**STUDENTSKÝ ČASOPIS A KORESPONDENČNÍ SEMINÁŘ**

Ročník XXVI
Číslo 4

# M&M

**MATEMATIKA FYZIKA INFORMATIKA**


Uvnitř najdete několik témat a s nimi souvisejících úloh. Zamyslete se nad nimi a pošlete nám svá řešení. My vám je opravíme, pošleme zpět s dalším číslem a ta nejzajímavější z nich otiskneme. Nejlepší řešitele zveme na podzim a na jaře na soustředění.

## Milí řešitelé,

právě držíte v rukou nejen čtvrté číslo tohoto ročníku, ale také první číslo nového roku. Doufáme, že jste do něj vkročili správnou nohou a bude pro vás nejméně tak úspěšný, jako byl rok předchozí.

A co vás čeká na následujících stránkách? V herním tématku tentokrát využijete nabyté znalosti o bimaticových hrách při řešení problémů evoluční biologie. V tématku o výpočetních modelech zjistíte, co je to nedeterminismus a jak vám náhoda může pomoci při řešení informatických úloh. Na stránkách třetího tématka rozšíříte data z radaru o další dimenzi a dále si můžete přečíst autorské řešení problémů z minulých čísel. V tématku o elektromagnetismu se seznámíte s Faradayovým zákonem elektromagnetické indukce. Určitě také nepřehlédněte hezký článek Bc.$^{\mathsf{MM}}$ Viléma Starosty o alternativních metodách pro hledání optimálních strategií.

Ačkoliv je teprve začátek února, přípravy na jarní soustředění, které proběhne v termínu 18.–26. dubna, jsou v plném proudu. Pokud se vám zatím nepodařilo získat dostatek bodů, nezoufejte, stále máte šanci se na něj dostat! Pokud nám pošlete svá řešení do 5. března, úlohy vám opravíme a body započítáme. Řešení můžete posílat i po tomto termínu, až do 26. března, avšak body z nich získané vám už k účasti na jarním soustředění bohužel započítat nestihneme.

Pokud se vám nechce čekat do dubna a chtěli byste své kamarády ze soustředění a některé organizátory potkat dříve, máte možnost se zúčastnit další zajímavé týmové soutěže, na kterou v současnosti běží přihlašování, a to Náboje[1].

*Vaši organizátoři*

# Zadání a řešení témat

**Termín odeslání úloh 4. série: 26. 3. 2020**
**(5. 3. 2020 pro účast na jarní soustředění)**

## Téma 1 – Hry

### Díl 4: Teorie her v biologii

Nyní bychom vám chtěli ukázat, že teorie her najde využití i jinde než v informatice a ekonomii. Ukážeme si, jak uplatnit, co víme o bimaticových hrách, při studiu evoluční biologie.

Představme si následující konflikt. Dva ptáčci se sletí nad jedním kouskem potravy a oba na něj mají chuť. Můžou o jídlo bojovat, můžou ho nechat být a odletět, ale taky se mohou o potravu rozdělit.

Zkusme si pomocí čísel vyjádřit, jak výsledek interakce ovlivní pravděpodobnost na přežití daného jedince, a tudíž i na jeho rozmnožení.

[1] `https://math.naboj.org/`

Nechť potrava má hodnotu 6 a zranění ze souboje má hodnotu $-18$. Z toho plyne, že rozdělení se o potravu dá každému 3 body. Souboj končí výsledkem $-6$, což je průměr hodnot $-18$ a 6. Pro jednoduchost zde totiž uvažujeme, že vítěz si odnáší pouze potravu a poražený si odnáší pouze zranění. Zároveň předpokládáme, že není možné předem odhadnout, kdo v souboji vyhraje.

Z pohledu biologie by se dalo zjednodušeně říct, že chování je určeno především genetickou výbavou jedince. Jednotlivé varianty genů se nazývají alely. Pro jednoduchost předpokládejme, že existuje jedna alela pro agresivní chování (bojovat o potravu) a jedna alela pro mírumilovné chování (chtít se o potravu rozdělit a v případě souboje hned utéct). Každý jedinec má jednu z těchto alel. Jedinci, kteří během obstarávání potravy získají více bodů, mají větší šanci na přežití a tedy i na rozmnožení a přenesení svých alel do dalších generací.

Uvažme strategie hrdličky (H) a jestřába (J). Na první pohled je nerozeznáme. Kdo je kdo se ukáže, teprve když dojde na lámání zobáků. V takovém případě hrdlička rychle uteče, zatímco jestřáb se pere za každou cenu. Pokud se potkají dvě hrdličky, poklidně se o potravu rozdělí. Upozorňujeme, že se nejedná o dva různé živočišné druhy, ale pouze o názvy strategií používaných jedinci stejného druhu.

Jedná se o symetrickou hru s nenulovým součtem. Dáme to teď do bimatice:

| | H | J |
|---|---|---|
| H | $3 \setminus 3$ | $0 \setminus 6$ |
| J | $6 \setminus 0$ | $-6 \setminus -6$ |

Co je optimální strategie? Lze snadno nahlédnout, že optimum v ryzích strategiích nenajdeme. Jaká je optimální smíšená strategie? Pokud poměr hrdliček v populaci označíme $z$, pak (kvůli symetrii zde $z$ označuje zároveň pravděpodobnost volby prvního řádku i pravděpodobnost volby prvního sloupce) průměrný zisk strategie H je $3z$, kdežto průměrný zisk strategie J je $6z + (-6)(1-z) = 12z - 6$. Rovnováha nastane, když $z = \frac{2}{3}$, neboť v tuto chvíli je průměrný zisk hrdliček i jestřábů 2. Jakmile se složení populace vychýlí od tohoto bodu, „nedostatkové variantě“ se začne dařit lépe, zplodí více potomků a poměr strategií se navrátí do rovnovážného stavu (2:1).

Všimněte si, že individuální zájmy jsou i zde v rozporu se skupinovými zájmy. Populaci jako celku by se nejlépe dařilo, kdyby všichni používali strategii H. Pak by zisk každého byl 3 body namísto zmíněných 2 bodů. Bohužel v evoluci zvítězí zájmy jedince nad zájmy skupiny, neboť, jakmile by do této zcela mírumilovné populace proniknul predátor (vyskytla by se mutace měnící chování na strategii J), začal by získávat 6 bodů a příslušná alela by se brzy exponenciálně začala rozšiřovat v populaci – přibližně až do dosažení zmíněné rovnováhy, kde jestřábi budou tvořit třetinu populace.

Minule jsme mluvili o vězňově dilematu a jeho rozšířeních. I tuto hru můžeme najít v biologii.

Představte si ptáčky, kteří občas mají parazity v opeření. Pokud se parazit nachází třeba na bříšku, dokáže si ho ptáček vyzobnout svým vlastním zobáčkem. Ale pokud má parazita třeba na vršku hlavičky, nedokáže se ho sám zbavit. Nyní by potřeboval, aby jiný ptáček přišel a tohoto parazita vyzobnul. Má však náš hrdina nějaký způsob, jak přesvědčit jiného ptáčka, aby mu pomohl?

Je celkem jisté, že kdyby si oba pomáhali navzájem, spolupráce by se jim vyplatila. Ztráta času při vyzobávání parazitů z cizího opeření má určitě menší evoluční cenu než zhoršení zdraví kvůli parazitům. Může se ale spolupráce nějak přirozeně vyvinout, když jsou jedinci či spíše jejich geny sobecké[2]?

První možnost je pokrevní příbuznost. Podle Hamiltonovy evoluční teorie[2] je výhodné pomáhat ostatním rodinným příslušníkům. Například bratr našeho hrdiny v sobě nosí polovinu genů, které má i náš hrdina, takže zvýšením bratrovy šance na přežití pomáhá šířit i své geny. Dle Hamiltona totiž nakonec nezáleží na tom, kolik potomků zplodí náš hrdina, ale na tom, kolik kopií alely způsobující sklony k vyzobávání bratrových parazitů z jeho opeření je přeneseno do dalších generací (nehledě na to, zda se daná alela přenese do další generace prostřednictvím jeho vlastních dětí nebo prostřednictvím bratrových dětí). Tudíž, je-li velikost jeho ztráty alespoň dvakrát menší než bratrův zisk, spolupráce se stane pro tuto alelu výhodná (neboť ji v 50 % případů nese i jeho bratr), takže se tato alela úspěšně rozšíří v populaci. Všimněte si, že zde nebyl potřeba jakýkoliv příslib vzájemné pomoci – té alele se vyplatí přesvědčit našeho hrdinu k „nezištné“ pomoci jeho bratrovi.

Nyní se zaměřme na ten záludnější případ, kde ti ptáčci nejsou pokrevně příbuzní. Co teď? Může se stát, že spolupráce vznikne sama od sebe na základě potenciální možnosti vzájemné pomoci?

Situace je již poměrně komplikovaná. Zkusíme si vzít na pomoc teorii her. Vzpomeneme si na vězňovo dilema, které lze popsat maticí níže:

| | C | D |
|---|---|---|
| C | 3 \ 3 | 0 \ 5 |
| D | 5 \ 0 | 1 \ 1 |

Tato matice vlastně docela dobře popisuje i problém našich ptáčků. Spolupráce (C) odpovídá vyzobávání parazitů. Zrada (D) odpovídá ignoranci druhého ptáčka.

Nyní už ptáčkova genetická výbava nemusí určovat volbu jedné konkrétní strategie, ale může určovat složitější rozhodovací mechanismus na to, jakou strategii volit v jaké situaci. Pokud by ptáčci neměli paměť na chování ostatních ptáčků, a tedy by soupeřova (či spíše kolegova) volba strategie nezáleželа na našem předchozím chování, vyplatilo by se vždy volit strategii D.

[2] https://en.wikipedia.org/wiki/Gene-centered_view_of_evolution

Většina ptáčků ale není tak hloupá. Buďme si však vědomi, že fakt, že si ptáček dokáže zapamatovat dřívější interakce se svými kolegy, ještě neznamená, že tento ptáček bude mít přirozenou tendenci se podle toho chovat. Je docela reálné, byť si ptáček bude pamatovat, kdo mu v minulosti pomáhal, že bude mít pouze tendenci vyhledávat v budoucnu interakci s těmito jedinci, aniž by těmto ptáčkům nazpět také pomáhal. Občas se však může vyskytnout mutace, která způsobí například osvojení postupu tzv. TitForTat:

„Když potkám nového kolegu, zkusím nejprve spolupracovat (C). Při dalších setkáních se k němu budu chovat tak, jak se tento kolega choval minule ke mně.“

Můžeme si všimnout, že postup TitForTat má spoustu dobrých vlastností. Když potkáme ptáčka, který je přátelský ke všem, kdo ho doposud nezradili (ze začátku volí strategii C, a jakmile soupeř použije D, navždy volí strategii D), budeme neustále získávat 3 body. Když potkáme ptáčka, který kašle na všechny (stále volí strategii D), budeme dlouhodobě získávat 1 bod, což je nejlepší možný výsledek proti tomuto ptáčkovi. Když potkáme ptáčka, který stejně jako my používá postup TitForTat, budeme taktéž neustále získávat 3 body, což je nejlepší možný dlouhodobě udržitelný výsledek.

Model samozřejmě není přesný. Interakce neprobíhá úplně zároveň, číselné hodnoty zisku nejsou úplně realistické…

Můžeme se teď ptát, co je úspěšná metastrategie (tj. dlouhodobá taktika neboli postup volení strategií v průběhu života). Nemluvíme teď o jedné strategii (volbě C nebo D), ale o přiřazení alely (která bude určovat chování ptáčka po celý jeho život) definující dlouhodobou taktiku výběru, s kým spolupracuje (C) a s kým ne (D). Situaci si můžeme představit tak, že se narodí nový ptáček s alelou určující metastrategii, např. postup TitForTat, a pak si v průběhu života zahraje s každým z ostatních ptáčků v populaci jedno kolo hry bez opakování, kde získaná odměna odpovídá dlouhodobému průměru získaných odměn při hraní vězňova dilematu se zafixovanými metastrategiemi (viz Problém 2).

