

DANIELA PITTNEROVÁ A KOLEKTIV

# Fyzikální korespondenční seminář

XXXI. ročník – 2017/18



Copyright © Daniela Pittnerová, 2018  
Copyright © MatfyzPress, nakladatelství Matematicko-fyzikální fakulty  
Univerzity Karlovy, 2018

ISBN 978-80-7378-370-9



## Obsah

<b>Předmluva</b>	<b>4</b>
<b>Zadání teoretických úloh</b>	<b>7</b>
<b>Řešení teoretických úloh</b>	<b>14</b>
<b>Zadání experimentálních úloh</b>	<b>104</b>
<b>Řešení experimentálních úloh</b>	<b>106</b>
<b>Numerické metody a počítačové simulace</b>	<b>134</b>
Kapitola 1: Rozjezdová . . . . .	134
Kapitola 2: Derivace a Monte Carlo integrace . . . . .	152
Kapitola 3: Na procházce s integrály . . . . .	167
Kapitola 4: Kořeni a automati . . . . .	183
Kapitola 5: Rostou nám diferenciální rovnice . . . . .	194
Kapitola 6: Matice a populace . . . . .	211
<b>Řešení úloh ze seriálu</b>	<b>230</b>
<b>Akce FYKOSu</b>	<b>283</b>
Podzimní soustředění v Zelené Lhotě . . . . .	283
Jarní soustředění v Zásece . . . . .	284
FYKOSí Fyziklání . . . . .	286
Fyziklání online . . . . .	288
Fyzikální Náboj . . . . .	289
Týden s aplikovanou fyzikou . . . . .	290
Den s experimentální fyzikou . . . . .	292
Víkend s aplikovanou fyzikou . . . . .	293
Cyklus přednášek pro středoškoláky . . . . .	293
<b>Pořadí řešitelů</b>	<b>294</b>



## Předmluva

Milá čtenářko, milý čtenáři!

Do rukou se Ti dostala publikace, která shrnuje veškerou činnost Fyzikálního korespondenčního semináře Matematicko-fyzikální fakulty Univerzity Karlovy v jeho XXXI. ročníku, který probíhal ve školním roce 2017/18.

FYKOS je nejstarší a také největší fyzikálně zaměřenou korespondenční soutěží pro žáky středních škol v České republice. Je organizován studenty především Matematicko-fyzikální fakulty UK, ale i jiných fakult a škol v České republice i zahraničí, a podporován zaměstnanci Matematicko-fyzikální fakulty UK. Snaží se oslovit studenty se zájmem o fyziku, matematiku, techniku, zkrátka svět kolem nás. Naším cílem je rozvíjet talent a fyzikální myšlení, protože věříme, že člověk, který se umí zastavit a zamyslet (nejen nad fyzikálními problémy) a cítí touhu dobrat se řešení, se v životě vždy velmi dobře uplatní.

Během školního roku každý z řešitelů obdrží celkem sedm brožur, v nichž nalezneme šest sérií po osmi úlohách, z nichž dvě jsou „jednoduché“ (zaměřené především na první dva ročníky středních škol), jedna více problémová, jedna experimentální a jedna tzv. seriálová. Zbýlé tři úlohy se týkají libovolného fyzikálního tématu a tvoří jádro série. Zadávané úlohy však nejsou příliš podobné těm, které znáte z hodin fyziky. Vyžadují mnohdy poněkud hlubší úvahu, trochu důvtipu nebo něco z vyšší matematiky. Nezřídka je třeba zapátrat na internetu nebo v odborné literatuře. Účastníci si mohou vybrat, které úlohy nakonec vypracují a pošlou nám k opravení, ať už klasicky poštou, nebo přes internet. Opravující pak jejich řešení okomentují a vysvětlí případné chyby. To vše pošleme zpět řešitelům, včetně výsledkových listin, kde se každý může podívat, jak obstál v konkurenci svých vrstevníků. Na konci ročníku jsou nejlepší řešitelé náležitě odměněni.

Mimo samotnou korespondenční soutěž pro řešitele připravujeme i další akce. Bezesporu nejpoblábnějšími jsou dvě týdenní soustředění v některém z malebných koutů české země. Jejich účastníci si užijí bohatý program, založený na dopoledních přednáškách z oblasti matematiky, fyziky nebo jiných přírodních věd a odpoledních aktivitách v přírodě. Nechybí ani prostor pro fyzikální experimenty a výlety na atraktivní místa. Tento akademický rok se podzimní soustředění konalo v Zelené Lhotě a jarní soustředění v Zásce.

Další akcí je Den s experimentální fyzikou, na kterém se spolupodílejí jednotlivé katedry MFF, ale i pracoviště Akademie věd ČR, resp. Ústav jaderného výzkumu v Řeži. Naším řešitelům tak umožňujeme navštívit velmi zajímavá výzkumná pracoviště, kde se dělá opravdová fyzika.

Nepravidelně organizujeme akce Týden s aplikovanou fyzikou (TSAF). Představuje týden strávený prohlídkami vědeckých center, muzeí i různých firem zaměstnávajících fyziky v České republice i v zahraničí. Letos proběhl TSAF v Praze a okolí.

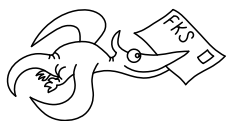
Proběhl také již dvanáctý ročník tradičního FYKOSího Fyziklání, tříhodinové soutěže pětičlenných týmů v řešení fyzikálních úloh. Vyhrává tým s největším bodovým ziskem, přičemž rozhoduje jak správnost, tak i rychlost. V letošním roce se soutěže účastnilo 135 družstev z ČR, Slovenska a poprvé i ze zahraničí – Makedonie, Španělska a Lotyšska. To je pro nás skvělým důkazem, že se fyzika a přírodní vědy obecně mezi středoškolskými studenty stále těší popularitě.

FYKOSím Fyzikláním je inspirovaná internetová soutěž Fyziklání online (v zahraniční verzi Online Physics Brawl). V listopadu 2017 proběhl její sedmý ročník a opět zaznamenala velký úspěch. Soutěž byla své elektronické formě otevřena všem zájemcům, nejenom středoškolákům, pro které je vytvořena především. Tohoto ročníku se zúčastnilo 220 středoškolských a 39 jiných týmů celkově z 33 zemí. To svědčí o narůstající popularitě soutěže, která se již řadí mezi tradiční FYKOSí akce.

Kromě toho FYKOS organizuje i další menší akce, o nichž se dozvíte dále v této publikaci nebo na našich webových stránkách.

Tato ročenka obsahuje kompletní zadání i řešení jednotlivých úloh XXXI. ročníku FYKOSu. Zadání jsou záměrně oddělena od řešení, abychom podnítili čtenáře k samostatnému zamyšlení nad možným řešením problému. Příklady jsou navíc pro snazší orientaci rozděleny na teoretické a experimentální. Další částí knihy je Seriál o numerických metodách a počítačových simulacích, který je rovněž doplněn úlohami. Na konci publikace se nachází krátké ohlédnutí za letošními soustředěními a jinými akcemi a seznam nejlepších řešitelů ročníku.

Pokud Tě FYKOS zaujal natolik, že by ses chtěl stát řešitelem nebo se pouze na něco zeptat, ať už se to týká fyziky, či studia na MFF, neváhej a napiš nám.



**FYKOS**  
**UK, Matematicko-fyzikální fakulta**  
**Ústav teoretické fyziky**  
**V Holešovičkách 2**  
**180 00 Praha 8**

www: <http://fykos.cz>  
 e-mail: [fykos@fykos.cz](mailto:fykos@fykos.cz)

A jak vypadal XXXI. ročník očima statistiků? FYKOS řešilo 109 studentů ze 47 středních škol z České a Slovenské republiky. Přehled škol podle úspěšnosti jejich studentů uvádíme níže.

Pořadí škol

	škola	řešitelů	body	modif. body
1.	G Brno, tř. Kpt. Jaroše	5	694	<b>605</b>
2.	G Jana Keplera, Praha	3	395	<b>382</b>
3.	G J. Ortena, Kutná Hora	1	329	<b>329</b>
4.	G, Olomouc-Hejčín	6	368	<b>322</b>
5.	G Komenského, Havířov	1	306	<b>306</b>
6.	G Velká okružná, Žilina	2	306	<b>303</b>
7.	G, Pelhřimov	2	284	<b>280</b>
8.	G, Jihlava	2	279	<b>278</b>
9.	G PdC, Piešťany	1	263	<b>263</b>
10.	G, Havlíčkův Brod	1	257	<b>257</b>
11.	Klasické a španělské G, Brno	1	249	<b>249</b>
12.	G, Třeboň	1	247	<b>247</b>
13.	G J. Heyrovského, Praha	2	232	<b>230</b>
14.	G M. Hattalu, Trstená	1	214	<b>214</b>
15.	G Nad Štolou, Praha	1	208	<b>208</b>
16.	G V. P. Tótha, Martin	1	198	<b>198</b>
17.	G, Litoměřická, Praha	1	194	<b>194</b>
18.	G, Šumperk	1	184	<b>184</b>
19.	Katolické gymnázium Třebíč	1	180	<b>180</b>
20.	G, nám. TGM, Zlín	2	177	<b>176</b>
21.	G Z. Wintra, Rakovník	1	175	<b>175</b>
22.	ESŠ, Lip. Mikuláš	1	171	<b>171</b>
23.	G, Dačice	1	163	<b>163</b>
24.	1. súkromné G v Bratislave	1	149	<b>149</b>
25.	G Jiřovcova, České Budějovice	1	148	<b>148</b>
26.	G J. Ressela, Chrudim	1	147	<b>147</b>
27.	G, Uherské Hradiště	2	142	<b>142</b>
28.	Matiční G, Ostrava	1	134	<b>134</b>
29.	G, Vídeňská, Brno	1	128	<b>128</b>
30.	G Poštová, Košice	2	120	<b>115</b>
31.–32.	G, Moravský Krumlov	1	114	<b>114</b>
31.–32.	G dr. V. Šmejkal, Ústí n. L.	1	114	<b>114</b>
33.	G, Písek	2	116	<b>113</b>
34.	G, Tanvald	1	109	<b>109</b>
35.	G F. Palackého, Val. Mez.	3	106	<b>102</b>
36.	G, Šternberk	1	90	<b>90</b>
37.	Slovanské G, Olomouc	1	87	<b>87</b>
38.	G, Trutnov	2	88	<b>86</b>
39.–40.	Biskupské G, Brno	1	81	<b>81</b>
39.–40.	G Jura Hronca, Bratislava	2	82	<b>81</b>
41.	G, Nymburk	2	82	<b>79</b>
42.	G Grösslingova, Bratislava	1	76	<b>76</b>



## Zadání teoretických úloh

### Úloha I.1 ... kávu si omléčním

3 body

Kdy je nejhodnější nalít do horké kávy chladné mléko, abychom ji mohli pít co nejdříve? Nepožadujeme přesný výpočet, ale podrobný slovní popis toho, jak káva chladne a jak byste postupovali. *(řešení str. 14)*

### Úloha I.2 ... zálohovací NAS(A)

3 body

Uvažujte optický switch (propustnost  $10 \text{ Gb}\cdot\text{s}^{-1}$ ), jehož výstup (po patřičném zesílení) použijete k ozáření Měsíce. Díky zrcátkům zanechaným na jeho povrchu z dob projektu Apollo se signál vrátí zpět a přivedete jej (po patřičném zesílení) na vstup switchu. Pokud zajistíme spolehlivé fungování switchu, budou jednou vyslaná data v systému „obíhat“ trvale, takže jsme získali paměť. Jaká je její maximální kapacita? Dobu zpracování ve switchi a velikost datových hlaviček zanedbejte. *(řešení str. 15)*

### Úloha I.3 ... oběšený úhelník

6 bodů

Máme homogenní úhelník ve tvaru L o stranách délek  $b, c$ . Je volně zavěšen v železničním vagoně za konec jedné strany tak, že jeho vrchol míří ve směru jízdy vagonu. S jakým zrychlením  $a$  se musí vagon pohybovat, aby spodní strana úhelníku byla rovnoběžná se směrem jízdy? Relativistické jevy neuvažujte.

*Bonus:* Relativistické jevy uvažujte.

*(řešení str. 16)*

### Úloha I.4 ... praská mi v láhvi

7 bodů

Co když si skoro prázdnou 1,5 litrovou PET láhev uzavřeme v dobře vytápěné kanceláři, dejme tomu na  $t_k = 26 \text{ }^\circ\text{C}$ , a pak vyjdeme vstříc novým zážitkům dolů ze schodů? Láhev začne praskat. Co má větší vliv? To, že se mění atmosférický tlak, jak scházíme 10 pater v budově, nebo to, že je na schodech, dejme tomu,  $t_s = 15 \text{ }^\circ\text{C}$ ? *(řešení str. 18)*

### Úloha I.5 ... planetární osidlování

7 bodů

Nejspíše jste již někdy přemýšleli o tom, jestli neexistují nějaké mimozemské civilizace. Zpravidla čím větší hvězda je, tím větší má zářivý výkon a tím kratší má také svůj život. Zaměřme se nyní na to, že máme dvě hvězdy, z nichž jedna má dvojnásobný zářivý výkon co druhá. Pokud je pásmo, ve kterém je možný život, dáno teplotou, na které by se ustálilo dokonale černé těleso, a určitými dvěma teplotami (stejně pro jakoukoliv soustavu), kolem které hvězdy je širší pásmo, ve

kterém by mohla být planeta se životem? Kolikrát bude větší oproti druhé hvězdě?  
*(řešení str. 22)*

### Úloha I.P ... modýlek letadla na ISS 10 bodů

Jak by se chovalo letadlo v mikrogravitaci (prostě uvažujte, že na něj gravitační síla nepůsobí)? Popište, jaký efekt by měla směrovka, výškovka, křídélka, případně vektorování tahu motorů. Jaké akrobatické manévry by byly možné? (Například plochá vývrtka asi ne.)  
*(řešení str. 23)*

### Úloha II.1 ... Zuběnka 3 body

Jak velké skladovací prostory by musela mít Víla Zubnička, aby mohla skladovat všechny mléčné zuby všech dětí? Resp. jakým tempem by její nároky na uskladnění rostly? Za jakou dobu by teoreticky měla ve svých skladech většinu zásob fosforu na Zemi?  
*(řešení str. 24)*

### Úloha II.2 ... irradie solární elektrárny 3 body

Solární konstanta, či správněji solární irradie, je tok energie přicházející ze Slunce ve vzdálenosti Země od Slunce. Nejde o konstantu, ale uvažujme, že má hodnotu  $P = 1\,370\text{ W}\cdot\text{m}^{-2}$ . Uvažujme, že Země obíhá Slunce po kružnici a sklon zemské osy vůči kolmici k její oběžné rovině je  $23,5^\circ$ . Jaký bude maximální výkon zachycený solárním panelem o ploše  $S = 1\text{ m}^2$  o letním a zimním slunovratu, pokud panel leží na rovném povrchu Země v Praze? Uvažujte, že ani atmosféra ani budovy nijak neovlivní měření.  
*(řešení str. 26)*

### Úloha II.3 ... pozorovací 6 bodů

Jakou část povrchu kulové planety není možné vidět ze stacionární oběžné dráhy planety (taková dráha, že se obíhající objekt nachází stále nad stejným bodem na planetě), která má hustotu  $\rho$  a periodu rotace  $T$ ?  
*(řešení str. 26)*

### Úloha II.4 ... jaderný odpad nikdy více 6 bodů

Představme si, že máme něco (například jaderný odpad) a chceme se toho zbavit. Těleso dostaneme na oběžnou dráhu Slunce shodnou s oběžnou dráhou Země, ale dostatečně daleko od Země, abychom mohli gravitační působení Země nadále zanedbávat. Otázka je, jaký způsob zbavení se inkriminovaného předmětu by nás stál kolik energie a který postup by byl tedy nejvýhodnější. Varianty jsou

- Hodit to do Slunce. Stačí, aby se to dostalo na sluneční povrch a bude to dostatečně usmažené.
- Převést to na kruhovou dráhu v Hlavním pásu (pás planetek mezi Marsem a Jupiterem).
- Vyhodit to zcela ze Sluneční soustavy.

