playlist?list=PLVrbfTRvGpyU2u-4iSNoGh8tprKIo9hvx>.

```

import numpy as np
import matplotlib.pyplot as plt

dx = 0.01          # delka intervalu diskretizace ve vodorovnem smeru
dy = 0.01          # delka intervalu diskretizace ve svislem smeru
dt = 0.5           # velikost casoveho kroku
N = 240            # pocet casovych kroku
lam = 80           # soucinitel tepelne vodivosti
rho_c = 8000*450   # hustota krat merna tepelna kapacita

# Vytvorime matici teplot v pocatecnim case: ny ... pocet radku,
# nx ... pocet sloupcu. Prvky matice tedy budeme indexovat v poradí y,x.
nx = 80            # pocet intervalu diskretizace ve vodorovnem smeru
ny = 60            # pocet intervalu diskretizace ve svislem smeru
U0 = 20*np.ones((ny,nx)) # Pocatecni teplotu nastavime na 20 stupnu.
U0[2:32,30:60] = 70   # Do oblasti "vlozime" ctverec o vyssi teplotě
U0[35:50,15:30] = 2   # a ctverec o nizsi teplotě.

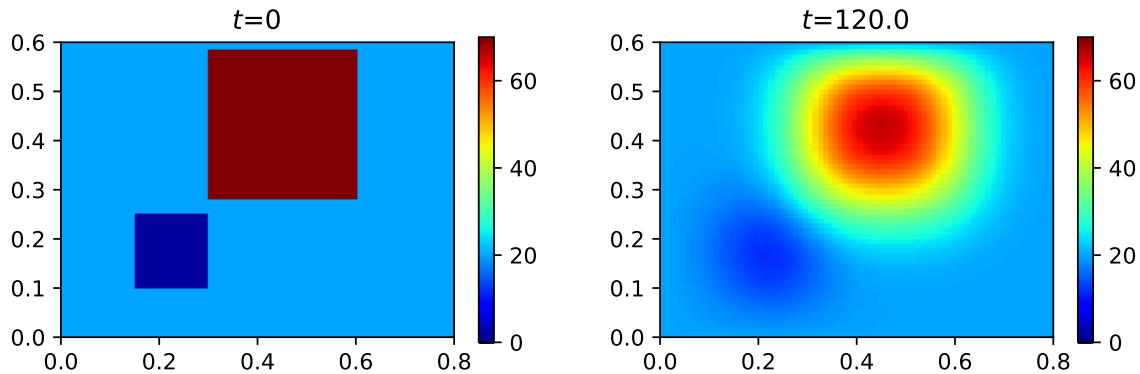
plt.figure()        # Vykreslime pocatecni teplotu.
plt.imshow(U0, cmap="jet", vmin=0, vmax=70, extent=(0,nx*dx,0,ny*dy))

const_dx = lam/rho_c*dt/(dx**2) # konstanta pro sireni ve smeru x
const_dy = lam/rho_c*dt/(dy**2) # konstanta pro sireni ve smeru y
for i in range(N):   # cyklus pres casove kroky
    U1 = U0.copy()   # Teplotu v novem casovem kroku nastavime
                    # na hodnoty z predchoziho casoveho kroku.
    for x in range(1,nx-1): # Projdeme vsechny vnitřni prvky matice U1,
        for y in range(1,ny-1): # teplotu na okraji nemeňime.
            u = U0[y,x] # teplota v bode (x,y) v predchozim casovem kroku
            # Teplotu v bode (x,y) upravime na zaklade hodnot v sousednich
            # 4 bodech. Vysledek uložíme do matice pro novy casovy krok.
            U1[y,x] += const_dx*(U0[y,x-1]-u)
            U1[y,x] += const_dx*(U0[y,x+1]-u)
            U1[y,x] += const_dy*(U0[y-1,x]-u)
            U1[y,x] += const_dy*(U0[y+1,x]-u)
    U0 = U1          # Pripravime matici U0 pro dalsi casovy krok.

plt.figure()        # Vykreslime teplotu v case N*dt.
plt.imshow(U0, cmap="jet", vmin=0, vmax=70, extent=(0,nx*dx,0,ny*dy))