**Problém 1** [7b]: *Upravte model konfliktu hrdlička-jestřáb, aby popisoval děj, ve kterém se dvě hrdličky nedělí o potravu napůl, ale soutěží v tom, kdo déle zůstane koukat na jídlo. Až to jedna z nich vzdá a odletí, druhá hrdlička sní celé jídlo. Rozdíl bude v tom, že se levá horní hodnota v bimatici se sníží o nějaké malé číslo $\varepsilon$, jež bude vyjadřovat penalizaci za ztrátu času při čekání v případě setkání hrdličky s hrdličkou. Jaká bude v upravené hře optimální strategie pro jedince? A co by v upravené hře bylo nejlepší pro celou populaci v součtu?*

**Problém 2** [8b]: *Převeďte iterované vězňovo dilema na obyčejnou bimaticovou hru, popište několik metastrategií a vyjádřete jejich vzájemné zisky (průměrný počet bodů na kolo při dlouhé interakci s daným protějškem). Analyzujte optimální metastrategii v této „statické verzi“ vězňova dilematu.*

**Problém 3** [6b]: *Najděte optimální (smíšenou) strategii pro každé hráče v této bimaticové hře bez opakování. K jakému průměrnému výsledku každého hráče hra povede, pokud se oba hráči budou rozhodovat čistě racionálně? (Tato úloha nevychází z textu tématka.)*

| | | |
|---|---|---|
| $3 \setminus 9$ | $5 \setminus 4$ | $6 \setminus 0$ |
| $7 \setminus 0$ | $2 \setminus 4$ | $5 \setminus 9$ |
| $3 \setminus 0$ | $8 \setminus 4$ | $4 \setminus 9$ |

## Řešení 2. série

Následujících pět úloh vychází ze hry určené níže uvedenou maticí $A$. První hráč má na výběr ze tří strategií, druhý hráč má na výběr ze dvou strategií.

| | |
|---|---|
| $1$ | $-2$ |
| $3$ | $-4$ |
| $-1$ | $2$ |

### Problém 1: optimální strategie prvního hráče

**Zadání:**
*Určete optimální strategii pro prvního hráče.*

**Řešení:**
Jako první si všimneme, že pro prvního hráče není výhodné používat strategii 1. Průměr mezi strategií 2 a strategií 3 nám dává vektor $(1; -1)$, což je lepší než vektor $(1; -2)$ odpovídající strategii 1. Z toho plyne, že pokud by smíšená strategie $(x_1; x_2; x_3)$ měla být optimální, pak smíšená strategie $(0; x_2 + \frac{x_1}{2}; x_3 + \frac{x_1}{2})$ je také optimální (neboť musí dát stejný nebo lepší průměrný výsledek[3]). Tím pádem můžeme bez újmy na obecnosti předpokládat, že optimální strategie prvního hráče má nulovou pravděpodobnost volby strategie 1. Tímto se nám hra zredukovala na matici $2 \times 2$:

| | |
|---|---|
| $3$ | $-4$ |
| $-1$ | $2$ |

[3] Pokud bychom trochu změnili pravděpodobnosti a místo strategie 1 bychom použili strategii 2 s pravděpodobností 51 % a strategii 3 s pravděpodobností 49 %, pak by výsledný vektor byl $(1{,}04; -1{,}06)$, což je dokonce v obou složkách ostře lepší než $(1; -2)$ patřící strategii 1. Nicméně pro zdůvodnění neoptimality strategie 1 toto pozorování není třeba.

Označme si pravděpodobnost volby strategie 2 (tj. první řádek upravené matice) proměnnou $p$. Budeme tedy zkoumat smíšené strategie ve tvaru $(0; p; 1-p)$. Nyní musíme uvážit nejnepříjemnější možnou strategii druhého hráče. Zde se bez újmy na obecnosti můžeme omezit na ryzí strategie druhého hráče, protože (když zafixujeme smíšenou strategii prvního hráče) výsledky jakékoliv smíšené strategie druhého hráče budou ležet v intervalu mezi výsledkem při použití strategie 1 druhým hráčem a výsledkem při použití strategie 2 druhým hráčem. Když druhý hráč zvolí strategii 1, my v průměru získáme $3p-(1-p)=4p-1$. Když druhý hráč zvolí strategii 2, my v průměru získáme $-4p+2\cdot(1-p)=2-6p$. Všimneme si, že první funkce je rostoucí v $p$, kdežto druhá funkce je klesající v $p$. Tím pádem maximum funkce $\min\{4p-1; 2-6p\}$ nastává při rovnosti $4p-1=2-6p$. Ekvivalentní úpravou získáme rovnici $10p=3$, takže 0,3 je optimální hodnota $p$.

Vypočítali jsme, že optimální strategie prvního hráče je (0 %; 30 %; 70 %).

### Problém 2: optimální strategie druhého hráče

**Zadání:**

*Určete optimální strategii pro druhého hráče.*

**Řešení:**

Druhý hráč může provést stejnou úvahu jako první hráč, takže hru také zredukuje na matici $2\times 2$, uvedenou výše. Označme si pravděpodobnost volby strategie 1 proměnnou $q$. Hledáme hodnotu $q$ takovou, že smíšená strategie $(q; 1-q)$ nabyde co nejnižší hodnoty $\max\{3q-4\cdot(1-q);\ -q+2\cdot(1-q)\}$. První funkce je rostoucí v $q$, druhá funkce je klesající v $q$, takže optimální hodnotu najdeme opět v průsečíku. Rovnice $7q-4=2-3q$ nám dá řešení $q=0{,}6$.

Vypočítali jsme, že optimální strategie druhého hráče je (60 %; 40 %).

### Problém 3: průměrný výsledek při optimálních strategiích

**Zadání:**

*K jakému průměrnému výsledku vede, pokud oba hráči použijí optimální strategii?*

**Řešení:**

$$\left(0; \frac{3}{10}; \frac{7}{10}\right)\cdot A\cdot\left(\frac{3}{5}; \frac{2}{5}\right)^T = 0\cdot 1 + 0\cdot(-2) + \frac{9}{50}\cdot 3 + \frac{6}{50}\cdot(-4) + \frac{21}{50}\cdot(-1) + \frac{14}{50}\cdot 2 =$$

$$= \frac{27-24-21+28}{50} = \frac{10}{50} = \frac{1}{5}$$

Pokud oba hráči použijí optimální strategii, bude průměrný zisk roven 0,2 pro prvního hráče.

Problém 4: možné průměrné výsledky, pokud pouze první hráč hraje optimálně

**Zadání:**
*Jaké jsou možné průměrné výsledky, pokud první hráč zvolí optimální strategii, ale druhý hráč se od optimální strategie odkloní?*

**Řešení:**

$$\left(0; \frac{3}{10}; \frac{7}{10}\right) \cdot A \cdot (q; 1-q)^T = \left(\frac{2}{10}; \frac{2}{10}\right) \cdot (q; 1-q)^T = \frac{1}{5}$$

Pokud první hráč použije optimální strategii a druhý hráč se od optimální strategie odkloní, bude průměrný zisk stále a jenom 0,2 pro prvního hráče.

Problém 5: možné průměrné výsledky, pokud pouze druhý hráč hraje optimálně

**Řešení:**

$$A \cdot \left(\frac{3}{5}; \frac{2}{5}\right)^T = \left(-\frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right)^T$$

Z výpočtu vidíme, že jakákoliv konvexní kombinace strategie 2 a strategie 3 dává stále stejný výsledek 0,2, ale použití strategie 1 dává horší výsledek −0,2, tedy ztrátu pro prvního hráče. Jakákoliv smíšená strategie je jejich konvexní kombinací, takže průměrný výsledek prvního hráče při optimální volbě druhého hráče bude ležet vždy v intervalu $\langle -0{,}2\,; 0{,}2\rangle$.

## Výsledky turnaje z 3. čísla

| | řešitel | body | body do M&M |
|---|---|---|---|
| 2. | Dr.^MM Kateřina Vokálová | 14 808 | 10 b |
| 3. | Bc.^MM Vilém Starosta | 13 863 | 7 b |
| 4. | Mgr.^MM Martin Fof | 13 833 | 5 b |
| 6. | Mgr.^MM Václav Janáček, Dr.^MM Jiří Kalvoda | 13 775 | 3 b |
| 11. | Bc.^MM Klára Pernicová | 5 717 | 1 b |

Odteď budeme turnaj opakovat každé dva týdny (akorát s nižšími bodovými odměnami do M&M). Při každém spuštění turnaje použijeme jiný počet kol (abyste nemohli optimalizovat na ukončení spolupráce několik kol před koncem). Současní účastníci budou automaticky zařazeni i do dalších kol, neuvedou-li jinak. Kdybyste se chtěli připojit do turnaje někdo další, neváhejte a pište. Rádi vám poradíme s technickou stránkou věci (psaní, testování a odevzdávání skriptů).

Zdrojové kódy vašich řešení nebudeme zveřejňovat, abyste je nemohli od sebe navzájem kopírovat. Pokud chcete získat orientační představu o tom, jak se které řešení v turnaji chovalo, podívejte se do souboru s podrobnými výsledky, který najdete na `https://mam.mff.cuni.cz/media/prilohy/26_4_turnaj_vysledky.txt`.

*Markét, Martin;* `martin.dvorak@matfyz.cz`
*e-mailová konference:* `hry@mam.mff.cuni.cz`

# Alternativní metody pro hledání optimálních strategií

*Bc.$^{MM}$ Vilém Starosta* 10 b

Při hledání optimálních smíšených strategií (dále jen optimálních strategií) vyvstává otázka, zda je možné použít jednodušší metodu, než je lineární programování. V tomto článku se budu konkrétně zabývat hledáním optimálních strategií v maticích, ve kterých mají oba hráči na výběr mezi dvěma strategiemi, a pokusím se najít obecný způsob, kterým je lze řešit. (Může se zdát, že řešení tohoto speciálního případu není nijak zvláště přínosné, ale velké množství her lze do tvaru $2 \times 2$ zjednodušit, a tak tento postup nalezne uplatnění i ve hrách větších.)

Představme si matici, ve které si mohou oba hráči vybírat mezi dvěma strategiemi. Pravděpodobnost, se kterou první hráč vybírá svoji první strategii, označme výrazem $x$, pravděpodobnost volby jeho druhé strategie tedy označme výrazem $1-x$. Analogicky u druhého hráče budou pravděpodobnosti volby jeho strategií popisovat výrazy $y$ a $1-y$. Jednotlivé prvky matice (zisky) označím písmeny $a$, $b$, $c$, $d$.

| | $y$ | $1-y$ |
|---|---|---|
| $x$ | $a$ | $b$ |
| $1-x$ | $c$ | $d$ |

Průměrný výsledek v tomto obecném tvaru udává výraz:

$$axy + bx(1-y) + c(1-x)y + d(1-x)(1-y)$$

Průměrný zisk druhého hráče pro jeho první strategii, pokud první hráč použije smíšenou strategii, mohu zapsat pomocí výrazu $ax + c(1-x) = ax - cx + c$ a pro jeho druhou strategii analogicky výrazem $bx + d(1-x) = bx - dx + d$.

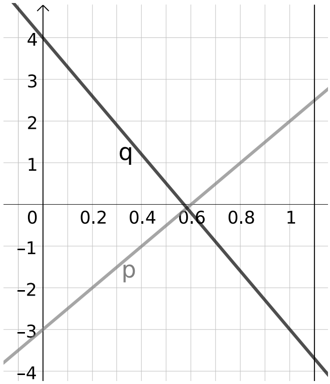
První hráč potřebuje, aby bez ohledu na to, jakou strategii zvolí druhý hráč, získal co největší počet bodů. Zmíněné výrazy mohu znázornit jako funkce a sledovat, pro jakou hodnotu $x$ platí, že funkce $y = \min\{ax - cx + c; bx - dx + d\}$ na intervalu $\langle 0; 1\rangle$ zde má maximum. Lépe pochopitelné to bude na konkrétním příkladu. Představme si matici[4]:

| | $p$ | $q$ |
|---|---|---|
| $x$ | $2$ | $-3$ |
| $1-x$ | $-3$ | $4$ |

Jednotlivé strategie druhého hráče jsem označil písmeny $p$ a $q$. Funkce udávající průměrný zisk pro jeho jednotlivé strategie v závislosti na smíšené strategii prvního hráče budou tedy vypadat takto:

[4] Tato hra je známá pod názvem „morra“. Funguje tak, že každý z obou hráčů ukáže buď jeden nebo dva prsty. Pokud oba hráči ukáží jeden prst, získává první hráč 2 body, pokud ukáže každý dva prsty, získává první hráč 4 body. V ostatních případech získává druhý hráč 3 body.

$$p = 2x + (-3)(1 - x) = 5x - 3$$
$$q = -3x + 4(1 - x) = -7x + 4$$




Nyní hledám bod, ve kterém má funkce $y = \min\{5x-3; -7x+4\}$ své maximum na intervalu $\langle 0; 1\rangle$. V tomto konkrétním případě tento bod odpovídá průsečíku obou funkcí. Když by totiž první hráč zvolil jiný bod než zmíněný, druhý by toho mohl zneužít a častěji volit strategii, která by mu přinesla více bodů, tzn. pokud by první hráč zvolil $x$ větší, tak by druhý hráč častěji volil strategii $q$, a naopak. Hodnota $x$, která určuje pravděpodobnost volby první strategie prvního hráče, lze tedy vyjádřit z rovnice

$$5x - 3 = -7x + 4 \Rightarrow x = \frac{7}{12} \text{ (průměrný výsledek je pak } -\tfrac{1}{12}\text{).}$$

Vraťme se k obecnému případu, tedy k matici, která byla v článku zmíněna jako první. $x$-ová souřadnice průsečíku určuje optimální strategii pouze v případě, pokud je jedna funkce rostoucí a druhá klesající a pokud průsečík leží v intervalu $\langle 0; 1\rangle$. Jedině tak totiž platí, že průsečík je maximum funkce $\min\{ax - cx + c; bx - dx + d\}$ na intervalu $\langle 0; 1\rangle$. Obě funkce jsou rostoucí pouze v případě, kdy platí $v_1\colon (a > c) \wedge (b > d)$; a naopak klesající, pokud platí $v_2\colon (a < c) \wedge (b < d)$. V prvním případě je u obou $x$ ve výrazech popisujících funkce kladný koeficient a v druhém případě záporný koeficient. Pokud jsou obě funkce rostoucí (tedy platí $v_1$), jednoduchou úvahou dojdeme k tomu[5], že $x = 1$. První strategii tudíž volí první hráč v každém případě. Naopak pokud jsou obě funkce klesající, určuje optimální strategii $x = 0$, takže hráčova první strategie není volena vůbec a je volena pouze strategie druhá. Jednoduše si také můžeme říct, že když jsou ve všech případech zisky při použití první strategie větší než

[5]Důkaz lze provést po jednotlivých případech (pro funkce, které mají průsečík v intervalu $(-\infty; 0)$, pro funkce, které mají průsečík uvnitř intervalu $\langle 0; 1\rangle$, pro funkce, které průsečík nemají atd.).

při použití druhé strategie, tak je zřejmé, že hráčovou optimální strategií je volba první strategii ve všech případech (a naopak). Dále může nastat situace, ve které je jedna z funkcí konstantní. Jinými slovy platí buď, že $a = c$, nebo $b = d$. Jestliže tento případ nastane, tak záleží na tom, zda je druhá funkce rostoucí nebo klesající. Pro rostoucí funkci platí, že maximum funkce $y$ je interval od průsečíku obou funkcí do $x = 1$, tudíž $x \in \langle p; 1\rangle \cap \langle 0; 1\rangle$ ($p$ získáme jako $x$-ovou souřadnici průsečíku funkcí). Obdobně je tomu v případě, kdy je druhá funkce klesající, tentokrát $x \in \langle 0; p\rangle \cap \langle 0; 1\rangle$[6]. Jakým způsobem najít průsečík si ukážeme již v následujícím odstavci. V případě, že jsou obě funkce konstantní, tak to v prvé řadě znamená, že má matice sedlové body; dále to znamená, že nezáleží na tom, jakou strategii první hráč volí, protože je v obou případech zisk stejný.