*(řešení str. 28)*



**Úloha II.5 ... skleněný déšť**

7 bodů

Dělník si na stavbu mrakodrapu přinesl vak se skleněnkami, aby se s nimi mohl pochlubit svým kolegům. A co se nestane – vak se vysype a kuličky padají skrze lešení směrem k zemi. Lešení se skládá z jednotlivých poschodí o výšce  $h$ . Podlaha každého poschodí se skládá ze stejných mříží, ve kterých díry zaujímají  $k\%$  z celkové plochy mříže. Uvažujme zjednodušený model propadávání kuliček lešením, kdy, pokud kulička spadne na díru v lešení, tak projde bez ovlivnění, a pokud spadne na pevnou část mříže, tak se její rychlost sníží na 0 a ihned začne dále padat (tj. velikost kuliček je zanedbatelná vůči velikosti děr v lešení, kuličky se od lešení nijak neodráží a po dopadu na pevnou část mříže se ihned skutálí do díry a dále začínají padat). Nakonec neuvažujme ani potenciální srážky kuliček mezi sebou. Předpokládejte, že kuličky se z tašky sypou s konstantním hmotnostním průtokem  $Q$ . Jakou silou budou kuličky působit na každé patro lešení, až se situace ustálí? (řešení str. 36)

**Úloha II.P ... ó Oganesson**

10 bodů

Jaké vlastnosti má 118. prvek periodické soustavy prvků? Respektive jaké by asi měl, kdyby byl stabilní? Diskutujte alespoň tři fyzikální vlastnosti. (řešení str. 40)

**Úloha III.1 ... zpomalená**

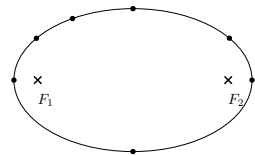
3 body

Představme si, že na kameru se snímkovou frekvencí 24 snímků za sekundu (uvažujeme časově rovnoměrně rozložené a dokonale ostré snímky) natočíme let vrtulníku s otáčkami hlavního rotoru 2900 ot./min. Následně si záznam přehrajeme. Jaká bude zdánlivá frekvence otáček rotoru na záznamu? (řešení str. 47)

**Úloha III.2 ... zrychleníčko, zrychlení**

3 body

Na obrázku vidíte náčrt elipsy s ohnisky  $F_1$  a  $F_2$  a několika vyznačenými body na ní. Uvažujte, že elipsa znázorňuje trajektorii nějakého hmotného bodu. Znázorněte do obrázku zrychlení, která působí na hmotný bod v jednotlivých vyznačených bodech dráhy pro dvě situace (jde o směry a vzájemné poměry zrychlení (které je větší/menší) v různých bodech v rámci jednoho náčrtu).



- V ohnisku  $F_1$  je umístěno hmotné těleso, kolem kterého hmotný bod obíhá. Uvažujeme, že platí 2. Keplerův zákon.
- Těleso má konstantní velikost rychlosti, pouze se pohybuje po elipse.

(řešení str. 48)**Úloha III.3 ... IDKFA**

6 bodů

Vypálili jste na impa z plazmové pušky, která střílí stabilní shluk částic s rovnoměrným rozdělením podélné rychlosti v intervalu  $\langle v_0, v_0 + \delta v \rangle$  (příčná rychlost je

nulová) a s celkovou energií  $E_0$ . Hlaveň pušky má průřez  $S$  a pulz trvá nekonečně krátký čas. Jak daleko musí imp stát, aby se mu nic nestalo? Předpokládejte, že jeho kůže bez problémů uchládí na malém prostoru tepelný tok  $q$ .

(řešení str. 51)

### Úloha III.4 ... upuštěná propiska

7 bodů

Propisku (tuhou tyč) upustíme na stůl tak, že během svého letu svírá úhel  $\alpha$  s vodorovnou rovinou. Jakou rychlostí dopadne její druhý konec (ten, co se stolu dotkne jako druhý), jestliže jsme těžiště upustili z výšky  $h$ ? Všechny srážky jsou nepružné a tření mezi stolem a koncem propisky dostatečně velké.

*Bonus* Spočítejte, jaký musíme zvolit úhel  $\alpha$ , aby druhý konec dopadl s co nejvyšší rychlostí. Pro jaké výšky se vyplatí propisku naklonit?

(řešení str. 52)

### Úloha III.5 ... rozpad sem, rozpad tam

8 bodů

Máme  $A_0$  částic typu  $A$ , které se s rozpadovou konstantou  $\lambda_A$  rozpadají na částice typu  $B$ . Ty se zase s rozpadovou konstantou  $\lambda_B$  rozpadají na částice typu  $A$  a na začátku jich je  $B_0$ . Najděte funkci udávající poměr počtů částic typů  $A$  a  $B$  v čase.

(řešení str. 56)

### Úloha III.P ... složený papír

8 bodů

Každý to jistě někdy slyšel a určitě i zkusil: „List papíru nelze na půlku přeložit více než sedmkrát.“ Je to ale skutečně pravda? Najděte hraniční podmínky.

(řešení str. 59)

### Úloha IV.1 ... zmrzlina

3 body

Odhadněte, kolik gramů zmrzliny dokážeme vyrobit, pokud máme k dispozici 5 l kapalného dusíku o teplotě  $-196^\circ\text{C}$  a neomezené množství mléka a smetany o pokojové teplotě  $22^\circ\text{C}$ ? Předpokládejme, že požadovaná zmrzlina se skládá jen z mléka a smetany (hmotnostně půl na půl) a měla by mít teplotu  $-5^\circ\text{C}$ . Protože se tepelné kapacity mléka a smetany v tomto intervalu teplot značně mění, počítejte s jejich průměrnými hodnotami  $c_m = 3,45 \text{ kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$  pro mléko a  $c_s = 4,45 \text{ kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$  pro smetanu. Zbylé potřebné údaje si dohledejte na internetu.

(řešení str. 63)

### Úloha IV.2 ... autisti

3 body

Kolik nejméně dětí by muselo roztočit svůj fidget spinner, aby se tak den na Zemi prodloužil o 1 ms? Všechny neznámé veličiny odhadněte.

(řešení str. 64)

### Úloha IV.3 ... divně tvarovaná nádobka

6 bodů

Máme válcovou skleničku, která má zboku u dna malou díru o ploše  $S$ . Tato nádoba je naplněná vodou, která samovolně přetéká do druhé nádoby, která je tentokrát

již bez díry. Jaký tvar by musela mít druhá nádoba, aby v ní hladina rostla rovnoměrně? Předpokládejte, že má být válcově symetrická.

*Bonus* Dna obou nádob jsou ve stejné výšce a nádoby jsou dírou spojené.

(řešení str. 66)

#### Úloha IV.4 ... vymyslete si sami

7 bodů

Máme černou skříňku se třemi výstupy (A, B a C). Víme, že obsahuje  $n$  rezistorů se stejným odporem, ale nevíme jak jsou zapojeny. Změříme tedy odpory mezi dvojicemi bodů AB, BC a CA a zjistíme, že  $R_{AB} = 3\ \Omega$ ,  $R_{BC} = 5\ \Omega$  a  $R_{CA} = 6\ \Omega$ . Zjistěte, kolik nejméně rezistorů může skříňka obsahovat a určete příslušný odpor jednoho rezistoru.

(řešení str. 69)

#### Úloha IV.5 ... nemožnost nakažení

7 bodů

Představme si, že roztlačíme nějakou bakterii obvyklé velikosti na rychlost  $v = 50\ \text{km}\cdot\text{h}^{-1}$  ve vodorovném směru a necháme ji volně letět ve vzduchu. Jakou vzdálenost zhruba urazí, než se zastaví?

Výsledek vás možná překvapí. Jak je tedy možné se infikovat tímto způsobem bakteriální infekcí? Diskutujte, proč je to možné i přes takový výsledek.

(řešení str. 73)

#### Úloha IV.P ... Voyager II a Voyager I žijí!

9 bodů

Máme nějaký satelit, který chceme vypustit ven ze Sluneční soustavy. Vypouštíme ho z oběžné dráhy Země tak, že po nějakých korekcích dráhy získá rychlost, která je vyšší než úniková rychlost ze Sluneční soustavy. Jaká je pravděpodobnost, že dojde ke kolizi sondy s nějakým kosmickým materiálem s průměrem větším než  $d = 1\ \text{m}$  před opuštěním Sluneční soustavy?

(řešení str. 76)

#### Úloha V.1 ... schodiště na Měsíci

3 body

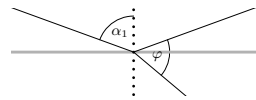
Pokud bychom jednou kolonizovali Měsíc, bylo by vhodné na něm používat schody? Představte si na Měsíci klesající schodiště s výškou schodu  $h = 15\ \text{cm}$  a délkou  $d = 25\ \text{cm}$ . Odhadněte počet schodů  $N$ , které by přeletěl člověk, jestliže by před vstupem na schody šel rychlostí  $v = 5,4\ \text{km}\cdot\text{h}^{-1} = 1,5\ \text{m}\cdot\text{s}^{-1}$ . Tíhové zrychlení na povrchu Měsíce je šestkrát slabší než na povrchu Země.

(řešení str. 81)

#### Úloha V.2 ... paprsky smrti na skle

3 body

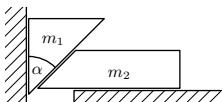
Na skleněnou desku s absolutním indexem lomu  $n = 1,5$  dopadá světelný paprsek. Stanovte jeho úhel dopadu  $\alpha_1$ , jestliže paprsek odražený od rozhraní svírá s lomeným paprskem úhel  $60^\circ$ . Deska je uložena ve vzduchu.



(řešení str. 82)

### Úloha V.3 ... klín

5 bodů



Máme dva klíny o hmotnostech  $m_1$ ,  $m_2$  a úhel  $\alpha$  (viz obrázek). Vypočítejte zrychlení levého klínu. Předpokládejte, že nikde nedochází ke tření.

*Bonus* Uvažujte tření s koeficientem  $f$ .

(řešení str. 83)

### Úloha V.4 ... tepelné ztráty

7 bodů

Na jaké teplotě se ustálí vnitřní prostředí bytu v panelovém domě? Uvažujte, že náš byt sousedí delšími stěnami, stropem a podlahou s dalšími byty, ve kterých je udržována teplota  $22^\circ\text{C}$ . Kratšími stěnami sousedí s okolím, kde je teplota  $-5^\circ\text{C}$ . Vnitřní rozměry bytu jsou: výška  $h = 2,5\text{ m}$ , šířka  $a = 6\text{ m}$  a délka  $b = 10\text{ m}$ . Součinitel měrné teplotní vodivosti stěn je  $\lambda = 0,75\text{ W}\cdot\text{K}^{-1}\cdot\text{m}^{-1}$ . Vnější stěny a stropy jsou tlusté  $D_{\text{out}} = 20\text{ cm}$  a vnitřní  $D_{\text{in}} = 10\text{ cm}$ .

Jak se změní výsledek, pokud budovu zvenku zateplíme polystyrenem o tloušťce  $d = 5\text{ cm}$  s měrnou tepelnou vodivostí  $\lambda' = 0,04\text{ W}\cdot\text{K}^{-1}\cdot\text{m}^{-1}$ ? (řešení str. 85)

### Úloha V.5 ... záludná kapka

8 bodů

Mějme kulatou kapku o poloměru  $r_0$  tvořenou vodou o hustotě  $\rho_v$ , která shodou okolností padá v mlze v homogenním tíhovém poli  $g$ . Uvažujme vhodnou mlhu se speciálními předpoklady. Tvoří ji vzduch o hustotě  $\rho_{\text{vzd}}$  a vodní kapičky s průměrnou hustotou  $\rho_r$ , když uvážíme, že se rozptýlí zcela rovnoměrně. Jestliže kapka propadne nějakým objemem takové mlhy, vysbírá všechnu vodu, která se v tomto objemu nachází. Na místě zůstane pouze vzduch. Jaká je závislost hmotnosti kapky na vzdálenosti uražené v takovéto mlze?

*Bonus* Řešte pohybové rovnice.

(řešení str. 86)

### Úloha V.P ... plovoucí rtuť

9 bodů

Vymyslete co nejvíce fyzikálních „figlů“, díky kterým by rtuť, alespoň po omezenou dobu, plavala na kapalné vodě. Čím trvalejší řešení naleznete, tím lépe.

(řešení str. 88)

### Úloha VI.1 ... asi se urazila

3 body

Máme dva hmotné body o stejných hmotnostech  $m$  ve vzdálenosti  $d$  od sebe volně v prostoru bez žádných vnějších gravitačních sil. Jakou minimální rychlost ve směru spojnice obou bodů musíme udělit jednomu hmotnému bodu, aby se od sebe stále vzdalovaly?

(řešení str. 90)

### Úloha VI.2 ... horký drát

3 body

Vypočítejte proud, který by měl procházet kovovým vláknem s průměrem  $d = 0,10\text{ mm}$  nacházejícím se ve vakuové baňce, aby teplota vlákna měla stálou

hodnotu  $T = 2600 \text{ K}$ . Předpokládejte, že povrch vlákna září jako ideální černé těleso. Zanedbejte ztráty tepla způsobené vedením tepla. Rezistivita materiálu vlákna při dané teplotě je  $\rho = 2,5 \cdot 10^{-4} \Omega \cdot \text{cm}$ .

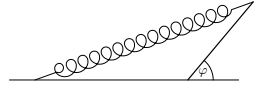
*Nápověda* Použijte Stefanův-Boltzmannův zákon.

(řešení str. 91)

### Úloha VI.3 ... neanalytická pružinka

6 bodů

Představme si tyč s délkou  $b = 5 \text{ cm}$  a hmotností  $m = 1 \text{ kg}$  a pružinku s klidovou délkou  $c = 10 \text{ cm}$ , s tuhostí  $k = 200 \text{ N} \cdot \text{m}^{-1}$  a se zanedbatelnou hmotností, které jsou na koncích spojeny. Druhé konce tyče a pružinky jsou



upevněny ve stejné výšce ve vzdálenosti  $a = 10 \text{ cm}$  od sebe. Kolem obou upevnění i kolem spoje lze libovolně rotovat. Označme  $\varphi$  sklon tyče od horizontálního směru. Najděte všechny hodnoty  $\varphi$ , pro které je soustava v rovnovážné poloze. Které z těchto poloh jsou stabilní a které labilní?

(řešení str. 93)

### Úloha VI.4 ... rozměrová analýza

7 bodů

Matěj si doma vyrobil střelnou zbraň a chce změřit, jakou rychlostí vystřeluje náboje. Bohužel nemá k dispozici jiný měřicí přístroj než pravítko. Našel ale kostku, jež je tvořena z poloviny ocelí a poloviny dřevem. Položí ji na kraj stolu (jehož výška je  $100 \text{ cm}$  a délka je  $200 \text{ cm}$ ) a horizontálně do ní vystřelí. Kulka se od ocelové strany dokonale pružně odrazí přesně opačným směrem a dopadne do vzdálenosti  $50 \text{ cm}$  od stolu. Kostka se na stole posune o  $5 \text{ cm}$ . Potom Matěj kostku otočí a střelí do její dřevěné strany, v níž se kulka zaryje. Nyní naměřil posunutí jen  $4 \text{ cm}$ . Pomozte mu s výpočtem rychlosti výstřelu. Možná se vám bude hodit, že zjistil, že pohyb rozjeté kostky po stole se nezastaví, pokud jednu stranu stolu zvedne do výšky alespoň  $20 \text{ cm}$ .

(řešení str. 95)

### Úloha VI.5 ... skok z letadla

8 bodů

Filip o hmotnosti  $80 \text{ kg}$  vyskočil z letadla, které je ve výšce  $h_1 = 500 \text{ m}$  nad zemí. Ve stejném okamžiku z druhého letadla skočila Danka o hmotnosti  $50 \text{ kg}$ , ale z výšky  $h_2 = 569 \text{ m}$  nad zemí. Předpokládejme, že oba mají stejný odporový koeficient  $C = 1,2$ , Filipova plocha příčného průřezu je  $S_F = 2,2 \text{ m}^2$  a Dančina je  $S_D = 1,5 \text{ m}^2$ . Hustota vzduchu  $\rho = 1,205 \text{ kg} \cdot \text{m}^{-3}$  se nemění s výškou. Za jakou dobu od výskoku bude Danka ve stejné výšce nad zemí jako Filip?

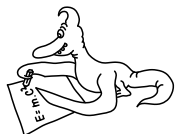
(řešení str. 97)

### Úloha VI.P ... kompenzace vesmírné expanze

9 bodů

Podle současných pozorování a vesmírných modelů se zdá, že vesmír se rozpíná a rychlost rozpínání se zvyšuje. Co kdyby to tak nebylo? Co kdyby byl vesmír stále stejně velký, ale měnily by se fyzikální zákony/konstanty? Jak by se musely konstanty měnit, aby se nám zdálo, že se vesmír rozpíná, jak ukazují pozorování? Popište co nejvíce zákonů, které by se musely měnit.

(řešení str. 100)



## Řešení teoretických úloh

### Úloha I.1 ... kávu si omléčnám

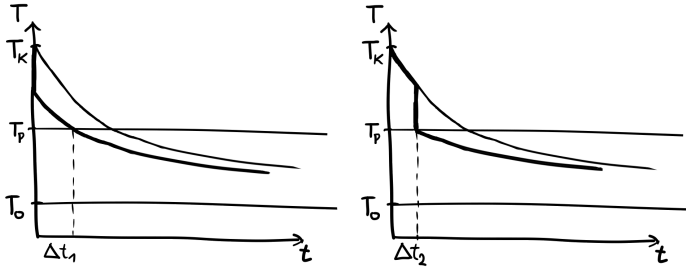
*Kdy je nejvhodnější nalít do horké kávy chladné mléko, abychom ji mohli pít co nejdříve? Nepožadujeme přesný výpočet, ale podrobný slovní popis toho, jak káva chladne a jak byste postupovali.*

Naším cílem při řešení této úlohy bude najít praktickou radu, podle které se můžeme opravdu při pití kávy řídit. Zcela obecný rozbor by byl velmi náročný i pro jednoduché modely chladnutí, neboť závisí na alespoň pěti parametrech: počáteční teplota kávy  $T_k$ , teplota okolí  $T_o$ , teplota vhodná k pití  $T_p$ , teplota mléka  $T_m$  a poměr objemu mléka k objemu kávy  $\varepsilon$ . Abychom získali jedno konkrétní řešení, je potřeba rozumně odhadnout hodnoty těchto veličin. My budeme volit hodnoty blížící se reálné situaci  $T_k = 100\text{ }^\circ\text{C}$ ,  $T_o = 20\text{ }^\circ\text{C}$ ,  $T_p = 60\text{ }^\circ\text{C}$ ,  $T_m = 10\text{ }^\circ\text{C}$  a  $\varepsilon = 0,5$ . Kdybychom volili kávu s velkým obsahem mléka, jako například cappuccino nebo latte, úloha by postrádala smysl – u těchto typů kávy musíme totiž mléko naopak přehřát, aby nebyla výsledná směs moc chladná, a může se do hrnku přidat okamžitě. K našim předpokladům ještě přidáme zjednodušení, že výměna tepla mezi kávou a mlékem proběhne okamžitě.