```

Na následujícím obrázku si můžeme prohlédnout grafický výstup výše uvedeného algoritmu, tedy porovnání počátečního rozložení teploty a rozložení teploty po 2 minutách. Obdélníková deska má rozměry $0,8\text{ m} \times 0,6\text{ m}$ a je vyrobena z homogenního materiálu s parametry $\lambda = 80\text{ W m}^{-1}\text{K}^{-1}$, $\rho = 8000\text{ kg m}^{-3}$ a $c = 450\text{ J kg}^{-1}\text{K}^{-1}$. Obdélník jsme rozložili na čtvercové podoblasti o délce strany $dx = dy = 0,01\text{ m}$, přičemž výpočet jsme prováděli s časovým krokem $dt = 0,5\text{ s}$.



Obrázek 13: Počáteční rozložení teploty: červená část má teplotu $70\text{ }^{\circ}\text{C}$, tmavě modrá $2\text{ }^{\circ}\text{C}$ a světle modrá $20\text{ }^{\circ}\text{C}$ (vlevo) a stav po uplynutí 120 s (vpravo)

Reference

- [1] J. Kubias: Učíme se programovat v jazyce Python 3. [cit. 3.1.2022]. Dostupné z: <http://howto.py.cz/index.htm>
- [2] P. Vodstrčil, J. Bouchala: Drobná překvapení spojená s numerickou integrací. [cit. 3.1.2022]. Pokroky matematiky, fyziky a astronomie, svazek 55, číslo 4 (2010). Dostupné z: <http://dml.cz/dmlcz/141970>

Definice A.1 Bud' $f : \mathbb{R} \rightarrow \mathbb{R}$. Funkci f^{-1} , pro niž platí současně:

- i) definiční obor Df^{-1} funkce f^{-1} je roven oboru hodnot funkce f ,
- ii) pro každé $x \in Df^{-1}$ platí, že $f^{-1}(x) = y \Leftrightarrow f(y) = x$,

nazveme funkcí inverzní k funkci f .

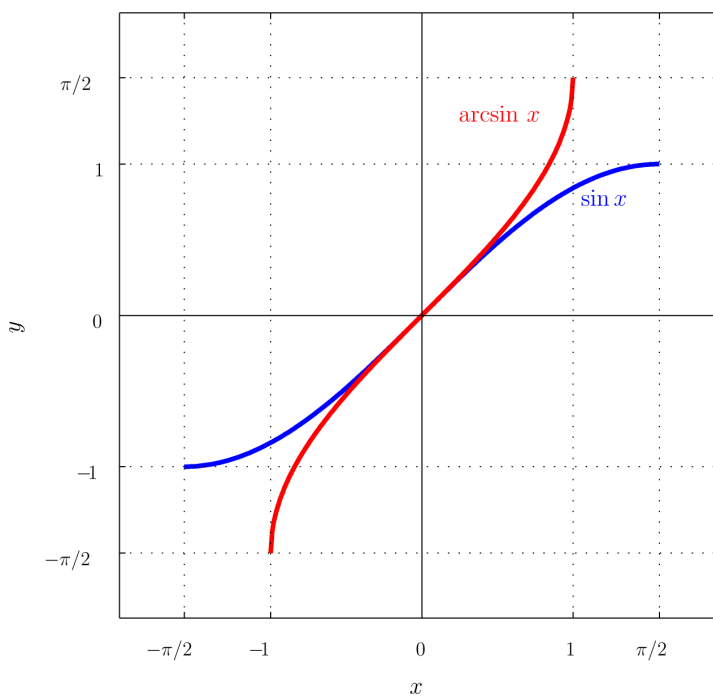
Poznámka A.1 Lze ukázat, že f^{-1} existuje právě tehdy, je-li f prostá. Graf f^{-1} je přitom osově souměrný s grafem f dle přímky $y = x$.