Pokud nejsou funkce obě rostoucí nebo obě klesající, ani není jedna či obě konstantní, tak můžeme hledat optimální strategii pomocí průsečíku funkcí, tudíž vyjádříme $x$ z rovnice:

$$ax - cx + c = bx - dx + d$$

$$x(a - c - b + d) = d - c$$

$$x = \frac{d - c}{a - c - b + d}$$

$$1 - x = \frac{a - c - b + d}{a - c - b + d} - \frac{d - c}{a - c - b + d} = \frac{a - b}{a - c - b + d}$$

Zde přirozeně vyvstává otázka, za jakých podmínek je výraz ve jmenovateli rovný nule. Pokud platí $v_3\colon a - c - b + d = 0$, neboli $a + d = b + c$, pak musí platit $v_4\colon (a \geq b) \wedge (c \geq d)$ nebo $v_5\colon (a < b) \wedge (c < d)$. Z $v_4$ plyne, že body $b$ a $d$ jsou minimální na svých řádcích, v závislosti na hodnotách $b$ a $d$ tak bude jeden z nich sedlovým bodem (pokud $b = d$, tak budou dokonce oba sedlovými body, a v případě, že $a = b = c = d$, budou všechny body matice sedlové). Z $v_5$ analogicky vyplývá, že sedlovým bodem bude $a$ nebo $c$ (nebo oba)[7]. Pokud si ověříme, že matice nemá sedlový bod, můžeme výraz pro hledání průsečíku funkcí, respektive optimálních strategií, s klidným svědomím použít.

Nyní tedy dosadíme konkrétní hodnoty do výrazu. Pokud nám vyšlo $x \in \langle 0; 1\rangle$, tak můžeme říct, že jsme našli optimální strategii pro prvního hráče. Hráč volí svou první strategii s pravděpodobností $x$ a druhou s pravděpodobností $1 - x$. Co když nám ale vyšlo $x$ mimo tento interval? Znovu se vraťme ke grafům funkcí. Víme, že jedna z funkcí $ax - cx + c$ a $bx - dx + d$ je klesající a druhá rostoucí. Stále postupujeme stejně, hledáme maximum funkce $y = \min\{ax - cx + c; bx - dx + d\}$

[6] Pro svou optimální strategii může první hráč teoreticky volit jakékoli $x$ ze zmíněných intervalů. Pokud je ale první hráč ochoten počítat i s faktem, že se jeho protivník může od optimální strategie odchýlit, je pro něj výhodnější zvolit v případě jedné rostoucí funkce $x = 1$, v případě klesající funkce $x = 0$. Pokud by se druhý hráč odchýlil od své strategie, ztratí tak více bodů.

[7] Když vztáhneme tyto vlastnosti k analogii funkcí, kterou v tomto článku používáme, tak zjistíme, že získané funkce nebudou mít průsečík, tedy budou rovnoběžné nebo budou splývat. Pokud budou obě rostoucí, tak optimální strategii určuje $x = 1$, pokud budou klesající, tak $x = 0$; pokud konstantní, tak určuje optimální strategii libovolné $x$.

na intervalu $\langle 0; 1\rangle$. Když si funkce představíme, tak snadno dojdeme k tomu, že pokud je $x$-ová souřadnice průsečíku větší než 1, tak je hledané maximum v bodě 1, tedy optimální strategii prvního hráče odpovídá $x = 1$, a pokud je $x$-ová souřadnice průsečíku menší než 0, tak je hledané maximum v bodě 0, takže $x$ určující optimální strategii $= 0$.

Při hledání optimální strategie pro druhého hráče postupujeme analogicky. Všechny úvahy vychází v tomto případě z funkcí $r = ay - by + b$ a $s = cy - dy + d$. Důležité je uvědomit si, že v případě druhého hráče hledáme naopak minimum funkce $x = \max\{ay - by + b; cy - dy + d\}$. Výsledný výraz pro hledání optimální strategie druhého hráče je: $y = \frac{d-b}{a-c-b+d}$.

Na závěr si shrňme celý postup hledání optimálních strategií po prvního hráče (pro druhého jsou úvahy analogické). Nejdříve ověřme, že matice nemá sedlový bod. Pokud ho má, je optimální strategie obou hráčů pevně dána. Poté se podívejme, zda platí $v_1\colon (a > c) \wedge (b > d)$. Jestliže platí, tak $x = 1$. Jestliže naopak platí $v_2\colon (a < c) \wedge (b < d)$, tak optimální strategii udává $x = 0$. Pokud neplatí ani jeden z výroků, použijme vzorec $x = \frac{d-c}{a-c-b+d}$. Když nám vyjde $x \in \langle 0; 1\rangle$, tak jsme našli optimální strategii. Jestliže platí $a = c$ nebo $b = d$, musíme ještě provést úvahu, ze které vyplyne, že hodnotu $x$ můžeme vybírat z intervalu $\langle p; 0\rangle$ nebo $\langle p; 1\rangle$, v závislosti na tom, zda je druhá funkce rostoucí či klesající. Pokud je $x$ menší než 0, tak optimální strategii odpovídá $x = 0$, pokud je větší než 1, tak $x = 1$.

# Téma 2 – Výpočetní modely

## Náhoda a nedeterminismus

### Generátor náhodných čísel

Dosud byl výsledek výpočtu pevně určen popisem stroje a jeho vstupy. Pojďme tuto vlastnost porušit a pořídit si generátor náhodných čísel. Dále jej budeme označovat jako RNG, což je zkratka anglického označení *random number generator*.

RNG je rozšíření, které v každém kroku vypíše na výpis buď jedničku, nebo nulu. Budeme předpokládat, že generátor je dokonale náhodný – pravděpodobnost, že RNG v prvních $n$ krocích vygeneruje danou posloupnost jedniček a nul délky $n$ je $\frac{1}{2^n}$ pro libovolné $n$ a libovolnou posloupnost délky $n$. (Tohle je poměrně formální popis toho, co si představíte pod „skutečnou“ náhodou.)

**Problém 1**: *Zamyslete se nad tím, jaké všechny vlastnosti od RNG chceme. Ukažte, že námi popsaná definice je splňuje. Pokud bychom měli jiné rozšíření, které některé z nich nesplňuje, umíme pomocí něj sestrojit stroj, jehož výstup by tyto vlastnosti měl? Pro představu uvádíme dva příklady alternativních rozšíření: „V každém kroku si nejdříve vygeneruje skutečně náhodný bit. Pokud je to 1, tak vygeneruje ještě jeden a ten vypíše. Pokud je to 0, tak zopakuje ten, co vypsal posledně.“ „Kdyby měl vypsat třetí jedničku v řadě, tak místo ní vypíše nulu.“*

**Problém 2**: *Co nám RNG umožňuje řešit (nebo alespoň usnadňuje)?*

## Nedeterminismus

Následující problém je údajně inspirován situací, kdy si dva loupežníci potřebovali rozdělit lup skládající se z různě cenných nedělitelných předmětů.[8] Proto se mu většinou říká „problém dvou loupežníků“: Lze rozdělit danou množinu čísel na dvě podmnožiny se stejným součtem?

Dovolím si uvést pár příkladů pro ujasnění:

| množina | odpověď | řešení |
|---|---|---|
| 1, 2 | NE | |
| 1, 2, 3 | ANO | 1, 2 \| 3 |
| 1, 2, 3, 4 | ANO | 1, 4 \| 2, 3 |
| 1, 2, 3, 4, 5 | NE | |
| 1, 2, 5 | NE | |

**Problém 3**: *Popište stroj na řešení problému dvou loupežníků. Uveďte jeho asymptotickou složitost.*

Nedeterminismus je mocný nástroj, na který se dá nahlížet několika různými způsoby. Asi nejsnazší způsob je následující: Máme problém, na který je skutečnou odpovědí buď ANO, nebo NE. Sestavíme stroj, který používá RNG a odpovídá buď ANO, nebo NEVÍM. Pokud splní následující dvě podmínky, tak říkáme, že stroj nedeterministicky řeší daný problém.

1. Pokud je skutečnou odpovědí NE, tak stroj nesmí nikdy odpovědět ANO.
2. Pokud je skutečnou odpovědí ANO, tak existuje šance, že stroj odpoví ANO.

Časovou složitost nedeterministického stroje pro daný vstup určujeme podle **nejkratšího** času potřebného na odpovězení ANO. Můžete si to představit tak, že pouštíme nekonečné množství kopií stroje najednou. Pokud víme, že pro libovolný vstup dané délky s odpovědí ANO by už alespoň část kopií stroje odpověděla ANO a zatím se tak nestalo, tak je určitě odpovědí NE.

[8] Předpokládejme, že se na cenách shodli, nebo (pokud se to odehrálo po vládě krále Šalamouna) možná pomáháme jednomu, aby jej druhý nemohl obrat o většinu lupu.

`vstup` (Tady dostaneme délku pole – předpokládejme, že je to mocnina dvojky)
`RNG` (Generátor náhodných bitů pro nedeterministické řešení)
`ALU` (Tato jednotka bude opakovaně dělit délku pole dvěma, abychom provedli log($n$) kroků.)
`ALU` (Tato jednotka postupně vypočítá náhodné číslo od 0 do n-1)
`pole` (Tady hledáme jedničku)
`výstup`

`n,_,_,_,_ → n,2,/,0,0,+,_,_,` (Inicializace)
`$,_,0,b,_ → 3,0,+,0,0,+,b,_,` (Když první ALU dosáhne nuly, můžeme přečíst náhodnou pozici v poli. Využíváme toho, že první ALU standardně vypisuje mocniny dvojky.)
`$,_,3,0,1 → 0,0,/,0,0,/,_,_,ANO`
`$,_,3,0,a → 0,0,/,0,0,/,_,_,NEVÍM`
`$,0,a,b,_ → a,2,/,b,0,+,_,_,` (Nulový náhodný bit)
`$,1,a,b,_ → a,2,/,b,a,+,_,_,` (Jedničkový náhodný bit)

**Stroj 1:** Příklad stroje, který nedeterministicky hledá jedničku v poli, jehož délka je mocnina dvojky. Stroj postupně dělí pole na poloviny a náhodně vybírá, do které půlky se podívá. Když mu zbývá jediná buňka, odpoví ANO pokud je to jednička (tj. trefil správnou buňku), jinak NEVÍ (buď v tom poli jednička vůbec není, nebo se jen netrefil).

**Problém 4**: *Popište stroj na nedeterministické řešení problému dvou loupežníků. Uveďte jeho asymptotickou složitost.*

Nedeterminismus dovoluje výrazně více než „jen“ urychlit výpočty. Některé kombinace rozšíření činí značně silnějšími.