Mezi vzduchem a kapalinou v hrnku probíhá tepelná výměna. Zásadní charakteristikou této výměny je, že tepelný tok (teplo vyměněné za čas) s časem klesá. Teplota kávy se tedy postupně blíží teplotě okolí, ale čím dál tím pomaleji. Jinými slovy, graf závislosti teploty na čase je představován konvexní funkcí. Podle našich odhadů je teplota mléka  $T_m$  výrazně nižší než teplota kávy  $T_k$  a minimální teplota kávy, při které ji přidání mléka nezchladí pod  $T_p$ , je o dost blíž  $T_k$  než  $T_p$ . Proto pokud přidáme mléko později, bude dosažený pokles teploty jen o zanedbatelně málo nižší, než když mléko přidáme dříve.

Nyní uvažujme případ A, tj. že mléko přidáme do kávy ihned. Potom jsme z teploty na  $T_k$  skočili na jistou teplotu  $T_k - \Delta T > T_p$  a pak bychom čekali, až káva ochladne na  $T_p$ . Přeskočili bychom tedy oblast rychlého chladnutí (horká káva chladne rychleji) a ponechali bychom oblast pomalého chladnutí (studená káva chladne pomaleji). Pokud však chvíli počkáme – případ B – a ochladíme kávu rychle na teplotu zhruba  $T_p + \Delta T$ , potom můžeme přidat mléko, srazit tak teplotu na  $T_p$  a oblast pomalého chlazení přeskočit. Káva musí stejně vychladnout o  $T_k - T_p - \Delta T$ , v případě B bude toto chladnutí ale probíhat při vyšší teplotě než v případě A. Na základě této úvahy je lepší přidat mléko později, viz grafické znázornění na obrázku 1.

Znovu však zdůrazněme, že v závislosti na zvoleném modelu a počátečních parametrech mohl každý dospět k jinému výsledku. Pro ilustraci, pokud bychom předpokládali, že jsme ochotni pít kávu poměrně horkou, přidáváme pouze malé množství mléka. Navíc můžeme tvrdit, že chladnutí je zpočátku přibližně lineární.



Obr. 1: Náčrt křivek chladnutí pro případ, kdy nalijeme mléko do kávy dříve (vlevo) a později (vpravo). Jak je naznačeno, v druhém případě je čas chladnutí  $\Delta t_2$  kratší než čas chladnutí  $\Delta t_1$  z prvního případu.

V tomto případě bychom získali opačný výsledek. Mléko by bylo kvůli lineárnímu chladnutí vhodné nalít do kávy co nejdříve, protože výsledná teplota je váženým průměrem teploty kávy a mléka, a proto „potenciál zchladit“, který mléko má, klesá s teplotou (a tedy i s časem). Jinými slovy,  $\Delta T$  zmíněné v případě A, by bylo nezanedbatelně větší než  $\Delta T$  v případě B. Tento rozdíl by přebil vliv konvexity křivky chladnutí.

Jako bonus si můžete zkusit najít model popisující čas chladnutí na danou teplotu a najít optimální čas přilít s rozбором všech případů v závislosti na počátečních parametrech.

Na závěr ještě poznamenejme, že pokud pijeme kávu s malým množstvím mléka, tak se pravděpodobně jedná o celkově malý objem nápoje, který sám o sobě bude chladnout rychle. Proto rozdíl způsobený pozdním či brzkým přilítím mléka pravděpodobně nepoznáte.

## Úloha I.2 ... zálohovací NAS(A)

Uvažujte optický switch (propustnost  $10 \text{ Gb}\cdot\text{s}^{-1}$ ), jehož výstup (po patřičném zesílení) použijete k ozáření Měsíce. Díky zrcátkům zanechaným na jeho povrchu z dob projektu Apollo se signál vrátí zpět a přivedete jej (po patřičném zesílení) na vstup switche. Pokud zajistíme spolehlivé fungování switche, budou jednou vyslaná data v systému „obíhat“ trvale, takže jsme získali paměť. Jaká je její maximální kapacita? Dobu zpracování ve switchi a velikost datových hlaviček zanedbejte.

Dostane-li naše úložiště data k uložení, vyšle je skrze optický switch (přepínač) k Měsíci a dokud se nevrátí zpět, nemusí se o ně starat. Po návratu je musí přeposlat a tedy nemůže přijímat nová data. Toto se stane za čas  $2l/c$ , kde  $l$  je vzdálenost mezi switchem a zrcátko,  $c$  je rychlost světla ve vakuu (průchod atmosférou zanedbáváme). Chceme-li využít systém naplno, stihneme za tuto dobu odeslat  $2lB/c$  dat, kde  $B$  je propustnost switche.

Použijme konkrétní hodnoty  $B = 10 \text{ Gb}\cdot\text{s}^{-1}$ ,  $l = 3,8 \cdot 10^8 \text{ m}$ ,  $c = 3 \cdot 10^8 \text{ m}\cdot\text{s}^{-1}$  a dostáváme<sup>1</sup>

$$s = \frac{2Bl}{c} \approx 25 \text{ Gb} \approx 3,2 \text{ GB} \approx 2,9 \text{ GiB}.$$

Vidíme, že se nejedná o příliš efektivní způsob ukládání dat, kapacitu má menší než běžně dostupná flashka a cena mise Apollo nebyla malá.

### Úloha 1.3 ... oběšený úhelník

Máme homogenní úhelník ve tvaru  $L$  o stranách délek  $b, c$ . Je volně zavěšen v železničním vagóně za konec jedné strany tak, že jeho vrchol míří ve směru jízdy vagonu. S jakým zrychlením  $a$  se musí vagon pohybovat, aby spodní strana úhelníku byla rovnoběžná se směrem jízdy? Relativistické jevy neuvažujte.

Bonus: Relativistické jevy uvažujte.

Nejprve musíme najít těžiště. Zvolíme soustavu souřadnou tak, aby měla počátek v bodě zavěšení a osa  $x$  mířila podél ramena úhelníku délky  $b$ , za které je zavěšený. Zavedeme délkovou hustotu  $\varrho$ , kterou sice neznáme, ale zpřehlední nám výpočty. Potom je polohový vektor těžiště dán vztahem

$$\mathbf{r}_t = \frac{\sum m_i \mathbf{r}_i}{\sum m_i},$$

kde  $m_i$  je hmotnost  $i$ tého hmotného bodu (nebo části tělesa) a  $\mathbf{r}_i$  je jeho polohový vektor (u větších celků těžiště celku – tady strany úhelníku). Dosadíme hodnoty

$$\mathbf{r}_t = \frac{\mathbf{r}_b \varrho b + \mathbf{r}_c \varrho c}{(b+c)\varrho},$$

kde  $\mathbf{r}_b$  a  $\mathbf{r}_c$  jsou polohové vektory těžišť jednotlivých stran, ležící v jejich středech. Můžeme dosadit polohové vektory  $\mathbf{r}_b = [b/2; 0]$ ,  $\mathbf{r}_c = [b; c/2]$  a pokrátit  $\varrho$ , čímž získáme

$$\mathbf{r} = \frac{[b/2; 0] \cdot b + [b; c/2] \cdot c}{b+c} = \left[ \frac{b^2/2 + bc}{b+c}; \frac{c^2/2}{b+c} \right].$$

Úhel, který v klidu svírá svislá strana úhelníku s kolmicí, pak bude

$$\arctg \frac{r_y}{r_x} = \arctg \frac{c^2/2}{b^2/2 + bc}.$$

To ovšem musí být i úhel, který musí svírat výslednice gravitační a setrvačné síly s kolmicí, aby byl moment síly působící vůči bodu zavěšení nulový. Proto z podobnosti trojúhelníků máme

$$\frac{ma}{mg} = \frac{c^2/2}{b^2/2 + bc},$$

$$a = g \frac{c^2/2}{b^2/2 + bc}.$$

---

<sup>1</sup>Gb je gigabit, GB je gigabyte a GiB je gibibyte, tedy  $1024^3$  bytů



### Relativistické jevy

Pokud chceme uvažovat relativistické efekty, musíme si začít dávat pozor, v jaké soustavě se nacházíme a v jaké soustavě nás zajímá zrychlení. Naskýtají se dvě význačné možnosti: V soustavě spojené s vlakem (inerciální soustavě pohybující se okamžitou rychlostí vlaku) nebo soustavě spojené se zemí. Není těžké si rozmyslet, že zrychlení v první soustavě (tedy zrychlení, kterým vlak sám sebe pohání) je stejné, jako tomu bylo v nerelativistickém případě, protože vlak se vůči této soustavě pohybuje pomalu.

Pro inerciální soustavu spojenou se zemí (nádražím) zkusme uvažovat, co by se asi dělo. Z počátku, kdy bude rychlost malá ve srovnání s rychlostí světla, nesmí zrychlení být výrazně jiné než to, které jsme vypočítali výše. Jak bude rychlost růst, začnou se projevovat relativistické efekty. Pro nás je důležité především zkracování spodní strany úhelníku.<sup>2</sup> V limitě dosažení rychlosti světla bude spodní strana úhelníku mít nulovou délku, a proto bude síla potřebná k udržení úhelníku v kolmé poloze nulová.<sup>3</sup>

Zkusme přece jen spočítat, jak se systém bude chovat z pohledu nádraží. Se zkracováním spodní strany se bude měnit setrvačná síla potřebná k udržení úhelníku ve správné poloze. Velikost síly dokážeme získat dosazením závislosti délky spodní strany na rychlosti  $c(v) = \frac{c_0}{\gamma(v)}$  do vzorečku, který už známe, kde  $\gamma$  je Lorentzův faktor daný vztahem

$$\gamma(v) = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}},$$

kde  $c_0$  je klidová délka. Rychlost světla budeme značit  $C$ , abychom se vyhnuli kolizi ve značení. Zároveň musíme použít vztah pro relativistickou změnu hmotnosti  $m = m_0\gamma$ . Pro polohu těžiště pak platí

$$\mathbf{r} = \frac{[b/2; 0] \cdot m_b + [b; \frac{c_0}{2\gamma}] \cdot m_c}{m_b + m_c} = \left[ \frac{bm_b/2 + bm_c}{m_b + m_c}; \frac{c_0 m_c}{2\gamma(m_b + m_c)} \right].$$

Dosazením za hmotnosti jednotlivých stran<sup>4</sup>  $m_b = \varrho\gamma b, m_c = \varrho\gamma c_0$  pak získáme vztah pro setrvačnou sílu  $F$  ( $m_0$  je klidová hmotnost úhelníku)

$$m_0 g c_0^2 = F (b^2 + 2bc_0).$$

Bohužel při relativistických rychlostech neplatí vztah  $F = ma$ , proto je třeba užít obecnější vztah  $F = \frac{dp}{dt}$ , který v relativitě přejde na

$$F = \frac{dp}{dt} = \frac{dm}{dt}v + m \frac{dv}{dt} = m_0 \frac{d\gamma}{dt}v + m_0\gamma \frac{dv}{dt} = m_0\gamma^3 \frac{v^2}{C^2} \frac{dv}{dt} + m_0\gamma \frac{dv}{dt} = m_0\gamma^3 \frac{dv}{dt}.$$

<sup>2</sup>Druhá strana se nezkracuje, protože uvažujeme, že síly jsou v rovnováze po celou dobu pohybu. Proto je tato strana po celou dobu pohybu orientována kolmo ke směru pohybu

<sup>3</sup>Což i ve speciální relativitě pro náš systém odpovídá nulovému zrychlení, což je správně (fyzikální zákony nás „nenutí“ přesáhnout rychlost světla)

<sup>4</sup>Používáme klidové délky  $b$  a  $c_0$ , protože škálujeme klidovou hmotnost Lorentzovým faktorem. Započítání relativistické kontrakce by znamenalo pouze přeškálování hustoty, aby se celkový vztah nezměnil.

Získáme tak diferenciální rovnici

$$\gamma^3 \frac{dv}{dt} = \frac{gc_0^2}{b^2 + 2bc_0} = a_0,$$

kde  $a_0$  je nerelativistické („klidové“) zrychlení. Tu umíme vyintegrovat na vztah ( $t_0$  je čas začátku jízdy vlaku)

$$\frac{Cv}{\sqrt{C^2 - v^2}} = a_0(t - t_0),$$

což umíme invertovat na rovnici

$$v = \frac{a_0(t - t_0)}{\sqrt{1 + \left(\frac{a_0(t - t_0)}{C}\right)^2}}.$$

Zrychlení pak získáme derivací podle času,

$$a = \frac{dv}{dt} = \frac{a_0}{\left(1 + \left(\frac{a_0(t - t_0)}{C}\right)^2\right)^{3/2}}.$$

Zkusme se podívat na limitní chování obou vzorců. Pro malé časy, kdy se dá odmocnina ve jmenovateli aproximovat jedničkou, vychází zrychlení konstantní,<sup>5</sup> zatímco rychlost je lineární funkcí času odpovídající danému zrychlení. Pro velké časy naopak můžeme v odmocnině ve jmenovateli zanedbat jedničku, a odmocnina se pak bude chovat jako lineární funkce času. To naopak dává konstantní rychlost (překvapivě rovnou  $C$ ) a nulové zrychlení, jak jsme předpovídali dříve.

Na závěr uveďme, že pokud dosadíme do vzorce pro relativistické zrychlení hodnotu  $a_0 = g$ ,<sup>6</sup> pak potřebné zrychlení klesne o 1% cca po měsíci zrychlování. Poloviční bude po přibližně devíti měsících.

## Úloha I.4 ... praská mi v láhvi

Co když si skoro prázdnou 1,5 litrovou PET láhev uzavřeme v dobře vytápěné kanceláři, dejme tomu na  $t_k = 26^\circ\text{C}$ , a pak vyjdeme vstříc novým zážitkům dolů ze schodů? Láhev začne praskat. Co má větší vliv? To, že se mění atmosférický tlak, jak scházíme 10 pater v budově, nebo to, že je na schodech, dejme tomu,  $t_s = 15^\circ\text{C}$ ?

Vyřešme problém nejdříve pro případ prázdné láhve (bez vody) s objemem  $V_0$ , kdy sejdem všechna poschodí a tlak i teplota v láhvi se vyrovnají s tlakem a teplotou okolí  $t_s$ , resp. odpovídající termodynamickou teplotou  $T_s$  (z původní teploty  $t_k$ , resp.  $T_k$ ). Sejdem-li deset poschodí, změní se atmosférický tlak z hodnoty  $p_0 \doteq 101\text{ kPa}$  na hodnotu  $p_{10}$ . Rozdíl tlaků je dán tíhou odpovídajícího sloupce

<sup>5</sup>A je stejně veliké jako nerelativistické, které jsme spočítali dříve.

<sup>6</sup>Což je spíš nadsazené, neboť to odpovídá spodní straně téměř dvaapůlkrát delší než  $t_a$ , za kterou úhelník visí. Navíc cestující by takové zrychlení příliš neoceníli, zvláště ti sedící proti směru jízdy.

vzduchu, tj.  $p_{10} - p_0 = h\rho g$ , kde  $h = 10 \cdot 3 \text{ m} = 30 \text{ m}$  je výškový rozdíl (počítáme se standardní výškou podlaží 3 m),  $\rho \doteq 1,3 \text{ kg}\cdot\text{m}^{-3}$  je hustota vzduchu a  $g$  je tíhové zrychlení. Výpočtem zjistíme, že změna tlaku je číselně rovna  $p_0 - p_{10} \doteq 380 \text{ Pa}$ . Objem láhve se zmenší na hodnotu  $V_s$ . Pro počáteční a koncové podmínky platí stavová rovnice

$$\frac{p_0 V_0}{T_k} = \frac{p_{10} V_s}{T_s}.$$

Odtud získáme výsledný objem

$$V_s = V_0 \frac{p_0 T_s}{p_{10} T_k}.$$

Číselně je  $V_s \doteq 1,4391$ . Objem láhve se tedy zmenšil asi o  $V_0 - V_s \doteq 61 \text{ ml}$ , tj. asi o 4%.

### Hrubá láhev – změna tlaku

Uvažujme hrubou láhev, která má větší schopnost odolávat změnám vnějšího tlaku. To znamená, že tlak v láhvi se nezmění okamžitě na hodnotu okolního atmosférického tlaku, ale až po překročení určité hranice prasknutí. Prasknutím láhve myslíme změnu objemu láhve za krátký okamžik, která je doprovázena charakteristickým zvukem.