Definice B.1 Funkci inverzní k funkci sinus zúžené na interval $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$ nazveme arkussinus a označujeme jako \arcsin , tj.

$$\arcsin := \left(\sin|_{\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle} \right)^{-1}.$$

Pozorování B.1 Funkce arkussinus má tyto vlastnosti (viz také níže uvedený obrázek):

- definiční obor je roven $\langle -1, 1 \rangle$,
- oborem hodnot je interval $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$,
- funkce \arcsin je lichá, tj. $\arcsin x = \arcsin(-x)$ pro každé $x \in \langle -1, 1 \rangle$.



Definice C.1 Bud' $f : \mathbb{R} \rightarrow \mathbb{R}$ a $x \in \mathbb{R}$. Existuje-li

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h},$$

značíme ji $f'(x)$ a nazýváme derivací funkce f v bodě x .

Poznámka C.1 Většinou – a nejinak je to v tomto textu – se pod pojmem derivace rozumí konečná (tzv. vlastní) derivace.

Věta C.1 Bud' $f, g : \mathbb{R} \rightarrow \mathbb{R}$ a $x \in \mathbb{R}$. Pak platí

- $(f \pm g)'(x) = f'(x) \pm g'(x)$, má-li pravá strana rovnosti smysl,
- $(fg)'(x) = f'(x)g(x) + f(x)g'(x)$, existují-li (vlastní) derivace $f'(x)$ a $g'(x)$,
- $\left(\frac{f}{g}\right)'(x) = \frac{f'(x)g(x) - f(x)g'(x)}{g^2(x)}$, existují-li (vlastní) derivace $f'(x)$ a $g'(x)$ a je-li $g(x) \neq 0$.

Pozorování C.1

- $(c)' = 0$, $c \in \mathbb{R}$ (konst.), $x \in \mathbb{R}$,
- $(x^r)' = rx^{r-1}$, $r \in \mathbb{R}$, $x \in (0, +\infty)$,
- $(\sin x)' = \cos x$, $x \in \mathbb{R}$,
- $(\cos x)' = -\sin x$, $x \in \mathbb{R}$,
- $(\operatorname{tg} x)' = \frac{1}{\cos^2 x}$, $x \in \mathbb{R} \setminus \left\{\frac{\pi}{2} + k\pi : k \in \mathbb{Z}\right\}$,
- $(\operatorname{cotg} x)' = -\frac{1}{\sin^2 x}$, $x \in \mathbb{R} \setminus \{k\pi : k \in \mathbb{Z}\}$,
- $(e^x)' = e^x$, $x \in \mathbb{R}$,
- $(\ln x)' = \frac{1}{x}$, $x \in (0, +\infty)$.

Definice D.1 Necht' funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ je definována na nějakém okolí bodu

$$c = (c_1, c_2, \dots, c_n) \in \mathbb{R}^n.$$

Existuje-li konečná

$$\lim_{h \rightarrow 0} \frac{f(c_1, \dots, c_{i-1}, c_i + h, c_{i+1}, \dots, c_n) - f(c)}{h},$$