**Problém 5**: *Popište stroj, který nedeterministicky poznává palindromy za použití pouze rozšíření RNG, IO a zásobník.*

**Problém 6**: *Popište stroj, který nedeterministicky pozná, zda množina čísel na vstupu obsahuje svůj průměr. Nepoužívejte zásobníky, fronty, pásky ani pole.*

**Problém 7**: *Představte si, že máte registr, který začíná na hodnotě 1, a vaším úkolem je zjistit, zda je možné jej vynulovat. Zádrhel je v tom, že registr má nějaká omezení na to, jak jej lze měnit. Dostanete číslo $n$ na vstupu a pole obsahující $n \times n$ bitů, kde bit na pozici $i \cdot n + j$ (číslováno od nuly) určuje, zda můžete změnit hodnotu v registru z $i$ na $j$, přičemž žádné jiné změny nejsou povoleny. Instance problému tedy může vypadat například takto:*

$n = 4$
*Obsah pole: 0000 0010 0001 1000*
*Pole reprezentuje následující tabulku:*

| | | nová | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| původní | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 |
| | 2 | 0 | 0 | 0 | 1 |
| | 3 | 1 | 0 | 0 | 0 |

*Odpověď: ANO (1 -> 2 -> 3 -> 0)*

*Jak rychle umíte tento problém řešit? Jak rychle jej umíte řešit nedeterministicky? Pokud vám to pomůže, tak můžete předpokládat, že n je mocnina dvojky.*

**Problém 8**: *Jakými jinými způsoby lze nahlížet na nedeterminismus? Jak by se to dalo implementovat v našich strojích?*

## Řešení úloh z druhého dílu

### Problém 1

**Zadání:**
*Sestrojte stroj, který pozná správně uzávorkované řetězce.*

**Řešení:**
Budeme si na zásobník ukládat otevřené závorky. Vždy, když přečteme uzavírací závorku, jednu otevřenou závorku ze zásobníku smažeme („uzavřeme“ ji). Na konci výpočtu na zásobník vložíme jiný znak, čímž způsobíme zastavení.

```
zásobník
IO

$,( →  (,
(,( → ((,
(,) →   ,        (uzavírací závorky ubírají ze zásobníku)
$,$ →  X,ANO
(,$ →  X,NE      (pokud zbyly neuzavřené závorky, odpověď je NE)
$,) →  X,NE      (stejně tak pokud přebývají uzavírací závorky)
```

### Problém 2

**Zadání:**
*Sestrojte stroj, který pozná palindromy.*

**Řešení:**
Jedno z mnoha možných řešení je zkopírovat vstup do zásobníku a fronty. Fronta pak vypíše vstup od začátku a zásobník pozpátku, což stačí porovnat. Pro přehlednost budu předpokládat, že po odpovězení je ukončen výpočet. Dalo by se to zařídit přidáním stavového registru, což by ale znepřehlednilo popis stroje.

```
IO
zásobník
fronta

$,$,$ → ANO,  , ,
$,a,a →    ,  , ,>
$,a,b →  NE,  , ,
a,b,_ →    ,ba,a,
```

Problém 3

**Zadání:**
*Nalezněte stroj, který dělá totéž, co stroj 2 níže, ale*
*a) má co nejkratší popis.*
*b) potřebuje méně kroků.*

```
IO
fronta
stavový registr (počáteční stav S)

0,_,S →  ,0, ,L
1,_,S →  ,1, ,L
0,_,L →  ,0, ,S
1,_,L →  ,1, ,S
$,0,S →  , ,>,L
$,1,S →  , ,>,L
$,a,L → a, ,>,S
```

**Stroj 2:** Stroj z 2. čísla, který vypíše pouze znaky s lichou vzdáleností od konce.

**Řešení:**
Příklad řešení splňující oba požadavky najednou:

```
IO
fronta
stavový registr (počáteční stav S)

$,$,L →  , ,  ,X
$,_,S →  , , >,L
$,a,L → a, ,>>,L
a,_,S →  ,a,  ,L
a,_,L →  ,a,  ,S
```

Problém 4

**Zadání:**
*Sestrojte násobičku. (Násobička na dvou vstupech dostane dvě přirozená čísla ve dvojkové soustavě a na výstup vypíše jejich součin.)*

**Řešení:**

```
stavový registr (počáteční stav K)
vstup (od nejméně významného bitu)
vstup (od nejméně významného bitu)
```

`fronta` (Tady si uložíme první vstup)

`fronta` (Tady si uložíme druhý vstup)

`fronta` (Tady budeme průběžně udržovat mezisoučet)

`výstup`

(Kopírování vstupů do front)

`A,$,$,_,_,$ → B,X, ,X, ,X, ,` (Na konec si přidáme zarážku a začneme násobit)

`A,$,b,_,_,$ → A, , ,b, , , ,`

`A,a,$,_,_,$ → A,a, , , , , ,`

`A,a,b,_,_,$ → A,a, ,b, , , ,`

(Kontrola toho, jak vypadá druhý vstup)

`B,$,$,_,X,X → X, , , , , , ,` (Kontrola, že druhý vstup je neprázdný)

`B,$,$,_,0,X → B, , , ,>, , ,0` (Pokud na konci druhého vstupu jsou nuly, tak rovnou vypisuju nulové bity.)

`B,$,$,_,1,X → C, , , , , , ,` (Tady můžeme začít násobit)

(Zamysleme se nad aktuálním stavem.)
(Stavový registr je ve stavu C.)
(Oba vstupy jsou vyčerpány.)
(V první frontě vidíme nejméně významný ještě nezpracovaný bit prvního vstupu.)
(V druhé frontě vidíme nejméně významný bit druhého vstupu.)
(Ve třetí frontě vidíme nejméně významný bit přenosu.)

(Pokud skončil první vstup, tak končíme)

`C,$,$,$,1,X → X, , , , , , ,`

`C,$,$,$,1,a → C, , , , , ,>,a`

(Pokud v prvním vstupu narazíme na nulu, tak rovnou vypíšeme další bit)

`C,$,$,0,1,X → C, ,>, , , , ,0` (Pokud v přenosu nic není, tak nula)

`C,$,$,0,1,a → C, ,>, , , ,>,a` (Jinak poslední bit přenosu)

(Pokud v prvním vstupu narazíme na jedničku, tak vypíšeme bit a jdeme přičítat k přenosu)

`C,$,$,1,1,0 → D, ,>,1,>, ,>,1`

`C,$,$,1,1,1 → E, ,>,1,>, ,>,0` (Stav E indikuje přenášení jedničky)

(Sčítání)

`D,$,$,_,X,X → C, , ,X,>,X,>,` (Když skončí přičítání, tak zahájíme další cyklus.)

`D,$,$,_,a,X → D, , ,a,>,a,>,` (Druhý vstup může být delší než přenos, naopak to být nemůže.)

```
D,$,$,_,a,0 → D, , ,a,>,a,>,
D,$,$,_,0,1 → D, , ,0,>,1,>,
D,$,$,_,1,1 → E, , ,1,>,0,>,

E,$,$,_,X,X → D, , , , ,1, ,
E,$,$,_,0,X → D, , ,0,>,1, ,
E,$,$,_,1,X → E, , ,1,>,0, ,
E,$,$,_,0,0 → D, , ,0,>,1,>,
E,$,$,_,1,0 → E, , ,1,>,0,>,
E,$,$,_,a,1 → E, , ,a,>,a,>,
```

## Problém 5

**Zadání:**

*Představte si, že nemáte k dispozici pásku. Sestavte stroj, který ji simuluje – na jednom vstupu čte znak k zápisu, na druhém směr posunu a na výstup píše aktuálně čtený znak.*

**Řešení:**

Příklad možného řešení je níže. Doporučuji nejdříve zkoumat normální chování, a až pak speciální případy.

`vstup`

`vstup`

`výstup`

`zásobník` (Tady si udržujeme známou část pásky vlevo od nás)

`zásobník` (Tady si udržujeme známou část pásky vpravo od nás)

(Speciální případy pro první krok:)

```
a,␣,$,$ → a,  ,
a,>,$,$ → ␣, a,
a,<,$,$ → ␣,  ,a
```

(Speciální případy pro okraje zatím viděné oblasti:)

```
a,␣,$,r → a,  ,r
a,>,$,r → r, a,
a,<,$,r → ␣,  ,ra
```

```
a,␣,l,$ → a, l,
a,>,l,$ → ␣,la,
a,<,l,$ → l,  ,a
```

(Normální chování:)

```
a,␣,l,r → a, l,r
a,>,l,r → r,la,
a,<,l,r → l,  ,ra
```

### Problém 6

**Zadání:**
*Jaká další rozšíření bychom mohli chtít? (Zamyslete se nad tím, jak s nimi bude interagovat přechodová funkce.)*

**Řešení:**
Viz další díly tématka. :-)

### Problém 7

**Zadání:**
*Zamyslete se nad tím, jak jinak bychom mohli dostávat vstupní řetězce. Které problémy by to usnadnilo? Co by to zkomplikovalo? Chtěli byste změnit i něco jiného? Proč a jak?*

**Řešení:**
Možnosti zahrnují vstup v zásobníku, frontě i pásce. Obecně nám to usnadňuje práci v situaci, kdy například stroj nestíhá zpracovávat vstup stejnou rychlostí, kterou jej aktuálně dostává.

Zásobník a fronta se liší tím, zda si stroj může připisovat věci na začátek nebo konec vstupu, zatímco páska umožňuje se ve vstupu pohybovat v obou směrech. Nutno říci, že s výjimkou zásobníku nám to přidává hned dvě položky do pokynu, což může být příliš.

Ještě zmíním, že poměrně častým řešením je zavedení „vstupní“ fronty nebo pásky, která je určená pouze ke čtení, pročež si vystačí s jednou položkou v pokynu. Toto řešení má také tu příjemnou vlastnost, že takové stroje jsou o něco slabší než ty se schopností modifikace vstupu, a tím umožňují o něco jemnější analýzu složitosti problému.

### Problém 8

**Zadání:**
*Někteří z vás již možná tuší, co je to nedeterminismus. (Pokud ne, tak se o něm dozvíte v příštím čísle.) Jak by se dal formulovat v řeči našich modelů?*

**Řešení:**
Viz tento díl tématka. :-)

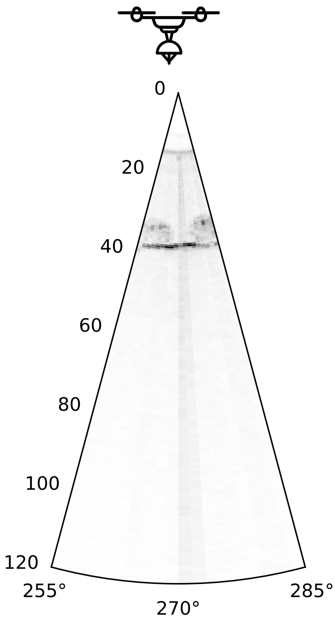
*Matej;* lieskovsky.matej+stroje@gmail.com
*e-mailová konference:* stroje@mam.mff.cuni.cz

# Téma 3 – Zpracování obrazových dat ze senzorů

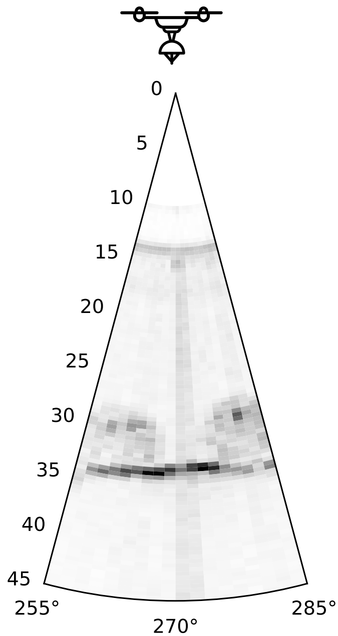
## 4. díl

V minulém díle jsme se zabývali převážně odstraňováním šumu z naměřených dat. Nyní nás čekají čistě programovací úkoly a na závěr i jeden úkol vhodný pro neinformatiky.

Doposud jsme měli k dispozici dvojrozměrné pole nekalibrovaných naměřených hodnot, kde jeden rozměr reprezentoval vzdálenost od senzoru a druhý směr. Přestože data byla naměřena v radiálních souřadnicích, pro jednoduchost jsem vám je zobrazoval jako obrázky. Přesnější vizualizace v radiálních souřadnicích vypadá takto:




**(a)** Originální obrázek v radiálních souřadnicích




**(b)** Stejný obrázek, ale přiblížený

**Problém 1**: *Naprogramujte v libovolném programovacím jazyce (doporučuji Python) funkci, která na vstupu dostane 2D pole s naměřenými daty a na výstupu vrátí vizualizaci těchto dat podobnou obrázku výše. Vstupní data najdete ve formátu* `.npy` *pro Python (knihovna numpy) nebo ve formátu* `.csv` *pro ostatní programovací jazyky. Odevzdejte zdrojový kód kompletního programu (ne jen implementaci této funkce), který přečte vstupní data ze souboru a vizualizaci uloží jako obrázek, nebo ji alespoň zobrazí na obrazovce.*

Nyní naměřená data rozšíříme o třetí dimenzi. Ve chvíli, kdy zpět k senzoru dorazí signál odražený od statického objektu, frekvence tohoto odraženého signálu je stejná jako vysílaná. Jenže ve chvíli, kdy se sledovaný objekt přibližuje nebo oddaluje, je odražená frekvence změněna na základě Dopplerova efektu[9]. Tato měnící se frekvence nám v datech přidává zmíněný třetí rozměr.