Rozeberme případ prasknutí láhve pod vlivem změny okolního tlaku. Provedme výpočet např. pro případ, kdy nám láhev praskne, až když sejdeme deset poschodí. Předpokládejme, že jsme schody seběhli dostatečně rychle, a proto se nestihl vzduch v lahvi zchladit. Prasknutí láhve pak můžeme považovat za přibližně adiabatický děj, protože probíhá velmi rychle a za tento krátký okamžik nedojde k tepelné výměně mezi vzduchem a PET láhví. Pro adiabatický děj platí Poissonův zákon

$$p_0 V_0^\kappa = p_{10} V_1^\kappa,$$

kde  $V_0$  je objem vzduchu na začátku,  $V_1$  je objem vzduchu po adiabatické kompresi a  $\kappa = 7/5$  je Poissonova konstanta pro plyn skládající se z dvouatomových molekul. Dále zde platí stavová rovnice

$$\frac{p_0 V_0}{T_k} = \frac{p_{10} V_1}{T_1},$$

kde  $T_1$  je výsledná teplota po adiabatické kompresi, která se změnila z původní hodnoty  $T_k$ . Použitím dvou výše uvedených rovnic vypočteme výsledný objem  $V_1$  a teplotu  $T_1$ .

$$V_1 = V_0 \left( \frac{p_0}{p_{10}} \right)^{\frac{1}{\kappa}} \doteq 1,4961.$$

Objem se nám změní asi o 4 ml, tedy o 0,27%. Výsledná teplota po adiabatické kompresi je

$$T_1 = T_k \left( \frac{p_{10}}{p_0} \right)^{\frac{\kappa-1}{\kappa}} \doteq 299,5 \text{ K}.$$

Tomu odpovídá teplota  $t_1 = 26,3 \text{ }^\circ\text{C}$ . Teplota se tedy zvýšila o  $0,3 \text{ }^\circ\text{C}$ , což je v porovnání se změnou o  $11 \text{ }^\circ\text{C}$  v druhém případě velmi málo.

### Hrubá láhev – změna teploty

Nyní vyšetřeme druhý případ, kdy teplota vzduchu v láhvi klesne z hodnoty  $t_k$  na  $t_s$ , resp. z hodnoty  $T_k$  na  $T_s$ . Opět předpokládejme, že dojde k jednomu prasknutí, a to až po vyrovnání teplot. Po dobu ochlazování dochází k izochorickému ději – objem láhve se nemění. Tlak vzduchu se změní z hodnoty  $p_0$  na hodnotu  $p_1$ . Pro stavové veličiny platí Charlesův zákon

$$\frac{p_0}{T_k} = \frac{p_1}{T_s}.$$

Odtud zjistíme tlak v láhvi před prasknutím z rovnice

$$p_1 = p_0 \frac{T_s}{T_k}. \quad (1)$$

Číselně je  $p_1 \doteq 97\,290$  Pa a jeho rozdíl s okolním tlakem je  $p_0 - p_1 \doteq 3,71$  kPa. Poté dojde podle předpokladu k prasknutí. Jedná se o adiabatickou kompresi, kdy se teplota zvýší z hodnoty  $t_s$  na  $t_2$ , resp. z hodnoty  $T_s$  na  $T_2$ , tlak se zvýší z hodnoty  $p_1$  na  $p_0$  a objem se zmenší z hodnoty  $V_0$  na  $V_2$ . Stavová rovnice a Poissonův zákon budou v tomto případě vypadat následovně

$$\begin{aligned} \frac{p_1 V_0}{T_s} &= \frac{p_0 V_2}{T_2}, \\ p_1 V_0^\kappa &= p_0 V_2^\kappa. \end{aligned}$$

Odtud získáme výsledný objem

$$V_2 = V_0 \left( \frac{p_1}{p_0} \right)^{\frac{1}{\kappa}}$$

a výslednou teplotu

$$T_2 = T_s \left( \frac{p_0}{p_1} \right)^{\frac{\kappa-1}{\kappa}}.$$

Ještě zde potřebujeme dosadit vztah pro tlak  $p_1$  z rovnice (1). Po úpravě dostaneme

$$V_2 = V_0 \left( \frac{T_s}{T_k} \right)^{\frac{1}{\kappa}},$$

$$T_2 = T_s \left( \frac{T_k}{T_s} \right)^{\frac{\kappa-1}{\kappa}}.$$

Číselně je  $V_2 \doteq 1,4601$ . Objem vzduchu v láhvi se tedy změnil o  $V_2 - V_0 \doteq 40$  ml, což je 2,7%. Teplota je číselně  $T_2 \doteq 291$  K, resp.  $t_2 \doteq 18^\circ\text{C}$ , liší se tedy asi o  $3^\circ\text{C}$  vzhledem k teplotě okolního vzduchu. Opět může dojít k jeho ochlazení a k následnému prasknutí.

**Shrnutí pro hrubou láhev**

Shrnmě, co jsme vyšetřením vlivu okolních podmínek na láhev zjistili. Míru daného vlivu na PET láhev můžeme porovnávat jak rozdílem odpovídajících tlaků, tak rozdílem odpovídajících teplot. Sejdeme-li deset poschodí, změní se okolní tlak působící na láhev asi o 380 Pa. Naopak necháme-li vzduch v lahvi jen postupně chladnout, změní se jeho tlak až o 3 710 Pa. Dále sejdeme-li deset pater, změní se teplota vzduchu během adiabatického prasknutí jen asi o  $0,3^\circ\text{C}$ , zatímco během izochorického chladnutí v druhém případě se teplota vzduchu změní asi o  $t_2 - t_s = 8^\circ\text{C}$ . Dále můžeme porovnat změnu objemu vzduchu po prasknutí. Výškový rozdíl 10 pater způsobil změnu asi 4 ml, vlivem teploty se objem zmenšil až o 40 ml.

Z našich výsledků vidíme, že v případě vlivu daném změnou okolního tlaku a teploty má zásadní vliv na prasknutí láhve právě změna teploty. Pro kompletnost řešení je třeba dodat, že láhev může na změnu okolního tlaku reagovat prasknutím prakticky okamžitě. Naproti tomu k tepelné výměně mezi okolním vzduchem a PET láhví a mezi samotnou PET láhví a vzduchem v ní uzavřeném dochází déle. Je to způsobeno rychlostí vedení tepla a tepelnou kapacitou láhve se vzduchem.

Tady se dostáváme k problému, kdy v láhvi bude trocha vody. Tento problém jenom prodiskutujeme slovně bez výpočtu. Kromě toho, že více vody způsobí méně vzduchu v láhvi, také zapříčiní poměrně velké zvýšení tepelné kapacity soustavy PET láhve s tekutinami vody a vzduchu. To znamená, že na danou změnu teploty soustavy je zapotřebí dodat více tepla – stejnou změnu teploty docílíme za jinak stejných podmínek až za delší čas. V našem případě se to může projevit tak, že sejdeme-li schody rychle, změní se odpovídajícím způsobem okolní tlak, ale teplota soustavy se při dostatečném množství vody změní jen velmi málo.

Lze též uvažovat relativně tenkostěnnou láhev, která přímo nepraská a tlak vzduchu v ní je vždy roven okolnímu tlaku.

**Tenkostěnná láhev – teplota**

V případě, kdy láhev podrobíme pouze změně teploty, můžeme uvažovat izobarický děj s tlakem  $p_0$ , pro který platí Gay-Lussacův zákon v podobě

$$\frac{V_0}{T_k} = \frac{V_s}{T_s},$$

kde  $V_s$  je výsledný objem, který je roven

$$V_s = V_0 \frac{T_s}{T_k} \doteq 1,4451.$$

Objem se zmenšil o  $V_0 - V_s \doteq 0,55$  ml, tedy asi o 3,7 %.

**Tenkostěnná láhev – tlak**

Nyní nás zajímá, jaký vliv má na tenkostěnnou láhev samotná změna okolního atmosférického tlaku. Předpokládáme, že teplotu vzduchu v láhvi udržujeme stále na teplotě  $t_k = 26^\circ\text{C}$ . Pak lze tento případ modelovat izotermickým dějem, pro který platí Boyleův-Mariottův zákon

$$p_0 V_0 = p_{10} V_3.$$

Výsledný objem je  $V_3 = V_0 \frac{p_0}{p_{10}} \doteq 1,4941$ . Objem láhve se v tomto prípade změnil o  $V_0 - V_3 \doteq 6$  ml, tj. asi o 0,4 %.

### Závěr

Pro hrubou láhev vychází změna objemu 0,27 %, resp. 2,7 %, při změně tlaku, resp. teploty. Pro tenkostěnnou láhev jsou tyto hodnoty podobné, 0,4 %, resp. 3,7 %, při změně tlaku, resp. teploty. To znamená, že i pro tenkostěnnou láhev má zásadnější vliv na praskání změna teploty než změna tlaku.

## Úloha I.5 ... planetární osidlování

Nejspíše jste již někdy přemýšleli o tom, jestli neexistují nějaké mimozemské civilizace. Zpravidla čím větší hvězda je, tím větší má zářivý výkon a tím kratší má také svůj život. Zaměřme se nyní na to, že máme dvě hvězdy, z nichž jedna má dvojnásobný zářivý výkon co druhá. Pokud je pásmo, ve kterém je možný život, dáno teplotou, na které by se ustálilo dokonale černé těleso, a určitými dvěma teplotami (stejně pro jakoukoliv soustavu), kolem které hvězdy je širší pásmo, ve kterém by mohla být planeta se životem? Kolikrát bude větší oproti druhé hvězdě?

Teplota na planéte závisí od množství prijatej energie z hviezdy. Pre rovnakú planétu nám na udržanie tej istej teploty stačí zabezpečiť, aby malo žiarenie prijímané z hviezdy pri povrchu planéty rovnakú intenzitu  $F$ . Z toho, že na sféru s polomerom  $r$  (t.j. vzdialenosťou od stredu hviezdy) a teda povrchom  $4\pi r^2$  dopadá žiarenie s rovnakou intenzitou  $F$ , vyplýva

$$F = \frac{L}{4\pi r^2},$$

kde  $L$  je svietivosť izotropne vyžarujúcej hviezdy. Ak porovnáme intenzity žiarenia dvoch hviezd, ktorých svietivosť (resp. žiarivý výkon) je  $L_1$  a  $L_2 = 2L_1$ , postupne dostávame:

$$\begin{aligned} F_1 &= F_2, \\ \frac{L_1}{4\pi r_1^2} &= \frac{L_2}{4\pi r_2^2}, \\ \frac{L_1}{r_1^2} &= \frac{2L_1}{r_2^2}, \\ r_2^2 &= 2r_1^2, \\ r_2 &= \sqrt{2}r_1. \end{aligned}$$

Miesta, ktoré majú rovnaku teplotu, teda budú pri svietivejšej hviezde  $\sqrt{2}$ krát ďalej v porovnaní s menej svietivou hviezdou. Označme  $r_1^{\min}$ ,  $r_1^{\max}$ ,  $r_2^{\min}$ ,  $r_2^{\max}$  najmenšie a najväčšie vzdialenosti od hviezd 1 a 2, v ktorých je možný život, a príslušné šírky obývatelných zón  $\Delta r_1$ ,  $\Delta r_2$ . Potom

$$\Delta r_2 = r_2^{\max} - r_2^{\min} = \sqrt{2}r_1^{\max} - \sqrt{2}r_1^{\min} = \sqrt{2}(r_1^{\max} - r_1^{\min}) = \sqrt{2}\Delta r_1.$$

Vidíme teda, že pri 2krát svietivejšej hviezda bude šírka obývateľnej zóny  $\sqrt{2}$ krát väčšia. Problémom však je, že čím je hviezda svietivejšia, tým má kratší život. Preto pri hľadaní vhodných planét treba zvoliť kompromis.

## Úloha I.P . . . modýlek letadla na ISS

*Jak by se chovalo letadlo v mikrogravitaci (prostě uvažujte, že na něj gravitační síla nepůsobí)? Popište, jaký efekt by měla směrovka, výškovka, křídélka, případně vektorování tahu motorů. Jaké akrobatické manévry by byly možné? (Například plochá vývrтка asi ne.)*

Nejprve by bylo vhodné podívat se na to, jak letadlo vlastně létá a co od něj tedy můžeme očekávat v mikrogravitaci.

Letadlo se pohybuje vpřed díky tahové síle proudového motoru. Ten není nijak závislý na přítomnosti gravitační síly, tahová síla bude tedy i v mikrogravitaci nezměněná.

Další důležitou součástí letadla jsou křídla. Ta jsou tvarována tak, aby vzduch obtékal nad křídlem rychleji než pod ním. Nad křídlem tak vzniká podtlak a vztlaková síla tlačí letadlo vzhůru. Tato vztlaková síla se má za normálních okolností vyrovnat s gravitační silou. V mikrogravitaci však tuto gravitační sílu nemáme, takže křídla budou způsobovat stoupání letadla. Pro častější lety v mikrogravitaci by potom samozřejmě bylo vhodnější přejít k symetrickému tvaru křídel, na nichž by žádná vztlaková síla nevznikala.

Křídélka i směrové a výškové kormidlo ovládají letadlo tím, že mění aerodynamické síly působící na křídla letounu. Jejich vliv je proto závislý jen na obtékání letounu vzduchem a v mikrogravitaci by jejich funkce byla shodná s fungováním na Zemi.

Vektorování motoru znamená natáčení trysek v různých směrech, což mění směr tahové síly, která na stroj působí. Efekt vektorování zůstane v mikrogravitaci nezměněný.

Sečteno podtrženo, největší rozdíl při letu v mikrogravitaci by byla ničím narušená vztlaková síla na křídlech. Letadlo je však stále ovladatelné ve všech třech osách pomocí výškového a směrového kormidla a pomocí křídélek.

Nyní se podívejme na některé (nejen) akrobatické manévry:

### • Rovný let

Triviální let rovnoběžný s osou letadla by byl v mikrogravitaci velmi tuhým oříškem. Vztlaková síla vyvolaná speciálním tvarem křídla totiž není ničím vyrušená, a letadlo tedy při běžném letu stále lehce stoupá. V tomto případě nám ani výškovka letadla nepomůže, protože nahnutím celého letadla rovnoběžného letu s osou trupu nedosáhneme.<sup>7</sup> Jedinou možnou pomocí je pouze vektorování motoru proti vztlaku.

<sup>7</sup>Samozřejmě, že např. let rovnoběžně se zemí by byl normálně možný. Důležitá je fráze „rovnoběžný s osou letadla“.

- **Výkrut**

Do výkrutu se pilot na Zemi dostane lehkou výchylkou výškovky, která je následována výchylkou křidélek. Oba ovládací prvky fungují v mikrogravitaci shodně, proto je výkrut proveditelný i v mikrogravitaci.

- **Přemet**

Pro správné provedení přemetu nám stačí pouze dobře fungující výškovka. Přemet je tedy dalším možným akrobatickým manévrem.

- **Vývrтка**

Aby pilot navedl letadlo do vývrtky, potřebuje snížit rychlost letounu na pádovou. Letadlo se poté točí kontrolovaně do spirály při prakticky volném pádu. Pádu se nám ovšem v mikrogravitaci dosáhnout nepodaří, tento manévr je tedy na ISS neproveditelný.

- **Další**

Převážná většina leteckých akrobatických kousků je potom kombinací výše zmíněných, například kubánská osmička je složena z několika přemetů a vývrtek (tedy půjde), či Immelmann – půlobrátko s půlpřemetem (půjde). Obecně by šlo říct, že všechny akrobatické kousky, které používají pouze křídýlka, směrovku a výškovku vykonat půjdou, zatímco jakéhokoliv manévru využívajícího volný pád nebude možné dosáhnout (za zmínku zde stojí například hammerhead a, jak už sám název napoví, pád po ocase).

## Úloha II.1 ... Zuběnka

*Jak velké skladovací prostory by musela mít Víla Zubnička, aby mohla skladovat všechny mléčné zuby všech dětí? Resp. jakým tempem by její nároky na uskladnění rostly? Za jakou dobu by teoreticky měla ve svých skladech většinu zásob fosforu na Zemi?*

Úloha je pekne rozdelená na tri časti, je teda vhodné riešiť ich v zadanom poradí. Najprv sa teda pozrieme na to, ako veľké skladovacie priestory by musela mať Zubnička<sup>8</sup>. Rôzne zdroje uvádzajú rôzne čísla o tom, koľko ľudí žilo doposiaľ na Zemi, väčšina sa ale zhodne na číslo blízkom 100 miliardám. Orientovali sme sa wikipédiou<sup>9</sup> ktorá tvrdí, že na svete žilo od roku 50 000 p.n.l. 106 miliárd ľudí. Ďalšou vecou, na ktorú nesmieme zabúdať, je to, že až 8 % detí zomrie do veku 5 rokov. V minulosti to bolo určite oveľa viac, som presvedčený, že napríklad počas obdobia morovej epidémie v Európe bolo toto číslo väčšie ako 50 %. Preto tento fakt výrazne ovplyvní to, ako veľké budú sklady v súčasnosti. Ďalej rátajme, že každý z týchto ľudí mal 20 mliečnych zubov a Víla Zubnička ich všetky uložila (ide o horný odhad – víla radšej nebude riskovať, že nebude mať dosť priestoru).