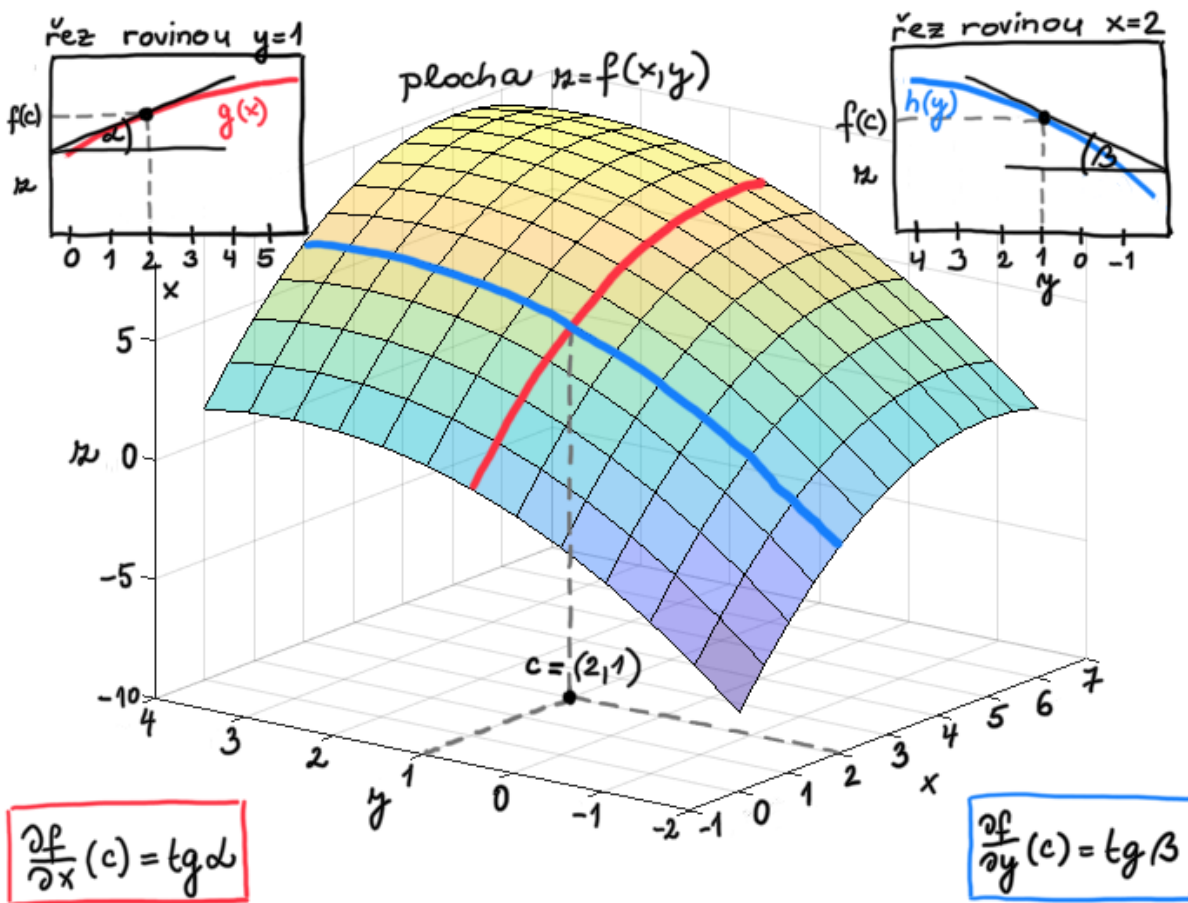
značíme ji $f_{x_i}(c)$ nebo $\frac{\partial f}{\partial x_i}(c)$ a nazýváme parciální derivací funkce f podle i -té proměnné v bodě c .

Poznámka D.2 Parciální derivace funkce f podle i -té proměnné x v bodě c je tedy rovna klasické derivaci funkce jedné proměnné

$$g(t) = f(c_1, \dots, c_{i-1}, t, c_{i+1}, \dots, c_n)$$

v bodě $t = c_i$.

Poznámka D.3 Budeme-li uvažovat $n = 2$, tak funkce f přiřazuje danému bodu (x, y) hodnotu $f(x, y)$ (tj. „výšku“ bodu $(x, y, f(x, y))$ nad rovinou xy). Grafem takovéto funkce je tedy plocha $z = f(x, y)$ definovaná na nějaké části roviny xy . Uvažujme na chvíli, že tato plocha je „hladká“ a definovaná na celé rovině xy a zamysleme se, jaký význam má parciální derivace funkce f podle proměnné x v bodě $c = (c_1, c_2)$. Provedeme-li řez dané plochy rovinou $y = c_2$ (tj. rovinou procházející bodem c , rovnoběžnou s osou x a kolmou k rovině xy), dostáváme křivku, kterou můžeme popsat funkcí $z = g(x) = f(x, c_2)$. Parciální derivace funkce f podle proměnné x v bodě c pak není nic jiného než směrnice tečny funkce g sestrojené v dotykovém bodě $(c_1, g(c_1))$. K pochopení může pomoci i následující obrázek.



Obrázek 14: Ilustrace významu parciálních derivací funkce f dvou proměnných x a y v bodě c .