Nová data jsou reprezentována jako trojrozměrné pole, kde první rozměr představuje vzdálenost (v rozsahu 0–255)[10], druhý frekvenci (v rozsahu 0–63) a třetí úhel natočení antény (v rozsahu 0–18). Naměřená hodnota uložená na daných souřadnicích je opět intenzita odraženého signálu v nekalibrovaných jednotkách. S těmito daty pracujeme stejně jako v minulém díle:

```
import numpy as np
image3D = np.load("filename.npy")
print(image3D.shape)
# Vypise (256, 64, 19),
# coz odpovida rozmerum (vzdalenost, frekvence, uhel)
```

Zobrazit 3D data jako obrázek už je o něco obtížnější. V minulém díle jsem vám ukázal 2D obrázky, které z těchto 3D dat vznikly sečtením intenzit všech frekvencí do jedné. Jinými slovy jsem sečetl všechny hodnoty $Y$ (intenzity různých frekvencí) v bodech ve stejné vzdálenosti pod stejným úhlem (tj. na stejných souřadnicích $X$ a $Z$).

```
image2D = np.sum(image3D, axis=1)
```

Kód výše můžeme pro názornost přepsat i bez použití funkce `np.sum()`:

```
image2D = np.zeros(256*19).reshape(256, 19)
for x in range(256):
    for z in range(19):
        for y in range(64):
            image2D[x,z] += image3D[x,y,z]
```

Vypočítaný 2D snímek můžeme zobrazit stejně jako v minulém díle:

```
import matplotlib.pyplot as plt
plt.imshow(image2D, interpolation='nearest', cmap='binary')
plt.show()
```

Z 3D dat si můžeme vybrat „řez“, který zobrazíme jako 2D obrázek. Následující příklad vybere z `image3D` hodnoty na všech souřadnicích $X$, $Y$, kde $Z = 0$:

```
plt.imshow(image3D[:,:,0], interpolation='nearest', cmap='binary')
plt.show()
```

[9] `https://cs.wikipedia.org/wiki/Doppler%C5%AFv_jev`

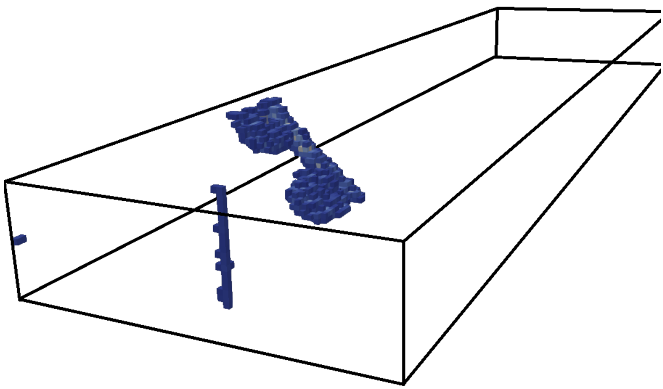
[10] Rozsah je daný použitým hardwarem.

Celý 3D snímek si můžeme zobrazit například jako animaci všech řezů podle libovolné osy nebo jako 3D graf:

**Problém 2**: *Libovolným způsobem vizualizujte 3D data jako 3D graf srozumitelný pro lidské oko. Můžete použít následující šablonu zdrojového kódu v jazyce Python:*

```
import numpy as np
import matplotlib.pyplot as plt
image3D = np.load("filename.npy")
print(image3D.shape)
#TODO: vizualizace promenne image3D
```

*Jako výsledek můžete získat například tento obrázek:*



***Bonus:*** *Zamyslete se nad tím, proč 3D naměřená data vypadají zrovna takto. Přidám malou nápovědu: Senzor umí měřit pouze radiální rychlost (tedy složka rychlosti ve směru k radaru).*

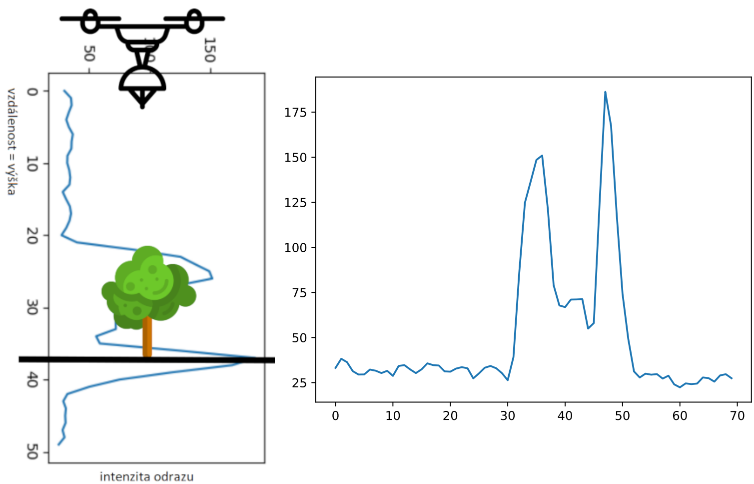
Nyní se opět vrátíme k problému odstraňování šumu. Když se podíváte na trojrozměrná data, tak zjistíte, že šum je najednou nižší než v předchozích 2D snímcích. V předchozích snímcích se sčítalo více naměřených hodnot, a tím vznikala kumulativní chyba.

**Problém 3**: *Zkuste nyní odfiltrovat šum z 3D snímku ještě před tím, než ho začneme převádět na 2D obrázek. Měli byste tak dosáhnout lepšího výsledku než v předchozím díle. Můžete k tomu použít šablonu níže:*

```
import numpy as np
import matplotlib.pyplot as plt
image3D = np.load("src/094057010609_3D.npy")
print(image3D.shape)
```

```
filteredImage3D = np.zeros((256, 64, 19))
for x in range(256):
    for y in range(64):
        for z in range(19):
            # Misto nasledujiho radku vlozte svuj filtr
            # na odstraneni sumu.
            # Zobrazeni na nasledujicim radku je nyni identita,
            # ktera zadny sum neodstrani.
            filteredImage3D[x,y,z] = image3D[x,y,z]

filteredImage2D = np.sum(filteredImage3D, axis=1)
plt.imshow(filteredImage2D)
plt.show()
```

**Problém 4 (Úkol vhodný i pro neinformatiky)**: *Vraťme se zpátky k prvnímu dílu, kde jsme naměřili jednorozměrný seznam hodnot. Tento seznam hodnot reprezentoval intenzitu odraženého signálu s narůstající vzdáleností od senzoru. Pokud se například 20 jednotek od senzoru nachází jediná překážka, tak seznam naměřených hodnot bude obsahovat nízké hodnoty (šum) pro všechny vzdálenosti menší než 20 jednotek. Naopak hodnoty reprezentující vzdálenost 20 jednotek budou výrazně vyšší, neboť z této vzdálenosti dorazil signál odražený od překážky. Pro upřesnění připomenu obrázkem z prvního dílu:*




*Zdrojová data k tomuto obrázku jsou uložena jako textový soubor, ve kterém je každá naměřená hodnota na samostatném řádku. Obsah souboru můžete například přímo kopírovat do Excelu, R či jiného nástroje.*

*Vaším úkolem je pro daný vstup najít dvě čísla reprezentující vzdálenost (souřadnice X). První číslo reprezentuje vzdálenost nejbližší překážky a druhé číslo reprezentuje vzdálenost nejvzdálenější překážky. Například pro obrázek výše je vzdálenost nejbližší překážky někde v intervalu 20 jednotek – 25 jednotek a nejvzdálenější někde na intervalu 38 jednotek – 42 jednotek. Odevzdávejte dvě čísla pro každý vstup. Dále odevzdejte jeden postup řešení, který jste použili pro všechny vstupy. Nalezení výsledků pomocí lidského oka nebude uznané jako řešení této úlohy.*

Data ke stažení k tomuto dílu

`https://mam.mff.cuni.cz/media/prilohy/26-4-3-data.zip`

## Řešení dílu 1 a 2

### Díl 1, zamyšlení 1

**Zadání:**
*Zkuste se zamyslet nad tím, jak byste na základě těchto dat měřili vzdálenost radaru od země (chtěli byste jej použít jako výškoměr). Odhadněte, jak spolehlivá data můžeme očekávat. Jestli třeba může nastat situace, kdy zem vůbec neuvidíme, jak bude naše měření ovlivňovat počasí nebo noc. Nemusíte nic programovat, ale pokud chcete, můžete své myšlenky ověřit na testovacích datech. Testovací data včetně návodu k jejich použití jsou na* `https://mam.mff.cuni.cz/media/prilohy/26-1-t3-data.zip`*.*

**Řešení:**
V průvodním textu je doslova uvedena věta: „Potom vidíme odraz od koruny stromu a následně od země. Aktuálně nemáme dostatečně silný radar, abychom viděli i pod zem, takže další odrazy nejsou měřitelné a v grafu už vidíme jen šum.“

Z toho vyplývá, že odraz od země je zároveň odraz od nejvzdálenějšího objektu. Ovšem pokud je na zemi nebo nad zemí příliš mnoho jiných překážek, tak se může stát, že na zem nedohlédneme vůbec. Naopak pokud jsme příliš vysoko, tak nemáme dostatečně silnou anténu na to, abychom odraz od země slyšeli. Intenzita signálu klesá s druhou mocninou (povrch koule)[11] vzhledem ke vzdálenosti.

Noc naše měření neovlivní, neboť radar si posílá vlastní pulzy (svítí sám). V průvodním textu je doslova napsáno: „Radar je anténa, která odešle krátký pulz elektromagnetického signálu na definované frekvenci a čeká na odrazy, které se vrátí.“

Naopak počasí má na měření nezanedbatelný vliv. V průvodním textu se dočteme: „Prvních několik metrů (postupujeme v ose $X$ od nuly) přijímáme jen

[11] Signál se šíří jako prostorová elektromagnetická vlna. Energie této vlny se rozprostírá rovnoměrně v celém svém povrchu, který můžeme reprezentovat povrchem rozpínající se koule. Povrch koule stoupá s druhou mocninou poloměru. Intenzita signálu je nepřímo úměrná k tomuto povrchu. Intenzita signálu tedy klesá s druhou mocninou vzhledem ke vzdálenosti.

malý odraz od vzduchu.“ Vzduch nám odeslaný signál částečně rozptyluje, pokud je vlhký, tak je rozptyl o něco větší. Pokud prší, tak dochází k částečnému pohlcování signálu kapkami vody, což nám způsobí kratší dohlednost se stejně silným vysílačem a se stejně citlivou anténou.

Obrázky v testovacích datech mají rozměr $256 \times 19$ pixelů, což znamená, že vzdálenost umíme měřit pouze v kvantech celého pixelu do maximální vzdálenosti 256 pixelů. Vzdálenost 256 pixelů může v reálu znamenat několik různých vzdáleností v metrech v závislosti na nastavení konkrétního radaru. (Na obrázcích v tomto tématku jeden pixel zhruba odpovídá vzdálenosti 0,5 jednotek.)

Pokud budeme chtít radar použít jako výškoměr, stačí nám hledat odraz od nejvzdálenější překážky. (Můžeme hledat i odraz od nejbližší překážky, což nám většinou poskytne informaci o výšce nad vrcholky stromů či nad střechami budov.) Pokud budeme hledat odrazy od překážek, musíme si stanovit intenzitu odrazu, kterou už budeme považovat za překážku a nikoliv za šum. Tento „threshold“ se relativně snadno počítá pomocí histogramu, zde si ale bohatě vystačíme s výrazně nadprůměrnými hodnotami.

Následující kód v jazyce Python otevře jeden snímek a vypočítá vzdálenost od země ve směru uloženém na indexu 5, radar tedy použije jako výškoměr:

```
import numpy as np

# Vybrany smer
direction = 5

# Precteni snimku ze souboru
image2D = np.load("1_2D.npy")
image1D = image2D[:, direction]

# Chceme jen vyrazne nadprumerne odrazy
average = np.sum(image1D)/len(image1D)
threshold = 2*average
maxDistance = 0
for y in range(256):
    if image1D[y] > threshold:
        maxDistance = y
print(maxDistance)
```

### Díl 1, zamyšlení 2

**Zadání:**

*Jak budou vypadat data pořízená nad rovnou loukou nebo třeba nad vodní hladinou? Jak bude vypadat snímek nad budovou, nad hustým nebo řídkým lesem, nad lesní cestou nebo v horách? Co všechno můžeme obyčejným radarem vidět z ptačí perspektivy? Pro inspiraci přidávám několik radarových snímků, ale hádání, co je jejich obsahem, už nechám na vás.*

**Řešení:**
Přímo v zadání jsem vám napověděl, co máte v obrázcích hledat. Vezměme to postupně:

1. Rovná louka, fotbalové hřiště, stejně bude vypadat libovolný rovný terén bez překážek a bez vegetace.

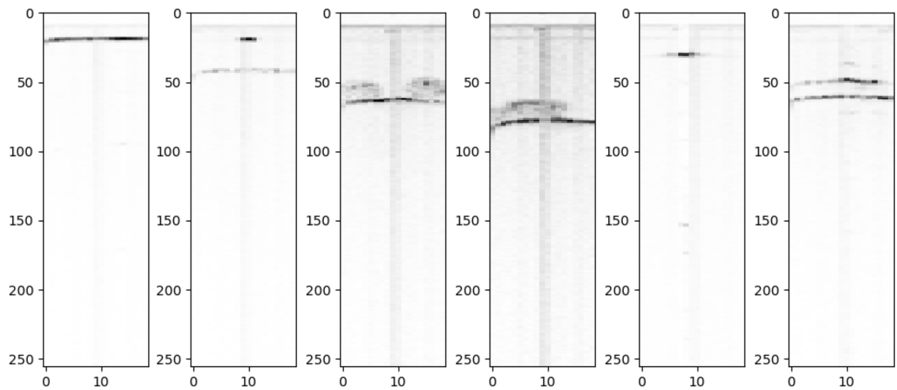
2. Stejná rovná louka z trošku větší výšky, ale tentokrát se mezi naší koptéru a zem vloudila ještě jiná koptéra, která simulovala létající překážku.