<sup>8</sup>preklad pre Čechov: Zuběnka

<sup>9</sup>[https://cs.wikipedia.org/wiki/Sv%C4%9Btov%C3%A1\\_populace](https://cs.wikipedia.org/wiki/Sv%C4%9Btov%C3%A1_populace); z tohto zdroja sme čerpali aj ďalšie čísla spomínané v riešení



Ďalej rátaťme, že pri vhodnom naskladaní zaberie jeden mliečny zub objem približne  $0,25\text{ cm}^3$  – v literatúre sme žiadnu relevantnú informáciu o objeme zubov nenašli, odmerali sme teda priemerný objem niekoľkých mliečnych zubov (nesmieťme zabúdať, že mliečne zuby sú o niečo menšie ako trvalé a zároveň nemajú korene). V takom prípade budeme potrebovať úložné priestory o veľkosti  $530\,000\text{ m}^3$ . Ale ako sme povedali v predošlej časti, ich objem by bol pravdepodobne niekde medzi  $530\,000\text{ m}^3$  a  $265\,000\text{ m}^3$ . Ak uvažujeme jednoposchodové sklady, mohli by sme zuby skladovať do výšky napríklad 5 m; v takom prípade by sa sklady rozprestierali na ploche 5 až 10 hektárov.

Teraz sa pozrieme na to, akým tempom by rástli skladovacie nároky. Podľa OSN bol v roku 2016 nárast populácie 1,1 % (1,8 % sa narodí a 0,7 % zomrie). Aj keď nie je jasné, ako bude trend pokračovať, všetko nasvedčuje tomu, že za najbližších 100 rokov sa počet obyvateľov zeme ustáli na nejakej hodnote, rôzne zdroje to odhadujú na 6 až 12 miliárd obyvateľov. Nech je ako chce, znamená to, že za tých 100 rokov sa ročne bude rodiť a umierať približne rovnaký počet ľudí, čo pre Zubničku znamená, že bude každým rokom stabilne inkasovať konštantný počet zubov. Budeme rátať, že sa populácia ustáli na súčasných 7,4 miliárdach. Zároveň budeme rátať, že úmrtnosť detí klesne vďaka medicíne takmer na nulu. Môžeme ale predpokladať, že úmrtnosť bude rovnaká v budúcnosti ako súčasných 0,7 % z tohto počtu ročne, s tým rozdielom, že sa bude aj rodiť rovnaký počet ľudí, t.j. natalita bude tiež 0,7 %. Ročne teda pribudne na zemi asi 52 miliónov ľudí, čo sa pri dobudovávaní skladu s rovnakou výškou efektívne premietne do nutnosti zväčšiť jeho plochu o  $50\text{ m}^2$  za rok. Avšak takéto tempo zvyšovania kapacít bude dostatočné až v roku povedzme 2115, dovtedy bude pôrodnosť najprv rásť a potom sa zasa zmenšovať. Ak by sa dlhodobo udržala pôrodnosť na takej úrovni ako teraz, t.j. ročne sa narodí 1,8 % ľudí z toho, čo aktuálne žije, muselo by byť prvých niekoľko rokov od súčasnosti zvyšovanie priestoru skladov 2,5krát rýchlejšie, t.j. bolo by treba pridávať  $130\text{ m}^2$  skladovacích priestorov za rok navyše.

Treťou otázkou je, kedy by VÍla Zubnička vlastnila viac ako polovicu všetkých zásob fosforu na Zemi. Z wikipédie<sup>10</sup> vieme, že na Zemi je 1 210 ppm fosforu. To znamená, že viac ako tisícina hmotnosti Zeme je fosfor. My sa uspokojíme s odhadom  $1/1000$  – hmotnosť Zeme je  $5,97 \cdot 10^{24}$  kg, je na nej teda zhruba  $6 \cdot 10^{21}$  kg fosforu. Z ďalších zdrojov<sup>11</sup> vidíme, že fosfor tvorí asi 19 % hmotnosti zubu. Väzným vlastných mliečnych zubov sme stanovili, že priemerná hmotnosť zubu nie je väčšia ako 0,3 g. (Aj to je veľmi nadstretý odhad.) Z daných údajov vieme spočítať, že súčasné Zubničkine zásoby pokrývajú len zlomok svetových zásob fosforu, asi jednu  $5 \cdot 10^{13}$ -tinu. Ako sme si už povedali, najreálnejší model vývinu populácie je, že sa do 100 rokov ustáli na konečnej hodnote, v rozmedzí 6 a 12 miliárd. V predošlej časti sme ráтали s prírastkom populácie z dlhodobého hľadiska (to, čo sa bude diať za tých 100 rokov dovtedy nás nezaujíma, lebo je to kvôli dlhšej časovej škále zanedbateľné) 52 miliónov ľudí ročne, čo činí asi miliardu zubov ročne. Ak by týmto tempom navždy pokračoval nárast počtu mliečnych zubov, tak by Zubnička dosiahla väčšinový podiel za približne  $1 \cdot 10^{17}$  rokov. Čo len tak pre zaujímavosť

<sup>10</sup>[https://en.wikipedia.org/wiki/Abundance\\_of\\_the\\_chemical\\_elements](https://en.wikipedia.org/wiki/Abundance_of_the_chemical_elements)

<sup>11</sup><https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1263223/?page=2>

znamená, že ak by jej sklady zaberali plochu celej Zeme vrátane morí a oceánov (ľudia by sa museli uskromniť a žiť na streche skladov) tak by mali pri rovnakej hustote ukladania zubov (4 na mililiter) budovy skladov výšku 50 km.

## Úloha II.2 . . . irradie solární elektrárny

*Solární konstanta, či správněji solární irradie, je tok energie přicházející ze Slunce ve vzdálenosti Země od Slunce. Nejde o konstantu, ale uvažujme, že má hodnotu  $P = 1370 \text{ W} \cdot \text{m}^{-2}$ . Uvažujme, že Země obíhá Slunce po kružnici a sklon zemské osy vůči kolmici k její oběžné rovině je  $23,5^\circ$ . Jaký bude maximální výkon zachycený solárním panelem o ploše  $S = 1 \text{ m}^2$  o letním a zimním slunovratu, pokud panel leží na rovném povrchu Země v Praze? Uvažujte, že ani atmosféra ani budovy nijak neovlivní měření.*

Protože máme počítat s kruhovou orbitou Země a výkonový tok Slunce máme zadaný, stačí ho jen přenásobit plochou solárního panelu kolmou na paprsky. Tady si ale musíme uvědomit, že paprsky nebudou dopadat kolmo na plochu panelu, ale pod určitým úhlem. Ten závisí na zeměpisné šířce Prahy  $\vartheta_P$ , kterou můžeme zjistit například odtud <sup>12</sup>,  $\vartheta_P \approx 50^\circ$ . Víme, že při zimním slunovratu dopadají paprsky kolmo na  $\vartheta_z = -23,5^\circ$ <sup>13</sup>. Při letním slunovratu dopadají paprsky kolmo na  $\vartheta_1 = +23,5^\circ$ . Z geometrie vidíme, že úhel, pod kterým budou dopadat paprsky na pražský solární panel (úhel mezi tečnou k zemi a paprskem) vypočítáme z rovnice

$$\alpha_i = 90^\circ - |\vartheta_P - \vartheta_i|.$$

Dosazením získáme  $\alpha_z = 16,5^\circ$  a  $\alpha_1 = 63,5^\circ$ .

Pro maximální výkon počítejme se stoprocentní účinností. Plocha panelu kolmá na paprsky je  $S \sin \alpha_i$ , maximální výkon tedy spočítáme jako

$$P_i = PS \sin \alpha_i,$$

kde  $S$  označuje plochu solárního panelu. Číselným dosazením dostaneme  $P_1 = 1,23 \text{ kW}$  a  $P_z = 0,39 \text{ kW}$ . Můžeme si všimnout, že jen díky jinému postavení Slunce na obloze se sníží výkon solární elektrárny v zimě na třetinu jejího letního výkonu. V zimě také bývá vyšší oblačnost než v létě, což rozdíl mezi zimním a letním výkonem elektrárny ještě více prohlubuje.

## Úloha II.3 . . . pozorovací

*Jakou část povrchu kulové planety není možné vidět ze stacionární oběžné dráhy planety (taková dráha, že se obíhající objekt nachází stále nad stejným bodem na planetě), která má hustotu  $\rho$  a periodu rotace  $T$ ?*

Vo vztažnej sústave spojenej s rotujúcou planétou sa satelit na stacionárnej dráhe nepohybuje, preto naň pôsobí nulová výsledná sila  $F$ . Táto sila sa skladá z dvoch

<sup>12</sup>[https://cs.wikipedia.org/wiki/Seznam\\_měst\\_podle\\_zeměpisné\\_šířky](https://cs.wikipedia.org/wiki/Seznam_měst_podle_zeměpisné_šířky)

<sup>13</sup> Používáme konvenci, že rovník má  $0^\circ$ , severní šířky jsou kladné a jižní šířky záporné.

zložiek - gravitačnej sily  $F_g$  a odstredivej sily  $F_{od}$ , ktoré pôsobia navzájom opačným smerom:

$$0 = F = F_{od} - F_g = m\omega^2 r - \frac{GmM}{r^2}, \quad (2)$$

kde  $m$  je hmotnosť satelitu,  $\omega$  je uhlová rýchlosť rotácie planéty,  $r$  je vzdialenosť satelitu od stredu planéty,  $M$  je hmotnosť planéty a  $G$  je gravitačná konštanta.

Pre uhlovú rýchlosť platí

$$\omega = \frac{2\pi}{T},$$

z čoho po dosadení do (2) a úprave máme

$$\frac{T^2 GM}{4\pi^2} = r^3.$$

Hmotnosť planéty určíme z jej hustoty  $\rho$  a polomeru  $R$

$$M = \rho \frac{4}{3}\pi R^3.$$

Po dosadení a úprave dostávame vyjadrenie vzdialenosti satelitu od stredu planéty v jednotkách jej polomeru ako

$$\sqrt[3]{\frac{T^2 G \rho}{3\pi}} = \frac{r}{R}. \quad (3)$$

Zvyšok úlohy je už len geometria. Pozorovateľ na satelite totiž vidí časť povrchu planéty, ktorá je ohraničená kružnicou  $k$ , tvorenou bodmi dotyku dotyčníc vedených zo satelitu ku guľovému povrchu planéty. Zaujíma nás teda povrch guľového vrchlíka  $S$  (ktorý pozorovateľ vidí), pre ktorý platí<sup>14</sup>

$$S = 2\pi R^2 (1 - \cos \alpha),$$

kde  $\alpha$  je uhol satelit – stred planéty – bod na kružnici  $k$ .

Ramená tohto uhla majú dĺžky  $r$ , resp.  $R$  a v bode dotyku je pravý uhol, preto

$$\cos \alpha = \frac{R}{r}.$$

Po dosadení do (3) dostávame

$$S = 2\pi R^2 \left( 1 - \sqrt[3]{\frac{3\pi}{T^2 G \rho}} \right).$$

Zaujíma nás ale, akú časť  $k$  celého povrchu planéty pozorovateľ nevidí:

$$k = 1 - \frac{S}{4\pi R^2} = \frac{\left( 1 + \sqrt[3]{\frac{3\pi}{T^2 G \rho}} \right)}{2}.$$

Pre zaujímavosť, v prípade Zeme, ktorá má strednú hustotu  $\rho = 5,515 \cdot 10^3 \text{ kg}\cdot\text{m}^{-3}$  a rotačnú dobu  $T = 86,4 \cdot 10^3 \text{ s}$ , pozorovateľ nevidí  $k = 0,575$  jej povrchu.

<sup>14</sup>Dá sa to odvodiť integrovaním. Známý je tiež vzorec  $S = 2\pi R h$ , kde  $h = R - R \cos \alpha$  je výška vrchlíka. Všimnite si, že napr.  $\alpha = \pi$  dá celý povrch gule  $4\pi R^2$ .

## Úloha II.4 . . . jaderný odpad nikdy více

Představme si, že máme něco (například jaderný odpad) a chceme se toho zbavit. Těleso dostaneme na oběžnou dráhu Slunce shodnou s oběžnou dráhou Země, ale dostatečně daleko od Země, abychom mohli gravitační působení Země nadále zanedbávat. Otázka je, jaký způsob zbavení se inkriminovaného předmětu by nás stál kolik energie a který postup by byl tedy nejvýhodnější. Varianty jsou

- Hodit to do Slunce. Stačí, aby se to dostalo na sluneční povrch a bude to dostatečně usmažené.
- Převést to na kruhovou dráhu v Hlavním pásu (pás planetek mezi Marsem a Jupiterem).
- Vyhodit to zcela ze Sluneční soustavy.

Nebudeme uvažovat gravitační vliv ostatních planet Sluneční soustavy. Kdybychom povolili využití gravitačního praku (tj. manévru, při kterém je těleso urychleno nebo zpomaleno při průletu okolo planety, přičemž lze změnit i směr rychlosti), stačilo by pak těleso dopravit k nějaké planetě (Venuši nebo Marsu) a pomocí opakovaných gravitačních manévru bychom ho byli schopni dopravit téměř kamkoliv bez dalších energetických nákladů. V této úloze nám nejde o to, jak dlouho bude trvat přeprava tělesa, proto bychom mohli velmi dlouho čekat, dokud se určitá planeta nevyskytne v určité pozici a postupnými manévry (mezi nimiž mohou být staleté prostoje) náklad dostat do požadované pozice.

Označme si hmotnost tělesa  $m$ , hmotnost Slunce  $M$ , vzdálenost Země–Slunce  $R$  a počáteční rychlost (před výstřelem)  $v_0$ , kterou spočítáme z rovnosti odstředivého a gravitačního zrychlení

$$\frac{GM}{R^2} = \frac{v_0^2}{R} \quad \Rightarrow \quad v_0 = \sqrt{\frac{GM}{R}}.$$

Z energetického hlediska je pro nás nejvýhodnější rovnou těleso vystřelit z původní orbity. Kdybychom ho naložili na nějakou raketu, která by ho postupně urychlovala, tak by se zbytečně spotřebovávala energie na urychlování rakety a paliva a raketový motor nemůže mít ani teoreticky 100% účinnost. Uvažujeme tedy, že těleso odpálíme jednorázově z nějaké stanice, které má mnohem větší hmotnost, a proto můžeme předpokládat, že celková spotřebovaná energie odpovídá změně rychlosti tělesa

$$E = \frac{1}{2}m|\Delta\mathbf{v}|^2.$$

*Pryč ze Sluneční soustavy*

Naším cílem je uvést těleso na parabolickou dráhu tak, aby se nikdy nevrátilo zpět. Odletí od Slunce tak, že se bude neustále vzdalovat a zpomalovat, ale teoreticky se zastaví až v nekonečnu (za nekonečný čas). V nekonečnu bude mít nulovou potenciální i kinetickou energii. Ze zákona zachování energie tedy vyplývá, že bychom ho měli urychlit na rychlost  $v_1$  tak, aby jeho celková energie byla nulová

$$\frac{1}{2}mv_1^2 - \frac{GmM}{R} = 0,$$

kde jsme využili vztah pro potenciální energii v radiálním gravitačním poli. Výsledná rychlost je

$$v_1 = \sqrt{\frac{2GM}{R}}.$$

Této rychlosti dosáhneme nejnázde (při minimálním  $|\Delta \mathbf{v}| = |\mathbf{v}_1 - \mathbf{v}_0|$ ) tak, že těleso urychlíme ve směru jeho rychlosti, tedy ve směru tečném k oběžné dráze. Potřebujeme tedy zvýšit jeho rychlost o

$$\Delta v_1 = v_1 - v_0 = (\sqrt{2} - 1) \sqrt{\frac{GM}{R}} \doteq 12,4 \text{ km} \cdot \text{s}^{-1}. \quad (4)$$

Této rychlosti se mimochodem někdy říká 3. kosmická rychlost. Potřebná energie je

$$E_1 = (3 - 2\sqrt{2}) \frac{GMm}{2R}.$$

### Hlavní pás

Chceme-li těleso dostat na kruhovou oběžnou dráhu v Hlavním pásu, bude potřeba provést dva manévry. Nejprve těleso převedeme na eliptickou dráhu (tzv. Hohmannova elipsa) s periheliem na oběžné dráze Země a afeliem v Hlavním pásu. Až se dostaneme do afelia, zvýšíme rychlost tak, aby těleso zůstalo na kruhové dráze v této vzdálenosti od Slunce. Tomuto přechodu mezi dvěma orbitami se říká Hohmannova trajektorie.

Problém je v tom, že nemůžeme těleso jen jednou urychlit (vystřelit), ale potřebujeme manévrovat dvakrát. Takže si musíme vyslat těleso i spolu s nějakou raketou, motorem, který ho pak může převést na kruhovou dráhu.