3. Průlet nad polní cestou, která je z obou stran lemována stromy.

4. Rozhraní louky a lesa, les se začíná svažovat mírně z kopce.

5. Toto byl asi nejtěžší snímek. Uznávám, že pokud bych nevěděl, kdy jsem tato data měřil, tak bych sám nepoznal, co je obsahem. Takto vypadá vodní hladina, která odráží signál jako zrcadlo. Tím pádem se k nám vrátí jen ten signál, který je od hladiny odražen pod pravým úhlem. Proto nevidíme celou hladinu, ale jen „tečku“. Signál, který dorazí k vodní hladině pod jiným úhlem, není odražen zpátky ke zdroji, ale naopak úplně na druhou stranu podle pravidla „úhel dopadu = úhel odrazu“.

6. Poslední snímek ukazuje pohled shora na zeď domu. Vrchní linka je odraz od střechy, spodní linka je odraz od podlahy.



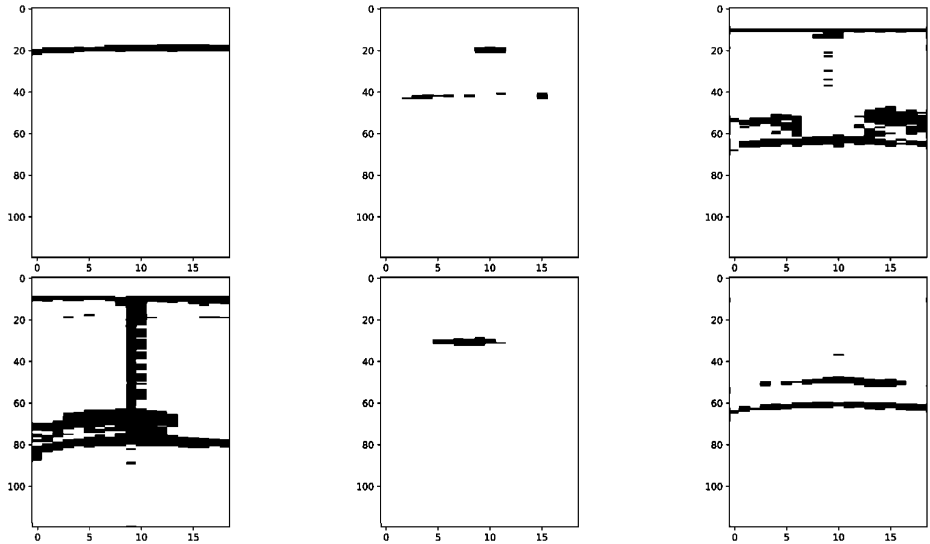

(1) (2) (3) (4) (5) (6)

## Díl 2, problém 1

**Zadání:**
*V minulém díle jste dostali celkem šest obrázků různých typů terénu, u kterých jste hádali, co asi zobrazují. Najdete je ve formátu PNG v odkazu na konci tohoto článku. Zkuste si tyto obrázky otevřít v nějakém grafickém editoru (doporučuji*

*například Gimp) a najít jeden postup aplikovatelný na všechny obrázky (ne úprava obrázku štětcem ručně pomocí myši), který z těchto obrázků co nejlépe odstraní šum a zvýrazní detekované objekty (zem, strom, skála, létající objekt, ...). Ve svých řešeních popište, jaké problémy jste museli řešit a jaké postupy jste zkoušeli aplikovat. Odevzdejte také postupy a vámi upravené obrázky.*

**Řešení:**
V rámci řešení tohoto problému jste si vyzkoušeli, že automatizované odstraňování šumu není zas tak jednoduchá operace. Akceptoval jsem libovolné řešení, které dosáhlo lepšího výsledku než obrázek v zadání. Osobně bych použil kombinaci filtrů rozmazání, filtr threshold (přebarvení všech bodů na černou nebo bílou podle zvolené hranice) 50% a nakonec filtr eroze, který odstraní všechny okrajové pixely. Tím se všechny objekty zmenší o jeden pixel a objekty šířky jeden pixel úplně zmizí. Filtr eroze je možné aplikovat vícekrát podle potřeby, například dokud nezbyde maximálně $N$ objektů. Všechny zmíněné filtry jsou podporované například programem Gimp.




**Obrázek 4:** Příklad výsledku získaného pomocí zmíněného postupu.

Nebudu zde tajit, že tímto procesem můžeme dosáhnout i výrazně lepších výsledků, k tomu už ale raději využijme silnějších nástrojů, které jsou dostupné v různých programovacích jazycích. Zde použijeme programovací jazyk Python.

## Díl 2, problém 2

**Zadání:**
*Šum v obrázku můžeme redukovat i bez pomoci grafického editoru. Pokusíme se naprogramovat vlastní filtr redukující šum v jazyce Python.*

*V datech k tomuto článku najdete soubory s příponou* `*.npy`*, které obsahují data ke stejným obrázkům jako v minulém díle. Tento soubor opět přečtete stejným způsobem jako v minulém díle, viz následující kód:*

```
import numpy as np
import matplotlib.pyplot as plt

frame = np.load("1.npy")
plt.imshow(frame, interpolation='nearest', cmap='binary')
plt.show()
```

*Projděte postupně všechny pixely načteného obrázku a aplikujte na každý pixel vámi vytvořený filtr:*

```
# Muzete upravit libovolnou cast kodu,
# staci ale, kdyz upravite jen radek 17.
import numpy as np
import matplotlib.pyplot as plt

frame = np.load("1.npy")

fig = plt.figure()
fig.add_subplot(1, 2, 1)
plt.imshow(frame, interpolation='nearest',
           cmap='binary', aspect='auto')

for y in range(frame.shape[0]):
    for x in range(frame.shape[1]):
        # zde vlozte svuj filtr, i jeden radek staci
        # nasledujici radek je priklad threshold filtru
        frame[y][x] = frame[y][x] if frame[y][x]>500 else 0

fig.add_subplot(1, 2, 2)
plt.imshow(frame, interpolation='nearest',
           cmap='binary', aspect='auto')
plt.show()
```

*Vyzkoušejte si, jestli váš filtr funguje na všechny obrázky (nebo aspoň na většinu), které máte k dispozici. Upravte výše uvedený kód v jazyce Python.*

**Řešení:**

V minulé úloze jste si vyzkoušeli filtrování šumu pomocí existujících metod v grafickém editoru. Nyní máme v ruce silnější nástroj, který je ale o něco obtížnější. Pokud jste použili například nástroje z knihovny OpenCV (Open Computer Vision), je to plně v pořádku. V následující ukázce jsem se knihovně OpenCV záměrně vyhnul, abych vám ukázal, jak techniky zpracování obrazu fungují uvnitř.

Možná jste si všimli, že se v kódu objevilo pár konstant. Tato čísla získáme nejčastěji na základě kalibrace. Pokud používáme stále stejný senzor, je bezpečnější stanovit konstanty předem. Pokud bychom chtěli robustnější řešení, můžeme si do kódu připsat dynamickou kalibraci, která kontroluje, jestli se od stanovených konstant příliš neodchylujeme.

V obrázcích výše jsme mohli najít šum ve tvaru písmene T, který byl silnější než naměřená data. V praxi to znamená, že v těchto místech je senzor slepý, neboť šum bude vždy silnější než reálné odrazy. Tento typ šumu můžeme odstranit třeba tak, že v těchto místech přepíšeme naměřené hodnoty na nuly nebo na hodnoty sousedních pixelů, které nejsou tímto typem šumu zatíženy. Tím odstraníme všechny falešné hodnoty.

Další vlastností senzoru je, že odrazy od atmosféry z menších vzdáleností generují silnější šum než ty vzdálenější. Na základě naměřených hodnot můžeme intenzitu tohoto šumu aproximovat rovnicí, kde $k$ je vypočítaná intenzita šumu a $y$ vzdálenost v pixelech:

$$k = \frac{10}{\frac{y+128}{25^2}}$$

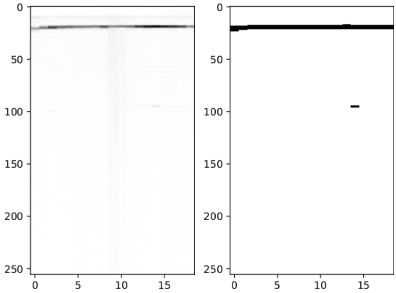
Tento šum následně odstraníme odečtením vypočítané intenzity pro každý řádek ve výsledném obrázku.

Nakonec aplikujeme threshold filtr podobně jako v minulé úloze. Další filtry nyní nechávám stranou, protože si vystačíme i bez nich. Vždy je dobré co nejlépe znát (nebo odhadnout) fyzikální charakteristiky zvoleného senzoru. Tyto charakteristiky nám přímo napoví, jaké techniky zpracování dat v daném případě použít. V našem tématku k tomuto účelu sloužilo celé první číslo.

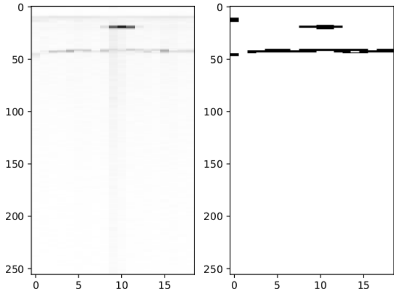
```
import numpy as np
import matplotlib.pyplot as plt

frame = np.load("1_2D.npy")

# noise background value for every row
rowConst = np.zeros(256)
for i in range(1, 256):
    rowConst[i] = 10/((i+128)/25**2)

# show original image
fig = plt.figure()
```

```
fig.add_subplot(1, 2, 1)
plt.imshow(frame, interpolation='nearest',
           cmap='binary', aspect='auto')

# blind areas reduction
frame[:,9] = frame[:,8]
frame[:,10] = frame[:,11]
frame[9,:] = frame[8,:]
frame[10,:] = frame[9,:]
frame[12,:] = frame[13,:]
frame[11,:] = frame[12,:]

# background noise reduction
for y in range(frame.shape[0]):
    frame[y,:] -= rowConst[y]

# negative numbers reduction
for y in range(frame.shape[0]):
    for x in range(frame.shape[1]):
        if frame[y][x] < 0:
            frame[y][x] = 0

# Faster alternative to negative numbers reduction
# frame[frame < 0] = 0

# threshold reduction
for y in range(frame.shape[0]):
    for x in range(frame.shape[1]):
        if frame[y][x]>50:
            frame[y][x] = 1
        else:
            frame[y][x] = 0
        # Another alternative to mentined "if"
        # frame[y][x] = 1 if frame[y][x]>50 else 0

# show result
fig.add_subplot(1, 2, 2)
plt.imshow(frame, interpolation='nearest',
           cmap='binary', aspect='auto')
plt.show()
```

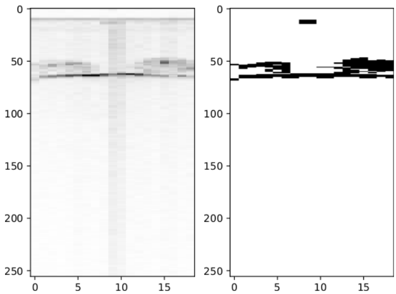
Výsledky filtrování šumu pomocí výše popsaného postupu:



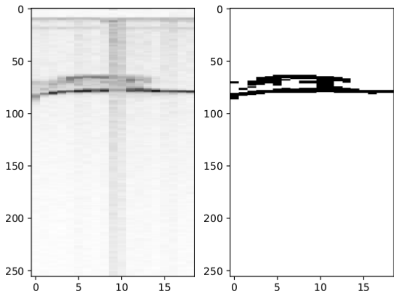

**(a)** Rovná louka, fotbalové hřiště



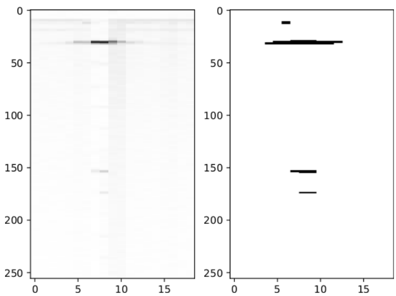

**(b)** Louka z větší výšky s překážkou



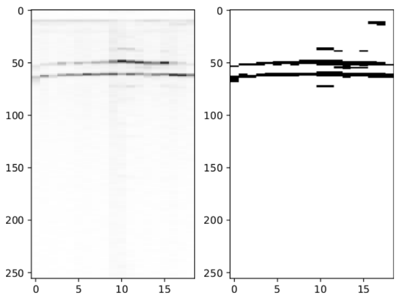

**(c)** Polní cesta lemovaná stromy




**(d)** Rozhraní louky a lesa




**(e)** Vodní hladina



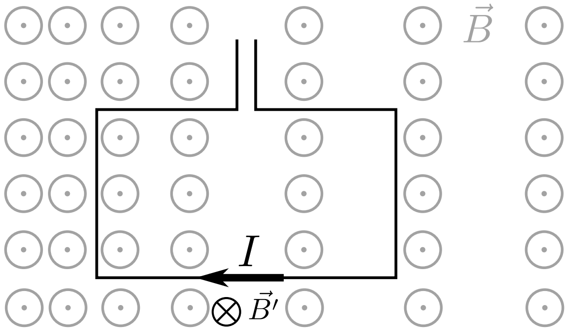

**(f)** Zeď domu

*Béďa;* `bedrich.said@gmail.com`
*e-mailová konference:* `radary@mam.mff.cuni.cz`

# Téma 4 – Vybrané kapitoly z elektromagnetismu

## Díl 4: Faradayův zákon elektromagnetické indukce

Ve druhém díle tohoto tématka jsme se zabývali případem, kdy proud tekoucí vodičem indukuje magnetické pole charakterizované vektorem magnetické indukce, obecně značeným jako $\boldsymbol{B}$. Nyní se podíváme na opačný případ – proměnným vnějším magnetickým polem $\boldsymbol{B}$ vytvoříme elektrické napětí, a tím i tekoucí proud $I$. A není náhodou, že tento proud procházející vodičem nám bude vytvářet indukované magnetické pole, které v tomto textu budeme značit jako $\boldsymbol{B'}$.




**Obrázek 6:** Model smyčky pohybující se v nehomogenním magnetickém poli $\boldsymbol{B}$. Vektor indukované magnetické indukce $\boldsymbol{B'}$ je tečna k indukovaným magnetickým čarám v rovině smyčky a směřuje od čtenáře kolmo na plochu nákresny. Indukované pole je vyvoláno proudem $I$ souvisejícím s indukovaným napětím $\varepsilon_i$.

Příkladem nám může být vodivá smyčka, tedy stočený vodič, který má vzhledem ke svým rozměrům konce blízko u sebe, umístěná v nehomogenním magnetickém poli, viz obrázek 6. Vidíme, že vnější magnetická indukce $\boldsymbol{B}$ (tedy ta, která nám indukuje napětí) směřuje kolmo na rovinu nákresu[12] směrem k vám, čtenářům, a vodič leží v rovině nákresu. Indukované napětí se měří na koncích smyčky. Pozorný čtenář si mohl všimnout, že smyčka je jeden závit cívky.

Pro jednodušší výpočty budeme při popisu následujících modelových situací zachovávat vzájemnou kolmost plochy smyčky a vektorů indukce vnějšího nehomogenního magnetického pole $\boldsymbol{B}$. Pokud by kolmé nebyly, tak bychom museli brát v úvahu jen složku vektoru magnetické indukce, která je kolmá na plochu smyčky. Ploše smyčky přiřazujeme normálový vektor, tj. vektor kolmý na plochu. Směr vektoru udáváme pomocí pravidla pravé ruky: prsty pravé ruky obtočíme kolem smyčky tak, že palec ukazuje proti směru vektoru indukce $\boldsymbol{B}$.

Vygenerované indukované napětí, které lze obecně spočítat jako záporně vzatou změnu magnetického indukčního toku v čase, matematicky zapíšeme pomocí

[12] Takto se běžně značí vektory kolmé k nákresně. Pokud by vektor nesměřoval k vám, ale od vás do nákresny, značil by se křížkem v kolečku.

Faradayova zákona elektromagnetické indukce jako

$$\varepsilon_i = -\frac{\mathrm{d}\Phi}{\mathrm{d}t},$$

kde $\varepsilon_i$ je indukované napětí vytvářející magnetické pole. Znaménko mínus značí, že indukované napětí se snaží působit proti změně, která jej vyvolala (lepší odůvodnění dále v textu). Souvisí tedy pouze se směrem, ne s velikostí elektrické intenzity (resp. poklesu napětí). Neznámá $\Phi$ značí onen magnetický indukční tok, jehož význam můžeme zjednodušeně interpretovat jako „počet indukčních čar procházejících plochou smyčky“. Vypočte se podle vzorce $\Phi = \oint_S \boldsymbol{B}\mathrm{d}\boldsymbol{S}$.

Vztah pro výpočet magnetického indukčního toku $\Phi$ obsahuje vektorové veličiny, což znamená, že umíme určit i výsledný směr proudu tekoucího cívkou, a to podle polarity indukovaného napětí a pravidla pravé ruky. V zásadě máme dvě možnosti směru toku proudu smyčkou. Správné řešení je na obrázku 6. Vzpomeneme si na pravidlo pravé ruky, které se aplikovalo na magnetickou indukci v okolí vodiče: palec přiložíme k vodiči ve směru proudu tak, že prsty obtočené kolem vodiče ukazují směr indukovaných magnetických siločar, jejichž tečna je vektor indukované magnetické indukce $\boldsymbol{B'}$ v daném bodě.

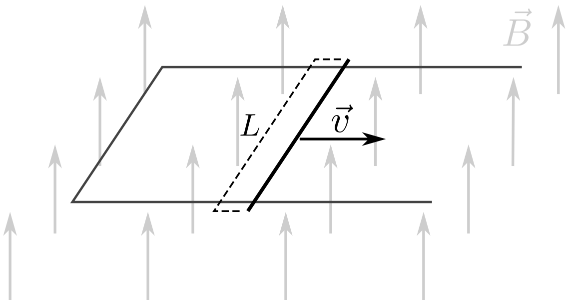
Problém je, že neznáme směr proudu. Víme však, jaký směr mají mít vektory magnetické indukce $\boldsymbol{B'}$. Chceme totiž, aby směr indukované magnetické indukce byl opačný oproti směru vektorů té vnější, značené $\boldsymbol{B}$. Tudíž si pravidlo pravé ruky „otočíme“ – pravou ruku kolem vodiče obtočíme tak, aby prsty směřovaly v opačném směru než mají vektory vnější magnetické indukce $\boldsymbol{B}$, tedy ve směru indukce $\boldsymbol{B'}$. Vztyčený palec nám tak ukáže směr proudu vyvolaného indukovaným napětím $\epsilon_i$.

Kdyby vektory magnetické indukce $\boldsymbol{B'}$ měly stejný směr jako vektory magnetické indukce vnějšího pole $\boldsymbol{B}$, oba vektory by se sečetly a výsledná magnetická indukce by vycházela větší než ta původní, což by porušovalo zákon zachování energie. Proto musí pole $\boldsymbol{B}$ a $\boldsymbol{B'}$ směřovat opačně.

**Problém 1** [4b]: *Máme vodič zanedbatelné tloušťky a z něj zformujeme kruhovou cívku se 150 závity o poloměru 2 cm. Na konce drátu přiložíme voltmetr a cívku vložíme do vnějšího magnetického pole. Velikost vektoru magnetické indukce zvyšujeme s časem podle rovnice* $B(t) = 12t - 0{,}003$ *(jednotka mT). Směr vektorů magnetické indukce je konstantní – svisle dolů. Rovina plochy smyčky je natočená oproti vodorovnému směru o* $30°$. *Vypočtěte, jaké napětí bychom naměřili na koncích drátu v tomto stavu.*

Podíváme se na další případ, kdy máme homogenní magnetické pole neměnné v čase. Napětí lze indukovat jen v případě, že měníme velikost plochy smyčky (viz definice magnetického indukčního toku).

Jak taková situace vypadá? Podívejme se na návrh situace na obrázku 7. Máme z vodiče vytvořený obdélník, jehož jedna hrana o délce $L$ je pohyblivá – posouvá se v naznačeném směru konstantní rychlostí $v$ a zvětšuje tím plochu obdélníku tvořeného vodičem.

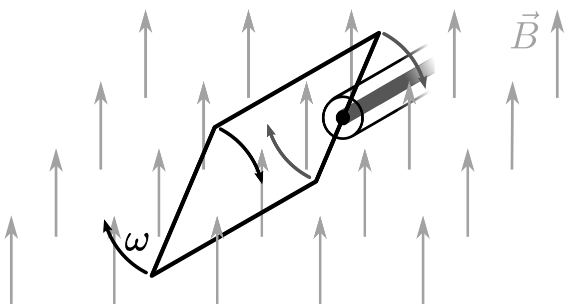



**Obrázek 7:** Modelová situace pro indukci napětí v homogenním vnějším magnetickém poli. Pohyb vodičem o délce $L$ bude měnit velikost plochy smyčky. Plocha smyčky je kolmá na vektory magnetické indukce vnějšího magnetického pole.

**Problém 2** [4b]: *Napište, jak byste vypočetli indukované napětí a jaký směr bude mít výsledný proud v situaci na obrázku 7.*

Nyní se podíváme na jednu z aplikací Faradayova zákona: generování střídavého napětí. Máme otáčející se smyčku v homogenním magnetickém poli, jejíž osa rotace je kolmá na svislé magnetické pole, viz obrázek 8. Víme, že smyčka rotuje s úhlovou rychlostí $\omega$ (roztáčí ji například lopatky větrné turbíny). Tím generujeme napětí.

**Problém 3** [4b]: *Na vás je zamyslet se, jaký bude časový průběh tohoto napětí při konstantní velikosti veličin $B$ a $\omega$. Plocha smyčky má povrch $S$. Vyneste své závěry do grafu závislosti generovaného napětí na čase. Porovnejte s průběhem velikosti složky magnetické indukce, která se podílí na indukci napětí. (Nápověda: Rozmyslete si, kam budou směřovat vektory normály plochy a vektor magnetické indukce $\boldsymbol{B}$.)*



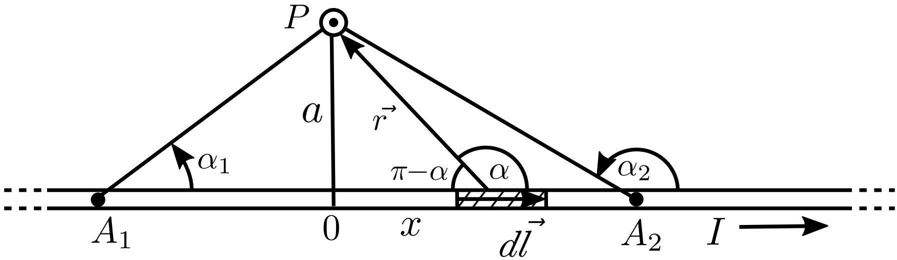

**Obrázek 8:** Rotující smyčka v magnetickém poli.

# Řešení 2. série

## Problém 1

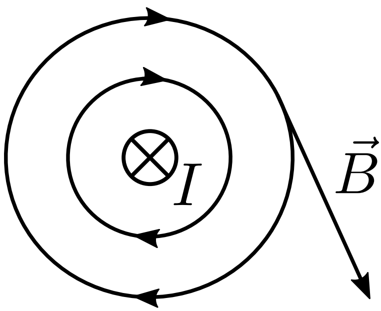
**Zadání:**
*Určete směr vektoru magnetické indukce v nekonečně dlouhém vodiči v příkladu výše. Jaký tvar budou mít magnetické siločáry?*

**Řešení:**
Součin vektorů d$\boldsymbol{l}$ a $\boldsymbol{r}$ určíme pomocí pravidla pravé ruky: prsty pravé ruky nám směrují ve směru vektoru d$\boldsymbol{l}$ tak, že vektor $\boldsymbol{r}$ nám vystupuje z dlaně. Vztyčený palec nám směřuje ve směru vektoru magnetické indukce $\boldsymbol{B}$. Víme, že vektor směřuje kolmo z nákresny směrem k nám, viz obrázek 9.




**Obrázek 9:** Směr magnetické indukce v bodě P v okolí nekonečně dlouhého vodiče.

Vektor B je tečnou k magnetické siločáře, tudíž pokud aplikujeme pravidlo pravé ruky na více bodů v okolí dlouhého vodiče, lze odhadnout, že indukční čáry magnetického pole jsou soustředné kružnice, viz 10.

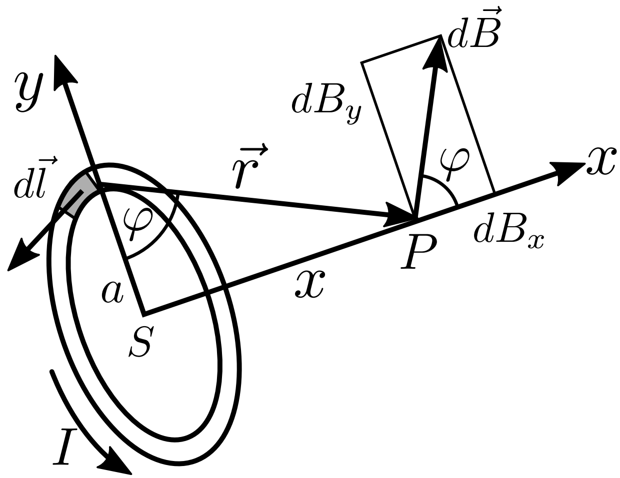



**Obrázek 10:** Magnetické indukční čáry v okolí nekonečně dlouhého vodiče.

Problém 2

**Zadání:**
*Napište Biot-Savartův zákon pro magnetickou indukci nacházející se v ose cívky, která má jeden závit a jejíž osa je kolmá na plochu závitu. Odhadněte, jak budou vypadat magnetické siločáry v ose a v rovině ploch cívky. (Nápověda: Rozložte si náhodně orientovaný vektor* $\boldsymbol{B}$ *do dvou kolmých směrů, z nichž jeden bude rovnoběžný s plochou cívky. Integrujte přes* $\mathrm{d}\boldsymbol{l}$ *po uzavřené křivce, tedy smyčce cívky.)*

**Řešení:**
Na ose smyčky najdeme obecný bod P. V tomto bodě popisuje indukční čáry magnetického pole obecný vektor magnetické indukce $\boldsymbol{B}$. Využijeme nápovědu ze zadání a rozložíme vektor $\boldsymbol{B}$ do dvou kolmých směrů tak, že jeden z těchto směrů bude rovnoběžný s rovinou plochy smyčky, viz obrázek 11.