Poloměr Hlavního pásu označíme  $R_A$ . Chceme spočítat rychlost  $v_3$ , na kterou těleso musíme urychlit, aby se pohybovalo po zmíněné eliptické dráze. Vyjdeme ze zákona zachování energie pro stav v periheliu a afeliu

$$\frac{1}{2} m' v_3^2 - \frac{GMm'}{R} = \frac{1}{2} m' v_A^2 - \frac{GMm'}{R_A},$$

kde  $m'$  je hmotnost tělesa i s raketou a  $v_A$  je rychlost v afeliu, kterou můžeme vyjádřit z Keplerova zákona

$$v_A R_A = v_3 R.$$

Vyjádřením  $v_3$  z těchto dvou rovnic dostáváme

$$v_3 = \sqrt{\frac{2GM R_A}{R(R_A + R)}}.$$

Nyní, když máme těleso na eliptické dráze, musíme v afeliu zvýšit jeho rychlost

$$v_A = \sqrt{\frac{2GMR}{R_A(R_A + R)}}$$

na rychlost

$$v_4 = \sqrt{\frac{GM}{R_A}}.$$

Postupné změny rychlostí (znaménko jsme volili tak, aby  $\Delta v_3, \Delta v_4 > 0$ ) jsou tedy

$$\Delta v_3 = \sqrt{\frac{GM}{R}} \left( \sqrt{\frac{2R_A}{R_A + R}} - 1 \right),$$

$$\Delta v_4 = \sqrt{\frac{GM}{R_A}} \left( 1 - \sqrt{\frac{2R}{R + R_A}} \right).$$

Uvažujme, že těleso společně s palivem a další zátěží o hmotnosti  $m_2 = m' - m$  vyšleme z původní orbity Slunce na eliptickou dráhu. Na to je potřeba energie

$$E_{3a} = \frac{1}{2}(m + m_2)\Delta v_3^2.$$

Když se vyslaný objekt dostane do afelia, potřebujeme jeho rychlost zvýšit o  $\Delta v_4$ . Ale to nepůjde jen tak. Abychom těleso urychlili, musíme podle zákona zachování hybnosti něco jiného zpomalit (nebo urychlit opačným směrem). To by se dalo vyřešit zapnutím raketového motoru, který vysílá urychlené palivo opačným směrem. Z energetického hlediska bude nejvýhodnější, když všechno palivo vyšleme určitou rychlostí  $\Delta u_2$  najednou. Skutečný motor by měl menší účinnost, ale následujícím způsobem lze spočítat alespoň teoretické minimum potřebné energie.

Ze zákona zachování hybnosti

$$\Delta v_4 m = \Delta u_2 m_2,$$

$$E_{3b} = \frac{1}{2} m \Delta v_4^2 + \frac{1}{2} m_2 \Delta u_2^2 = \frac{1}{2} m \Delta v_4^2 + \frac{m^2}{2m_2} \Delta v_4^2.$$

My hledáme takovou hmotnost  $m_2$ , aby celková spotřebovaná energie  $E_3 = E_{3a} + E_{3b}$  byla co nejmenší. To najdeme tak, že derivaci  $E_3$  podle  $m_2$  položíme rovnou nule.

$$\frac{dE_3}{dm_2} = -\frac{m^2}{2m_2^2} \Delta v_4^2 + \frac{1}{2} \Delta v_3^2 = 0,$$

$$\frac{m^2}{2m_2^2} \Delta v_4^2 = \frac{1}{2} \Delta v_3^2,$$

$$m^2 \Delta v_4^2 = m_2^2 \Delta v_3^2,$$

$$m_2 = m \frac{\Delta v_4}{\Delta v_3}.$$

Jde skutečně o minimum, protože v extrémních případech pro velmi malé nebo velmi velké  $m_2$  jde  $E_3$  k nekonečnu a tento výsledek je jakýsi kompromis mezi těmito extrémy. Po dosažení můžeme počítat energii  $E_3$

$$E_3 = \frac{1}{2}(m + m_2)\Delta v_3^2 + \frac{1}{2}m\Delta v_4^2 + \frac{m^2}{2m_2}\Delta v_4^2,$$

$$E_3 = \frac{1}{2}m \left(1 + \frac{\Delta v_4}{\Delta v_3}\right) \Delta v_3^2 + \frac{1}{2}m\Delta v_4^2 + \frac{1}{2}m\Delta v_3\Delta v_4 = \frac{1}{2}m (\Delta v_3 + \Delta v_4)^2.$$

Vyšlo nám tedy, že potřebujeme minimálně takovou energii, která by stačila na urychlení tělesa na rychlost  $\Delta v_{34} = \Delta v_3 + \Delta v_4$ . Tomu se říká delta- $v$  budget, což je celkový součet změn rychlosti během manévru, a to určuje celkovou potřebnou energii. Součet změn rychlosti je

$$\begin{aligned} \Delta v_{34} = \Delta v_3 + \Delta v_4 &= \sqrt{\frac{GM}{R}} \left( \sqrt{\frac{2R_A}{R_A + R}} - 1 \right) + \sqrt{\frac{GM}{R_A}} \left( 1 - \sqrt{\frac{2R}{R + R_A}} \right) = \\ &= \sqrt{\frac{GM}{RR_A(R_A + R)}} \left( \sqrt{2}(R_A - R) + \left( \sqrt{R} - \sqrt{R_A} \right) \sqrt{R_A + R} \right). \end{aligned}$$

Je známo<sup>15</sup>, že se Hlavní pás nachází ve vzdálenosti od 2AU až po 4AU, tedy  $R_{A\min} = 2R$  a  $R_{A\max} = 4R$ .

$$\begin{aligned} \Delta v_{34\min} &= \left( \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{2}} - 1 \right) \sqrt{\frac{GM}{R}} \doteq 8,46 \text{ km}\cdot\text{s}^{-1}, \\ \Delta v_{34\max} &= \left( \frac{3}{\sqrt{10}} - \frac{1}{2} \right) \sqrt{\frac{GM}{R}} \doteq 13,3 \text{ km}\cdot\text{s}^{-1}. \end{aligned}$$

Z energetického hlediska je tedy nejvýhodnější vyslat těleso na bližší okraj Hlavního pásu. Potřebná energie pak je

$$E_{3\min} = \left( \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{2}} - 1 \right)^2 \frac{GMm}{2R}.$$

Tento způsob vyžaduje nejméně energie ze všech tří možností, i když je relativně složitý.

*Poznámka* Někdo by mohl navrhnout použití přechodu pomocí dvou půlelips (tzv. „bi-elliptic transfer“), který spočívá v uvedení tělesa na eliptickou dráhu, jejíž hlavní poloosa je řádově větší, než je poloměr požadované finální orbity v Hlavním pásu. Následně je těleso v afeliu, kde se pohybuje malou rychlostí, zrychleno (na to není potřeba velké množství energie) tak, aby se perihelium nové eliptické trajektorie nacházelo ve vzdálenosti  $R_A$  od Slunce. V periheliu je pak těleso zpomalené,

<sup>15</sup>[https://cs.wikipedia.org/wiki/Hlavn%C3%AD\\_p%C3%A1s](https://cs.wikipedia.org/wiki/Hlavn%C3%AD_p%C3%A1s)

aby zůstalo na kruhové oběžné dráze. Tato trajektorie může být v některých případech opravdu energeticky výhodnější, než je Hohmannova trajektorie. Ukazuje se ale<sup>16</sup>, že poměr poloměru konečné a původní trajektorie musí být alespoň 11,94 nebo větší. Takže v našem případě by se to nevyplatilo.

### Do Slunce

V předchozích případech jsme těleso urychlovali, teď se nabízí ho zpomalit a nechat ho „spadnout“ do Slunce. Konkrétně ho stačí zpomalit tak, aby jeho eliptická trajektorie měla afelium na původní trajektorii a perihelium na obrácené straně povrchu Slunce (jiné trajektorie vyřešíme později). Při svém pohybu tedy jen „škrtně“ o povrch Slunce, ale to stačí na pohlčení. Označme  $R_S$  poloměr Slunce. Znovu použijeme zákon zachování energie.

$$\frac{1}{2}mv_P^2 - \frac{GMm}{R_S} = \frac{1}{2}mv_2^2 - \frac{GMm}{R},$$

kde  $v_2$  je rychlost, na kterou jsme těleso zpomalili a  $v_P$  je rychlost v periheliu, kterou vyjádříme pomocí druhého Keplerova zákona

$$v_P = \frac{v_2 R}{R_S}.$$

Po dosazení dostáváme

$$\begin{aligned} \frac{v_2^2 R^2}{2R_S^2} - \frac{GM}{R_S} &= \frac{1}{2}v_2^2 - \frac{GM}{R}, \\ v_2 &= \sqrt{\frac{2GMR_S(R - R_S)}{R(R^2 - R_S^2)}} = \sqrt{\frac{2GMR_S}{R(R + R_S)}}. \end{aligned}$$

Rychlost potřebujeme změnit o

$$\Delta v_2 = v_0 - v_2 = \left(1 - \sqrt{\frac{2R_S}{R + R_S}}\right) \sqrt{\frac{GM}{R}} \approx \left(1 - \sqrt{\frac{2R_S}{R}}\right) \sqrt{\frac{GM}{R}}.$$

Při porovnání s (4) vidíme, že pokud platí

$$1 - \sqrt{\frac{2R_S}{R + R_S}} > \sqrt{2} - 1,$$

odeslání tělesa ven ze Sluneční soustavy je výhodnější, než by bylo jeho odeslání do Slunce. Po úpravě nerovnice dostáváme

$$R/R_S > \frac{2}{\sqrt{2} - 1},$$

což platí vždy pro  $R_S \ll R$ .

<sup>16</sup>[https://en.wikipedia.org/wiki/Bi-elliptic\\_transfer](https://en.wikipedia.org/wiki/Bi-elliptic_transfer)



Existují ale i jiné, energeticky výhodnější způsoby, jak poslat těleso do Slunce. Například můžeme poslat těleso do nekonečna (tedy pryč ze Sluneční soustavy) a tam ho o malinko zpomalit. Vhodně zvolené (nekonečně malé) zpomalení způsobí, že jeho celková energie bude záporná a Slunce si ho tedy přitáhne zpět a pohltí. Sice spotřebujeme mnohem menší množství energie (stejně, jako v prvním případě – výstřelu do nekonečna), problém této metody ale spočívá v tom, že zabere nekonečně dlouhý čas.

Zajímá-li nás pouze minimální potřebná energie (a ne čas), je naše delta- $v$ :

$$v_1 = (\sqrt{2} - 1) \sqrt{\frac{GM}{R}}.$$

Chceme-li těleso spálit Sluncem rychleji, stačí ho urychlit o něco menší rychlostí než  $v_1$ . Tím se dostane na eliptickou trajektorii, jejíž afelium bude velmi daleko, je-li rychlost jen o málo menší než  $v_1$ . V afeliu pak těleso zpomalíme na (skoro) nulovou rychlost a těleso tak prakticky spadne do Slunce. Na to ale spotřebujeme více energie, než by stačilo na samotný výstřel do nekonečna.

Čím rychleji ho budeme chtít dostat do Slunce, tím větší delta- $v$  bude potřeba a tím více energeticky náročný náš manévr bude. V extrémním případě můžeme těleso vystřelit vysokou rychlostí směrem ke Slunci (resp. kousek stranou, aby se tangenciální složka rychlosti odečetla), čímž by se dostalo do Slunce velmi rychle.

Tvrzení, že pro hod do Slunce je nejefektivnější vyslání tělesa do nekonečna a následné zpomalení, je dokázáno v další podkapitole.

### *A nevyplatilo by se střílet šikmo?*

Pojďme případ hození do Slunce vyřešit obecněji. Budeme se nyní zabývat jen bieliptickým manévrem, který je jednodušší a na teoretický popis nám bude stačit. Jiné typy manévrů zahrnují i konstrukci rakety, účinnost jejích motorů, která může záviset na jejich výkonu, a další parametry.

Máme těleso, které obíhá Slunce ve vzdálenosti  $R$  rychlostí  $v_0$ . Rychlost si rozdělíme na tři vzájemně kolmé složky

- tečná – ve směru pohybu oběhu okolo Slunce
- radiální – ve směru od Slunce
- axiální – ve směru kolmém na rovinu oběhu

Počáteční rychlost je v tečném směru  $v_0$ , v ostatních směrech je nulová. V okamžiku po výstřelu si složky označíme po řadě  $v_t, v_r, v_a$ . Těleso tak navedeme na eliptickou dráhu a v jejím afeliu ho zpomalíme tak, aby spadlo do Slunce. Poloměr Slunce je velmi malý v porovnání s poloměrem oběhu Země, jenž je menší, než hlavní poloosa vzniklé elipsy. Proto můžeme předpokládat, že těleso v afeliu úplně zastavíme. Pak začne padat přímo do středu Slunce.

Celá naše  $\Delta v$  spotřeba pak bude

$$\Delta v = v_A + \sqrt{(v_t - v_0)^2 + v_r^2 + v_a^2}, \quad (5)$$

kde  $v_A$  je rychlost v afeliu. Tuto spotřebu se snažíme minimalizovat.

Zamyslíme-li se nad tím, jak axiální rychlost ovlivňuje naši trajektorii, tak zjistíme, že jen „naklání“ rovinu elipsy. Potom si stačí změnit souřadnou soustavu a pozorovat pohyb v rovině nakloněné elipsy. Radiální složka v nové soustavě zůstane stejná, axiální složka je nulová a tečná složka je součet původní radiální a tečné

$$v'_t = \sqrt{v_t^2 + v_a^2}$$

Urychlili jsme-li těleso v axiálním směru, způsobili jsme tak jen náklon rotace a zvýšení tečné rychlosti. Stejně jako při výstřelu ze Sluneční soustavy ale víme, že se nám nevyplatí urychlovat těleso v axiálním směru. Dále tedy budeme počítat s  $v_a = 0$ . Taky si můžeme rozmyslet, že stačí uvažovat  $v_t \geq 0$ .

Budeme znovu vycházet ze zákona zachování momentu hybnosti a zákona zachování energie

$$\begin{aligned} v_t R &= v_A R_A, \\ v_t^2 + v_r^2 - 2K &= v_A^2 - 2K \frac{R}{R_A}, \end{aligned}$$

kde je vzdálenost v afeliu a  $v_A$  rychlost v afeliu. V ZZK jsme rovnou vypustili hmotnost tělesa a zavedli substituci  $K = \frac{GM}{R}$ .  $R_A$  můžeme z rovnic eliminovat

$$v_A^2 - 2K \frac{v_A}{v_t} - v_t^2 - v_r^2 + 2K = 0.$$

Důležité je, že náš model platí pro eliptický pohyb, tedy dodaná rychlost nesmí být větší než úniková. Vyjádříme  $v_A$  a dosadíme do (5)

$$\begin{aligned} v_A &= \frac{K}{v_t} - \sqrt{\frac{K^2}{v_t^2} + v_t^2 + v_r^2 - 2K}, \\ \Delta v &= \frac{K}{v_t} - \sqrt{\frac{K^2}{v_t^2} + v_t^2 + v_r^2 - 2K} + \sqrt{(v_t - v_0)^2 + v_r^2}, \end{aligned}$$

kde jsme při řešení kvadratické rovnice zvolili pouze menší řešení (s mínusem před odmocninou), druhé řešení by udávalo rychlost v periheliu (která je vyšší).

Hledáme minimum funkce  $\Delta v$ . Problém je v tom, že závisí na dvou nezávislých parametrech  $v_t$  a  $v_r$ . To uděláme pomocí parciálních derivací

$$\begin{aligned} \frac{\partial \Delta v}{\partial v_r} &= -\frac{v_r}{\sqrt{\frac{K^2}{v_t^2} + v_t^2 + v_r^2 - 2K}} + \frac{v_r}{\sqrt{(v_t - v_0)^2 + v_r^2}} = 0, \\ \frac{v_r}{\sqrt{\frac{K^2}{v_t^2} + v_t^2 + v_r^2 - 2K}} &= \frac{v_r}{\sqrt{(v_t - v_0)^2 + v_r^2}}. \end{aligned}$$

Vidíme, že jedno řešení je  $v_r = 0$ . Za  $v_0$  můžeme dosadit  $\sqrt{K}$  a pokračovat v úpravách

$$\begin{aligned} \sqrt{\left(\frac{K}{v_t} - v_t\right)^2 + v_r^2} &= \sqrt{(v_t - \sqrt{K})^2 + v_r^2}, \\ \left|\frac{K - v_t^2}{v_t}\right| &= |v_t - \sqrt{K}|, \\ \frac{|\sqrt{K} - v_t||\sqrt{K} + v_t|}{v_t} &= |\sqrt{K} - v_t|, \\ v_t &= \sqrt{K} = v_0. \end{aligned}$$

Docházíme tedy k tomu, že pro  $v_t = v_0$  je řešením podmínky nulové parciální derivace libovolné  $v_r$ . To odpovídá stavu, kdy těleso vystřelíme kolmo na původní kruhovou dráhu; dosazením do  $\Delta v$  vidíme, že  $\Delta v = v_0$ . Pro jiné  $v_t$  může minimum nastat jen pro  $v_r = 0$  nebo maximální  $v_r$  (tedy výstřel do nekonečna, který jsme už vyřešili).

Zjistili jsme tedy, že je nejvýhodnější nestřílet „šikmo“ a těleso urychlovat pouze v tečném směru. Vztah pro  $\Delta v$  se nám znovu zjednoduší

$$\Delta v = \frac{K}{v_t} - \left|\frac{K}{v_t} - v_t\right| + |v_t - \sqrt{K}|.$$

Řešíme dva případy,  $v_t < \sqrt{K}$  a  $v_t > \sqrt{K}$ . Pro  $v_t < \sqrt{K}$  je afelium na původní dráze, proto tento případ nemusíme uvažovat. Pro  $v_t > \sqrt{K}$  dostaneme

$$\Delta v = \frac{K}{v_t} + \frac{K}{v_t} - v_t + v_t - \sqrt{K} = \frac{2K}{v_t} - \sqrt{K}.$$

To znamená, že čím vyšší rychlost  $v_t$ , tím méně bude potřeba energie na manévry. Jsme ovšem omezeni podmínkou, že rychlost nesmí být vyšší, než je rychlost úniková, jinak náš model selže a může vycházet záporná  $\Delta v$ . Nejeefektivnější bude znovu výstřel do nekonečna.