**Obrázek 11:** Magnetická indukce v bodě P na ose smyčky. Poloha bodu P je charakterizována polohovým vektorem $\boldsymbol{r}$ v souřadném systému os $x$ a $y$. Vektor $\boldsymbol{B}$ je rozložen do obou kolmých směrů.

Z obrázku můžeme vyčíst, že každý velmi malý úsek smyčky (označen $\mathrm{d}\boldsymbol{l}$) generuje v bodě P malý příspěvek celkové magnetické indukci, který označíme $\mathrm{d}\boldsymbol{B}$. Složky těchto příspěvků $\mathrm{d}\boldsymbol{B}$ ve směru osy $y$ se pro všechny malé části smyčky $\mathrm{d}\boldsymbol{l}$ vyruší. Tím pádem nás zajímá jenom $x$-ová složka tohoto příspěvku magnetické indukce. Označme si jej $\mathrm{d}\boldsymbol{B}_x$. Z obrázku určíme, že $\mathrm{d}\boldsymbol{B}_x = \mathrm{d}\boldsymbol{B}\cos\varphi$ a z podobnosti trojúhelníku určíme, že $\cos\varphi = \frac{a}{x^2+a^2}$, kde $a$ je poloměr smyčky a $x$ je kolmá vzdálenost bodu P od středu plochy smyčky. Aplikujeme Biot-Savartův zákon a použijeme substituci za $\cos\varphi$:

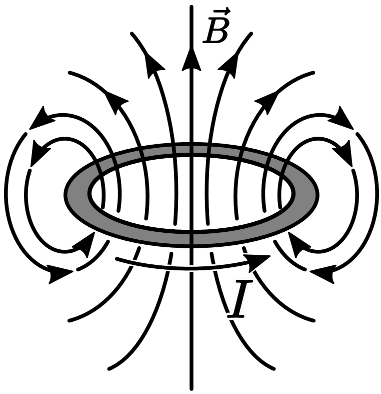
$$B = \oint \mathrm{d}B_x$$

$$\oint \mathrm{d}B \cos\varphi = \oint \frac{\mu_0}{4\pi} \frac{I \mathrm{d}l}{x^2 + a^2} \frac{a}{(x^2 + a^2)^{\frac{1}{2}}} = \frac{\mu_0}{4\pi} \frac{Ia}{(x^2 + a^2)^{\frac{3}{2}}} \oint \mathrm{d}l = \frac{\mu_0}{4\pi} \frac{I \cdot 2\pi a^2}{(x^2 + a^2)^{\frac{3}{2}}}$$

Výsledný tvar pro výpočet magnetické indukce v ose smyčky je

$$B - \frac{\mu_0}{4\pi} \frac{I \cdot 2\pi a^2}{(x^2 + a^2)^{\frac{3}{2}}}$$

Tvar magnetických indukčních čar lze odhadnout, když se na smyčku podíváme jako na zkroucený vodič. Magnetické indukční čáry budou tedy smyčku „obtáčet“, viz obrázek 12.




**Obrázek 12:** Nákres magnetických indukčních čar. Smyčkou protéká proud $I$ ve vyznačeném směru.

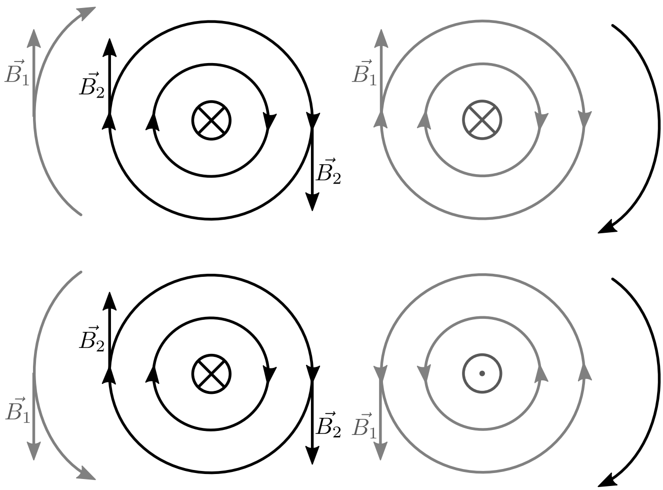
## Problém 3

**Zadání:**
*Pokud dáme vedle sebe dva nekonečně dlouhé rovnoběžné vodiče, kterými protéká proud, začnou spolu interagovat díky generované magnetické indukci. Popište, jak bude vypadat výsledná magnetická indukce v prostoru kolem vodičů, pokud proudy ve vodičích potečou stejným směrem a pokud potečou opačným.*

**Řešení:**
Opět aplikujeme pravidlo pravé ruky postupně na oba vodiče na obrázku 13. První dvojice v horní polovině obrázku reprezentuje dva vodiče, v nichž tečou proudy stejným směrem (směrem do vás, čtenářů), tudíž magnetické indukční čáry budou „obíhat“ vodič shodně. Nicméně v prostoru mezi vodiči je směr magnetických indukčních čar opačný, a tak bude velikost výsledné magnetické indukce menší než

hodnota $\boldsymbol{B}_1$ nebo $\boldsymbol{B}_2$. Pro případ, kdy budou proudy protékat různými směry, se vektory magnetické indukce v prostoru mezi vodiči sečtou.

V prostoru vně od dvojice vodičů budou magnetické indukční čáry (a tedy i vektory magnetické indukce) polí obou vodičů směřovat opačně. V případě vodičů, kterými teče proud stejným směrem, se jednotlivé vektory magnetických indukcí obou vodičů budou sčítat. Pro případ dvou vodičů s proudy tekoucími opačným směrem se budou odčítat.




**Obrázek 13:** Nákres dvojic nekonečně dlouhých rovnoběžných vodičů kolmých na nákresnu. První případ popisuje situaci, kde proud v obou vodičích teče stejnými směry a vektory magnetických indukcí polí $\boldsymbol{B}_1$ a $\boldsymbol{B}_2$ od obou vodičů se mezi nimi odečítají. Druhý případ ukazuje opak – proudy tečou opačnými směry, vektory indukce se sčítají. V prostoru vpravo a vlevo od vodičů vektory magnetické indukce budou interagovat přesně opačně, budou se sčítat pro stejný směr proudů a odčítat pro opačný.

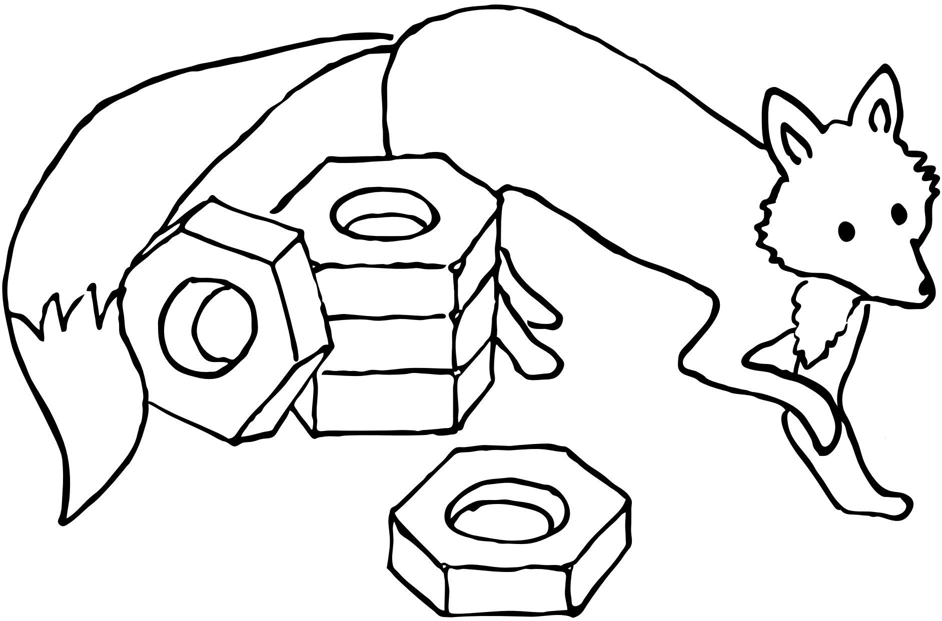
*Pája, Fanda;* `fandazajic@gmail.com`
*e-mailová konference:* `elmag@mam.mff.cuni.cz`

# Výsledková listina 2. čísla

| **Poř.** | **Jméno** | **R.** | $\sum_{-1}$ | **Úlohy** t1 | t2 | t4 | turnaj | $\sum_0$ | $\sum_1$ |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Dr.$^{\mathrm{MM}}$ K. Vokálová | 4 | 69,7 | 6,1 | 13,0 | 7,0 | 10,0 | 36,1 | 57,3 |
| 2. | Dr.$^{\mathrm{MM}}$ J. Kalvoda | 3 | 51,7 | | 44,5 | | 2,2 | 46,7 | 51,7 |
| 3. | Mgr.$^{\mathrm{MM}}$ V. Janáček | 3 | 34,2 | 7,0 | 14,5 | | 2,2 | 23,7 | 34,2 |
| 4. | Mgr.$^{\mathrm{MM}}$ M. Fof | 2 | 31,2 | 6,4 | 10,1 | | 5,0 | 21,5 | 31,2 |
| 5. | Doc.$^{\mathrm{MM}}$ J. Havelka | 4 | 129,4 | | | | | 0 | 25,0 |
| 6. | Mgr.$^{\mathrm{MM}}$ J. Knillová | 1 | 20,1 | | | | | 0 | 20,1 |

| Poř. | Jméno | R. | $\sum_{-1}$ | Úlohy t1 | t2 | t4 | turnaj | $\sum_0$ | $\sum_1$ |
|---|---|---|---|---|---|---|---|---|---|
| 7. | Dr.MM K. Hloušková | 4 | 50,5 | | 2,1 | | | 2,1 | 20,0 |
| 8. | Mgr.MM J. Piroutek | 4 | 33,1 | 5,0 | | 7,0 | | 12,0 | 19,9 |
| 9. | Bc.MM K. Pernicová | 3 | 18,7 | 5,3 | | | 1,0 | 6,3 | 18,7 |
| 10. | Mgr.MM T. Flídr | 2 | 43,9 | 4,4 | | | | 4,4 | 17,9 |
| 11. | Bc.MM O. Piroutek | 2 | 17,6 | 4,6 | | 3,0 | | 7,6 | 17,6 |
| 12. | Mgr.MM L. Kunčarová | 4 | 20,6 | | | | | 0 | 17,0 |
| 13. | Bc.MM J. Kaifer | 4 | 19,8 | | | | | 0 | 16,0 |
| 14. | Dr.MM M. Kalousková | 4 | 65,8 | 3,5 | 4,0 | | | 7,5 | 15,5 |
| 15. | Bc.MM D. Perout | 3 | 15,0 | 2,2 | | | | 2,2 | 15,0 |
| 16. | Mgr.MM E. Vítková | 4 | 45,6 | | | | | 0 | 14,3 |
| 17. | Mgr.MM O. Gonzor | 3 | 42,7 | | | | | 0 | 12,8 |
| 18. | Bc.MM V. Starosta | 2 | 12,3 | 5,3 | | | 7,0 | 12,3 | 12,3 |
| 19. | Bc.MM J. Kvapil | 2 | 15,7 | | | | | 0 | 12,0 |
| 20. | Mgr.MM O. Chlubna | 3 | 46,7 | 3,9 | | | | 3,9 | 11,9 |
| 21. | Bc.MM A. Opl | 2 | 11,5 | | | | | 0 | 11,5 |
| 22. | Mgr.MM B. Kopčák | 4 | 28,6 | | | | | 0 | 11,1 |
| 23. | A. Žáčková | 3 | 9,5 | | | | | 0 | 9,5 |
| 24. | Bc.MM L. Vomelová | 4 | 15,3 | | | | | 0 | 8,2 |
| 25. | Bc.MM V. Žák | 4 | 11,6 | | | | | 0 | 8,0 |
| 26. | Mgr.MM M. Boček | 1 | 29,2 | | | | | 0 | 7,8 |
| 27. | Mgr.MM M. Holubička | 4 | 44,4 | 0,1 | | | | 0,1 | 6,6 |
| 28. | J. Bláhová | 3 | 6,5 | | | | | 0 | 6,5 |
| 29. | J. Jedlička | 2 | 5,7 | | | | | 0 | 5,7 |
| 30. | Bc.MM E. Neumannová | 3 | 15,3 | | | | | 0 | 5,2 |
| 31. | M. Bučková | 3 | 9,9 | | | | | 0 | 5,0 |
| 32. | A. Cmielová | 1 | 3,6 | 3,6 | | | | 3,6 | 3,6 |
| 33. | Bc.MM F. Bujnovský | 2 | 12,2 | | | | | 0 | 3,4 |
| 34. | M. Turinská | 3 | 3,0 | | | | | 0 | 2,0 |
| 35. | Bc.MM M. Vícha | 1 | 12,4 | | | | | 0 | 1,4 |

Sloupeček $\sum_{-1}$ je součet všech bodů získaných v našem semináři, $\sum_0$ je součet bodů v aktuální sérii a $\sum_1$ součet všech bodů v tomto ročníku. Tituly uvedené v předchozím textu slouží pouze pro účely M&M.





## Kontakty:

M&M, OPMK, MFF UK
Ke Karlovu 3
121 16 Praha 2

E-mail: `mam@matfyz.cz`
Web: `mam.matfyz.cz`
FB: `casopis.MaM`