Pojďme se ještě zamyslet nad tím, co kdybychom nepoužili bieliptický přesun, ale pouze bychom těleso z oběžné dráhy vystřelili tak, že perihelium jeho trajektorie bude uvnitř Slunce (resp. na jeho okraji ve vzdálenosti  $R_S$ , protože určitě bude výhodnější těleso zpomalit/urychlit tak, aby se dostalo na povrch Slunce, než někam dovnitř). Stejnou úvahou, jako v předchozím případě, dojdeme k tomu, že nemá smysl dodávat tělesu axiální složku rychlosti. Znovu vyjdeme z rovnic pro zákon zachování energie a momentu hybnosti

$$\begin{aligned} v_P^2 - 2K \frac{R}{R_S} &= v_t^2 + v_r^2 - 2K, \\ v_t R &= v_P R_S. \end{aligned}$$

Označíme poměr  $R/R_S = r$  a eliminujeme rychlost v periheliu  $v_P$

$$v_r^2 = v_t^2 r^2 - 2Kr - v_t^2 + 2K. \quad (6)$$

Dosadíme do vztahu pro  $\Delta v$ , který nyní nezahrnuje rychlost  $v_A$ , protože nepotřebujeme nikde brzdit

$$\Delta v = \sqrt{(v_t - \sqrt{K})^2 + v_t^2},$$

$$\Delta v = \sqrt{(v_t - \sqrt{K})^2 + v_t^2 r^2 - 2Kr - v_t^2 + 2K} = \sqrt{v_t^2 r^2 - 2v_t \sqrt{K} + (3 - 2r)K}.$$

Hledáme nyní takovou rychlost  $v_t$ , pro kterou je  $\Delta v$  nejmenší. Výraz pod odmocninou je kvadratická funkce, víme tedy, že minimum má pro  $v_t = \sqrt{K}r^{-2} = v_0 r^{-2}$ . Toto minimum je ale vždy záporné; podíváme-li se ale na rovnici (6), zjistíme, že není splněna podmínka řešitelnosti pro  $v_t$  ( $v_t^2$  vychází taky záporné).

Fyzikální význam je takový, že ZZE určuje minimální možnou rychlost v periheliu a tedy i minimální možnou hodnotu tečné složky rychlosti; pokud zvolíme tečnou rychlost menší, neexistuje dráha s periheliem na povrchu Slunce. Rychlost  $v_t$  tedy musíme zvýšit z hodnoty  $v_0 r^{-2}$  tak, aby  $v_t^2$  bylo nezáporné. Protože minimalizujeme kvadratickou funkci, víme, že je optimální zvýšit  $v_t$  minimálně, čímž dostaneme znovu  $v_t = 0$ . To je případ, který jsme počítali výše, kdy těleso pouze (tečně) zpomalíme a necháme ho spadnout po elipse.

Víme ale, že výhodnější zůstává stejně případ vyhození do nekonečna a ještě výhodnější vyhození do Hlavního pásu.

## Úloha II.5 ... skleněný déšť

Dělník si na stavbu mrakodrapu přinesl vak se skleněnkami, aby se s nimi mohl pochlubit svým kolegům. A co se nestane – vak se vysype a kuličky padají skrze lešení směrem k zemi. Lešení se skládá z jednotlivých poschodí o výšce  $h$ . Podlaha každého poschodí se skládá ze stejných mříží, ve kterých díry zaujímají  $k\%$  z celkové plochy mříže. Uvažujeme zjednodušený model propadávání kuliček lešením, kdy, pokud kulička spadne na díru v lešení, tak projde bez ovlivnění, a pokud spadne na pevnou část mříže, tak se její rychlost sníží na 0 a ihned začne dále padat (tj. velikost kuliček je zanedbatelná vůči velikosti děr v lešení, kuličky se od lešení nijak neodráží a po dopadu na pevnou část mříže se ihned skutálí do díry a dále začínají padat). Nakonec neuvažujme ani potenciální srážky kuliček mezi sebou. Předpokládejte, že kuličky se z tašky sypou s konstantním hmotnostním průtokem  $Q$ . Jakou silou budou kuličky působit na každé patro lešení, až se situace ustálí?

Patro, ve kterém dělník upustí kuličky, označíme číslem 0. Zároveň předpokládáme, že počáteční rychlost všech kuliček je nulová. Pro maximální rychlost kuličky v  $n$ tém patře (v žádném z předchozích pater se nezpomalila o lešení) potom platí rovnice

$$\begin{aligned} v_n &= gt_n, \\ nh &= \frac{1}{2}gt_n^2, \end{aligned}$$

jejichž řešením je vztah  $v_n = \sqrt{2ngh}$ .

Každá kulička může v každém patře buď narazit na lešení, nebo propadnout dírou a pokračovat dál. To znamená, že její cestu až do  $n$ tého patra můžeme jednoznačně popsat pomocí řetězce jedniček a nul o délce  $n - 1$ . Například 0100 bude znamenat, že kulička v prvním patře propadla dírou, ve druhém narazila na mříž, ve třetím znovu propadla a ve čtvrtém také. Do  $n$ tého patra se tak každá kulička mohla dostat  $2^{n-1}$  způsoby. Vydělme  $k$  stovkou; pravděpodobnost, že kulička propadne patrem, pak je  $k$ , zatímco pravděpodobnost, že narazí, je  $(1 - k)$ . Pro pravděpodobnost cesty do  $n$ tého patra ve tvaru 0100 tak platí  $P(0100) = k(1 - k)k \cdot k = k^3(1 - k)$ . V praxi to znamená, že cestou 0100 se vydá hmotnostní tok  $Q_{0100} = P(0100)Q = k^3(1 - k)Q$ .

Počet nul na konci zápisu cesty kuličky do  $n$ tého patra označíme  $d$ . V našem případě tedy platí  $d(0100) = 2$ . Odtud vidíme, že daná kulička, která dopadla do  $n$ tého patra touto cestou, narazila naposledy přesně v  $(n - d - 1)$ tém patře. Její rychlost v  $n$ tém patře tedy bude  $v_{d+1}$ . Je zřejmé, že  $d \in \langle 0, n - 1 \rangle$ .

Jestliže kulička narazí, veškerá její hybnost se změní na nulu. Pokud tedy za čas  $t$  dopadnou na patro lešení kuličky s celkovou hmotností  $m = Q't$  a rychlostí  $v$ , působí tím silou

$$F = m \frac{v}{t} = Q'v.$$

Hmotnostní tok pro kuličky popsané řetězcem  $i$  je rovný  $Q' = QP(i)$ .

Nyní už dokážeme popsat všechny možné cesty kuliček do  $n$ tého patra a víme, jaký hmotnostní tok jimi bude proudit. Zároveň umíme vyjádřit rychlost, kterou se dané kuličky budou pohybovat, a víme, jak z toho spočítat výslednou sílu. Ještě je potřeba dodat, že jen  $(1 - k)$  kuliček se v  $n$ tém patře zastaví, takže celkovou sílu působící na dané patro můžeme spočítat sumou

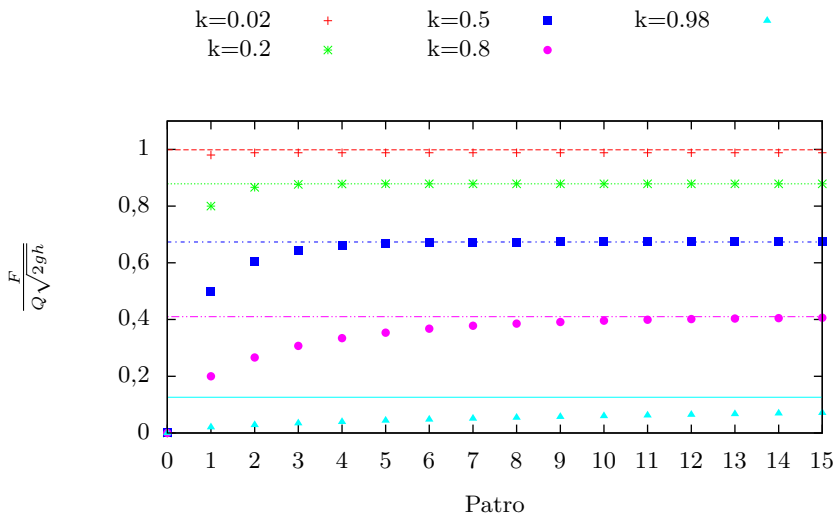
$$F_n = Q(1 - k) \sum_{i \in M_{n-1}} P(i) v_{d(i)+1}, \quad (7)$$

kde  $M_{n-1}$  je množina všech řetězců jedniček a nul s délkou  $n - 1$ .

Tato suma obsahuje celkem  $2^{n-1}$  sčítanců. Rozdělíme si je na  $n$  skupin tak, aby všechny členy v jedné skupině měly stejnou hodnotu  $d$ . Díky tomu můžeme člen  $v_{d+1}$  z každé skupiny vytknout a potom počítat jen sumu členů  $P(i)$  v dané skupině. Všechny řetězce z jedné skupiny mají  $n - 1$  číslic a končí na číslici 1, následovanou  $d$  nulami.<sup>17</sup> To znamená, že ještě nemáme určených prvních  $n - d - 2$  číslic. V každé skupině budou přítomny jejich všechny možné kombinace. Sumu všech členů z jedné skupiny tak můžeme vyjádřit výrazem

$$S(d) = v_{d+1}(1 - k)k^d \sum_{j \in M_{n-d-2}} P(j), \quad (8)$$

kteří ovšem neplatí pro skupinu s  $d = n - 1$ . Ta obsahuje jen samé nuly, takže pro ni platí  $S(n - 1) = v_n k^{n-1}$ .



Obr. 2: Síla působící na jednotlivá patra pro různé hodnoty  $k$ . Vodorovné přímky ukazují teoretickou hodnotu, ke které síla konverguje v nekonečném patře.

Připomeňme si, že množina  $M_{n-d-2}$  obsahuje všechny možné řetězce jedniček a nul o celkové délce  $n-d-2$ . Pro každý řetězec  $j$  z této množiny potom platí  $P(j) = k^\alpha (1-k)^\beta$ , kde  $\alpha$  je počet nul v této sekvenci a  $\beta = n-d-2-\alpha$  je počet jedniček v této sekvenci. Jestliže známe binomickou větu, neměla by nás překvapit rovnost

$$\sum_{j \in M_{n-d-2}} P(j) = (k + (1-k))^{n-d-2} = 1,$$

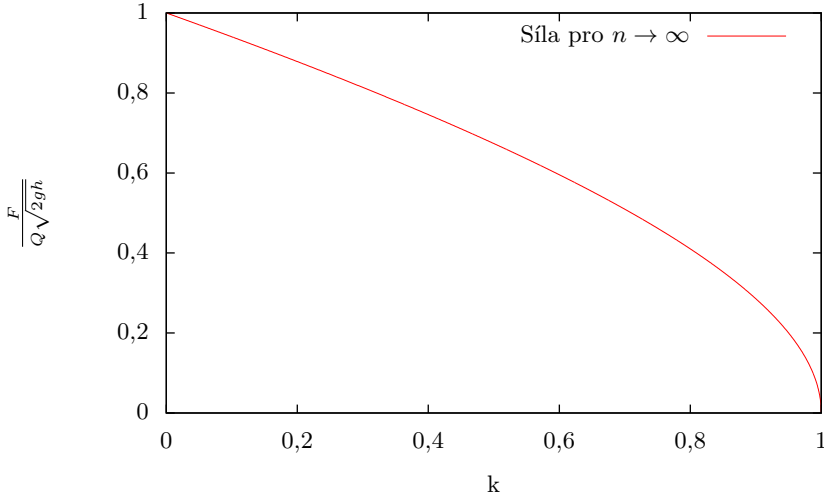
která nám umožní přepsat vzorec (8) do tvaru  $S(d) = v_{d+1} (1-k) k^d$  pro  $d \neq n-1$  a  $S(n-1) = v_n k^{n-1}$ .

Všechny členy ze sumy ve vzorci (7) jsme si rozdělili do několika skupin podle  $d$  a poté jsme spočítali součet všech členů v každé této skupině. Celkovou sumu z (7) tak můžeme nahradit sumou těchto jednotlivých součtů a dostaneme

$$F_n = Q(1-k) \sum_{d=0}^{n-1} S(d) = Q(1-k) \left( v_n k^{n-1} + (1-k) \sum_{d=0}^{n-2} v_{d+1} k^d \right)$$

$$F_n = Q\sqrt{2gh}(1-k) \left( k^{n-1}\sqrt{n} + (1-k) \sum_{i=1}^{n-1} k^{i-1}\sqrt{i} \right).$$

<sup>17</sup>S výjimkou skupiny s  $d = n-1$ , ve které je jen jeden řetězec tvořený samými nulami.

Obr. 3: Síla působící na patro  $n \rightarrow \infty$  pro různé hodnoty  $k$ .

Tento vzorec je řešením úlohy, protože sumu v něm obsaženou už není možné dále zjednodušit<sup>18</sup>.

Ačkoli se to na první pohled nezdá, tento výsledek je velmi intuitivní a dává smysl: první člen v závorce je případ, kdy kulička ani jednou nenarazí a dopadne až na  $n$ té poschodí. Její hybnost je tedy úměrná  $\sqrt{n}$ . Druhý člen představuje případy, ve kterých kulička padá přímo z výšky  $i$  nad  $n$ tým poschodím. Aby spadla z výšky  $i$ , musela překonat  $i - 1$  poschodí (proto se v sumě vyskytuje činitel  $k^{i-1}$ ) a dopadnout na poschodí předtím, z čehož máme  $(k - 1)$ . Její hybnost je potom úměrná  $\sqrt{i}$ . Pro kontrolu, pokud ze závorky vypustíme členy s odmocninami (takže nebudeme počítat součet hybností, ale jen součet hmotností), dostaneme v závorce výraz

$$k^{n-1} + (1 - k) \sum_{i=1}^{n-1} k^{i-1} = k^{n-1} + (1 - k) \frac{k^{n-1} - 1}{k - 1} = 1.$$

To je ale jen jinak vyjádřený fakt, že se na  $n$ té poschodí (a tím pádem na každé poschodí) dostanou všechny kuličky.

Můžeme si všimnout další zajímavé skutečnosti – pro velká  $n$  suma konverguje ke kladnému reálnému číslu, zatímco výraz  $k^{n-1}\sqrt{n}$  konverguje k nule. Díky tomu se i  $F_n$  blíží k nějaké konkrétní hodnotě, což je přesně to, co bychom z fyzikálního hlediska čekali. Tohoto chování si můžeme všimnout v grafu 2 pro různé hodnoty  $k$ .

<sup>18</sup>Ize ji pouze vyjádřit pomocí speciálních funkcí

## Úloha II.P ... ó Oganesson

*Jaké vlastnosti má 118. prvek periodické soustavy prvků? Respektive jaké by asi měl, kdyby byl stabilní? Diskutujte alespoň tři fyzikální vlastnosti.*

*Úvod, radioaktivita, elektronová konfigurace a reaktivita Og*

Přestože Wikipedie není nezpochybnitelným zdrojem informací, ohledně základních informací o 118. prvku pro nás bude tato encyklopedie dostatečná.<sup>19,20</sup> Prvek s protonovým číslem 118 již byl oficiálně objeven. Nicméně i tak se o něm mnoho neví, protože jeho objev byl potvrzen v roce 2015 na základě potvrzených a naměřených rozpadů tří (možná čtyř) atomů tohoto prvku z roku 2006. V roce 2016 pak bylo vybráno jméno pro tento prvek – oganesson – se značkou Og. Je pojmenován po ruském jaderném vědci Juriji Colakovičovi Oganessianovi. Vzhledem k tomu, že tento prvek patří do posledního sloupce tabulky, mělo by se jednat o vzácný plyn. Minimálně vzácný opravdu je. O tom, jestli by to byl plyn, se budeme bavit dále.

Naměřený izotop měl 294 nukleonů. Pokud bychom se zajímali o různé další odhady toho, jaké prvky by mohly být stabilní a jaká jejich nukleonová čísla by mohla být vhodná, pak bychom teoreticky mohli čekat, že by mohly být takto těžké prvky stabilnější, pokud by měly vyšší počet neutronů; existují ale modely, dle kterých je nejstabilnější izotop oganessonu skutečně ten s 294 nukleony. Tyto izotopy se zatím nějak nepodařilo naměřit, takže jde zatím pouze o nepotvrzenou teorii. Nicméně hned atomová hmotnost by mohla být zajímavou fyzikální vlastností k prozkoumání.

Ještě hned na úvod můžeme uvést, že když daný prvek patří do VIII.A skupiny, tak bychom očekávali, že bude mít zaplněnou valenční elektronovou slupku a jeho elektronová konfigurace by měla být  $5f^{14} 6d^{10} 7s^2 7p^6$ . Tím pádem by měl být obecně málo reaktivní, i když na druhou stranu by měl být pravděpodobně nejvíce reaktivní z celé skupiny. To plyne z toho, že čím dále jsou elektrony od jádra, tím slaběji jsou vázány, a tak mohou těžší prvky VIII.A skupiny snadněji vstupovat do reakcí. Taky bude nejsnazší jej ionizovat.

Snadno také můžeme tipovat, že bude radioaktivní, a to ve všech izotopech. Prvek s nejvyšším protonovým číslem, který má stabilní izotop, je olovo s protonovým číslem 82. Ještě bismut<sup>21</sup> je hodně stabilní se svým izotopem 209, ve kterém se vyskytuje v přírodě a jehož poločas rozpadu je delší než stávající stáří vesmíru.

Pro naše extrapolace budeme využívat program Wolfram Mathematica 11.0.1.0 a údaje o prvcích budeme získávat přímo z knihovny ElementData.<sup>22</sup> Tato knihov-

<sup>19</sup> *Wikipedie: Otevřená encyklopedie: Oganesson* [online]. c2017 [citováno 16. 10. 2017]. Dostupný z WWW: <<https://cs.wikipedia.org/w/index.php?title=Oganesson&oldid=14661315>>

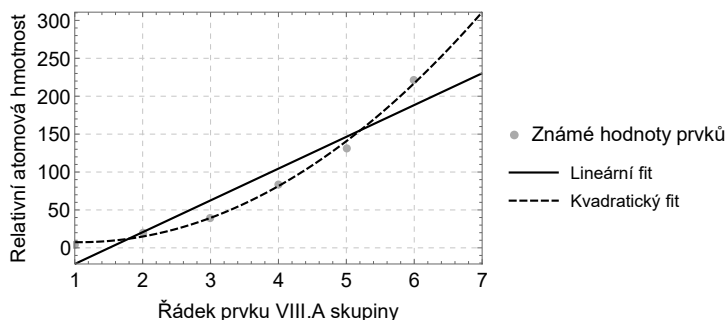
<sup>20</sup> *Wikipedia, The Free Encyclopedia: Oganesson* [online]. c2017 [citováno 16. 10. 2017]. Dostupný z WWW: <<https://en.wikipedia.org/w/index.php?title=Oganesson&oldid=805672099>>

<sup>21</sup> *Wikipedie: Otevřená encyklopedie: Bismut* [online]. c2017 [citováno 18. 10. 2017]. Dostupný z WWW: <<https://cs.wikipedia.org/w/index.php?title=Bismut&oldid=15391054>>

<sup>22</sup> ElementData Source Information: ElementData is based on a wide range of sources, with enhancement at the Wolfram Research Companies by both human and algorithmic processing. Among principal sources for ElementData are:

- Atomic Mass Data Center. *NUBASE*. 2003.
- Barbalace, K. *Periodic Table of Elements*. 2007.





Obr. 4: Extrapolace relativní atomové hmotnosti Og – dle periody

na obsahuje hodnoty známých veličin pro jednotlivé prvky a je tedy praktické, že přímo v programu, ve kterém probíhají extrapolace, můžeme snadno tyto údaje využít. Účastníci samozřejmě mohli využít jiné vhodně dostupné zdroje. Program Wolfram Mathematica je sice pro běžného středoškoláka drahý, ale je možné, alespoň částečně, využívat jeho funkčnosti v rámci webu WolframAlpha.<sup>23</sup> Bohužel je u něj omezení na výpočetní složitost, takže při jeho využití pravděpodobně musíte údaje o prvcích nalézt jinde a také vám asi nepůjde snadno naformátovat nějaký graf. Nebo lze využít zcela jiné programy či získat údaje o prvcích a ty až následně zpracovat například v Excelu nebo jiném tabulkovém procesoru.

Pro ověřitelnost a opakovatelnost výsledků přikládáme odkazy na soubory, ve kterých jsou použité příkazy,<sup>24</sup> a to jak ve formátu nb, tak cdf, který se dá případně alespoň zdarma zobrazit pomocí Wolfram CDF Player. Nicméně zde nejsou žádné interaktivní prvky (Manipulate), takže jde pak pouze o statické grafy.

Pro různé fyzikální vlastnosti bude různě účelné vycházet z hodnot buď na základě přímo protonového čísla nebo periody (čísla řádku), ve které daný prvek vyskytuje. Logicky pokud se budeme zajímat o atomovou hmotnost, tak bude spíš rozhodující protonové číslo. Pokud se ale zajímáme o nějaké chemické či fyzikální chemické vlastnosti, tak se stává potenciálně zajímavější číslo periody, ve které se daný prvek nalézá. My budeme provádět pouze jednoduché odhady vlastností prv-

- Cardarelli, F. *Materials Handbook: A Concise Desktop Reference*. Springer, 2000.
- Lide, D. R. (Ed.). *CRC Handbook of Chemistry and Physics*. 87<sup>th</sup> ed. CRC Press, 2006.
- Speight, J. *Lange's Handbook of Chemistry*. McGraw-Hill, 2004.
- United Kingdom National Physical Laboratory. *Kaye and Laby Tables of Physical and Chemical Constants*. 2007.
- United States National Institute of Standards and Technology. *Atomic Weights and Isotopic Compositions Elements*. 2005.
- United States National Institute of Standards and Technology. *NIST Chemistry Webbook*. 2005.
- Winter, M. *WebElements*. 2007.

<sup>23</sup><http://www.wolframalpha.com/>

<sup>24</sup>[https://fykos.cz/\\_media/rocnik31/ulohy/problem-2p.nb](https://fykos.cz/_media/rocnik31/ulohy/problem-2p.nb),  
[https://fykos.cz/\\_media/rocnik31/ulohy/problem-2p.cdf](https://fykos.cz/_media/rocnik31/ulohy/problem-2p.cdf)

ku 118 na základě známých fyzikálních vlastností jiných prvků, a to jednoduchou extrapolací, tedy proložením rozumnou funkcí. Nebudeme tedy uvažovat nějaké složitější fyzikální modely na vysokoškolské úrovni, spíše se budeme snažit prokládat hodnoty grafu nějakými rozumnými závislostmi a na základě toho tak trochu tipovat. I takto jednoduchý postup nám může dát často dostatečně dobré výsledky např. na to, aby mohlo být provedeno přesnější měření. Tedy v principu, pokud je daný prvek/sloučenina dost stabilní na to s ním něco naměřit. Musíme ale mít na paměti, že se může stát, že závislost, která do nějaké doby byla např. lineární, se pak prudce změní kvůli tomu, že se začne projevovat nějaký nový fyzikální jev.

Prokládat se budeme snažit co nejjednoduššími závislostmi, protože povětšinou máme velice málo dostupných bodů. Pokud se budeme zajímat pouze o prokládání v rámci vzácných plynů, pak budeme mít maximálně 6 hodnot předcházejících prvků (dle protonových čísel: 2 – helium, 10 – neon, 18 – argon, 36 – krypton, 54 – xenon, 86 – radon). Pro některé fyzikální vlastnosti, jako například pro teplotu tuhnutí, pak budeme mít těchto bodů i méně, protože například helium (alespoň za normálního tlaku) netuhne a radon není stabilní, ale podléhá radioaktivnímu rozpadu.

Naše metoda bude tím pádem opravdu relativně prostá. Budeme se prostě dívat na data, snažit se odhadnout, jestli nám na ně „dobře sedí“ lineární či kvadratická závislost, a to ještě jestli nám sedí líp na protonová čísla nebo index řádku prvku.

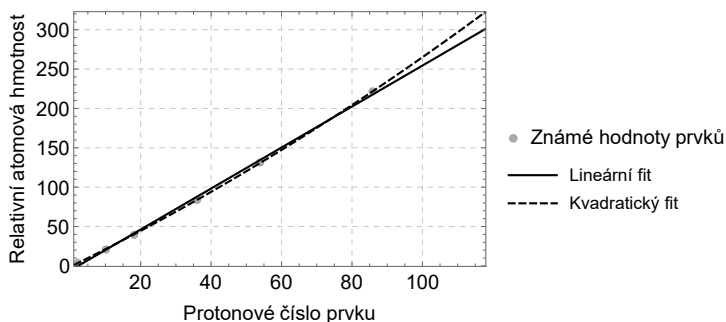
### *Relativní atomová hmotnost $O_g$*

Vraťme se k určování relativní atomové hmotnosti oganessonu a ukažme si, co jsme mysleli tou extrapolací. V grafu na obrázku 4 vidíme relativní atomové hmotnosti prvků VIII.A skupiny v závislosti na periodě, na obr. 5 pak v závislosti na jejich protonovém čísle a v tomto případě na obr. 6 vidíte i data v závislosti na protonovém čísle za všechny prvky. U této vlastnosti oganessonu je ovšem taková výjimka, že je již v datech ElementData odhadnutá, a to na 294. Tím se ovšem nenecháme připravit o tu zábavu s odhadováním a zkusíme si to ukázat právě na tomto příkladu. Navíc u jiných vlastností nám to už tato knihovna nepokazí, protože skoro žádné jiné vlastnosti  $O_g$  v ní nejsou definované.

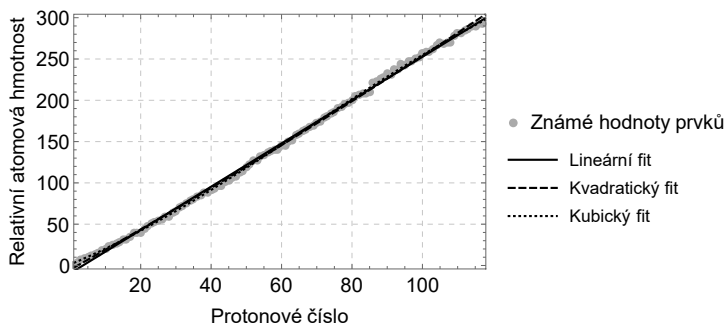
Pokud bychom se podívali na nějaké číslo, které nám může říci něco o tom, jak moc dobře data sedí na náš model, tak budeme využívat funkci AdjustedRSquared, kterou budeme značit dále  $R$ . Tato funkce nám zhruba říká to, jak moc daleko máme data od proložené křivky. Pokud by data ležela na přímce a my jsme je proložili přímkou, dostaneme  $R = 1$ . Pokud jsou data rozházená, dostáváme menší hodnoty a snažíme se přiblížit té 1, které ale v principu nemůžeme dosáhnout.<sup>25</sup> Nicméně ani toto číslo nám neřekne, jestli je odhad správný. Pokud bude ale hodně vzdálené od 1, tak nám to říká, že náš odhad nebude přesný.

Vidíme, že u obr. 4 data vůbec nesedí na přímku a na parabolu také moc nesedí. Je tam vidět to, že střídavě je hodnota pod a nad křivkou, což si dokážeme vysvětlit právě tím, že rozestupy mezi těmi prvky jsou nestejně (podívejte se na periodickou soustavu prvků). Pokud bychom se podívali na hodnoty kvadratického

<sup>25</sup>Kvůli nepřesnostem měření, kvůli tomu že nebudeme fitovat přesně tu funkci, podle které se daná fyzikální veličina řídí atd.



Obr. 5: Extrapolace relativní atomové hmotnosti Og – dle protonových čísel prvků VIII.A



Obr. 6: Extrapolace relativní atomové hmotnosti Og na základě všech prvků

fitu v řádku 7, odpovídalo by to relativní atomové hmotnosti 311 s  $R = 0,995$ . Vzhledem k tomu, že se nám tam střídají ty data nad a pod křivkou, tak si docela jistě ještě můžeme říci, že jsme tímto relativní atomovou hmotnost nadhodnotili.

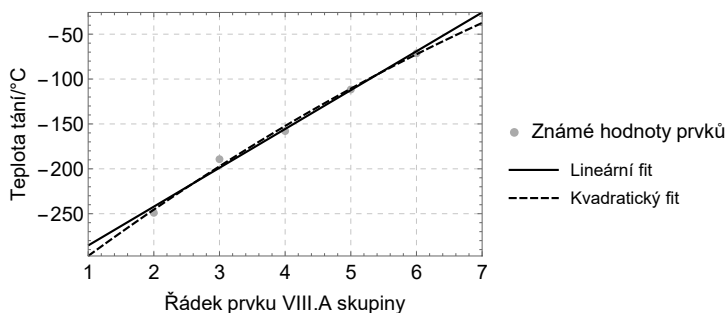
U grafu na obr. 5 už data sedí daleko lépe. Nicméně nám to naopak možná ukazuje, že i když formálně bude lépe sedět kvadratický fit, tak někdy nám ten lineární dá možná lepší extrapolovanou hodnotu. Důvodem je, že kvadratická funkce má o jeden volný parametr víc, máme tedy větší šanci se s ní trefit na naměřená data; pro libovolná data dokonce můžeme najít polynom vysokého řádu, který všemi body přesně prochází, ale mezi nimi se prudce mění. Lineární fit má  $R = 0,998$  a kvadratický  $R = 0,99994$ . Hodnota relativní atomové hmotnosti by podle lineární extrapolace byla 301, kdežto podle kvadratické 323.

Když se do třetice podíváme na graf na obr. 6, pak dostáváme hodnoty pro jednotlivé fity: lineární 300 ( $R = 0,998$ ), kvadratický 304 ( $R = 0,9989$ ) a kubický 298 ( $R = 0,9996$ ). V tomto případě si můžeme dovolit i ten kubický fit a bude dokonce pro naše využití nejlepší, protože máme hodně dat uvnitř našeho souboru

a extrapolujeme těsně za jeho hranici.

Z výše uvedených grafů a hodnot bychom tedy asi vybrali jako nejpravděpodobnější hodnotu relativní atomové hmotnosti Og jako 298. To se sice přesně neshoduje s 294, které jsou předpovězeny, ale jde o relativně blízký odhad. Hned si díky tomu musíme uvědomit, že naše další odhady nebudou úplně přesné. Je ovšem zajímavou informací, že se čeká, že jisté těžší prvky a možná i těžší izotopy oganessonů budou stabilnější, nicméně zatím se je nepodařilo připravit.<sup>26</sup>

### Teplota tání a tuhnutí Og

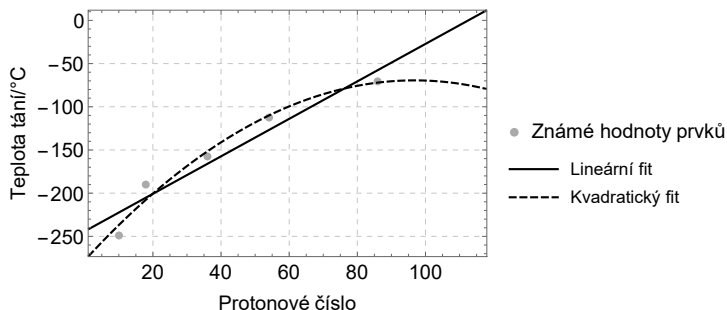


Obr. 7: Extrapolace teploty tání Og – dle periody prvku

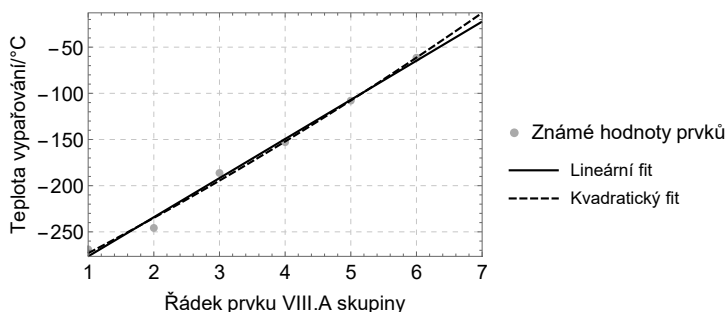
Dále se můžeme zabývat například teplotou tání. Jak jsme již zmínili, tak přijdeme o helium jako jeden z bodů, ale budeme se zabývat pouze pěti ostatními vzácnými plyny. Teplota tání už přeci jen trochu více souvisí s „chemií“ daného prvku, tedy s elektronovým obalem viz srovnání v grafech na obr. 7 a 8. Proto je také lépe vidět nějaká jednoduchá závislost z grafu, který je vytvořen v závislosti na periodě, ve které se prvek nachází (obr. 7). V grafu na obr. č. 8 naopak vidíme, že nám dává lineární a kvadratický odhad výrazně jiné výsledky, což tak trochu ukazuje, že odhad dle protonového čísla není úplně vhodný. Pokud graf v obr. 7 proložíme přímkou, dostáváme teplotu tání  $-26\text{ °C}$  ( $R = 0,99$ ), pro kvadratické proložení dostáváme  $-39\text{ °C}$  ( $R = 0,99$ ). Vzhledem k tomu, že se spolehlivost příliš neliší, bude pravděpodobně přesnější lineární odhad. Pokud bychom se podívali na nějaké další pravidelnosti, pak naměřená teplota tání je nejprve pod přímkou, pak nad, pak pod, zase mírně nad a nakonec opět pod. Z toho by se dalo usoudit, že pro Og možná bude opět mírně nad a na základě tohoto „věštění z křišťálové koule“ bychom mohli říci, že bude kolem  $-20\text{ °C}$ .

Podobnou vlastností jako teplota tání je teplota varu. Tak se hned podívejme, jak to bude s teplotou varu viz obr. 9 a 10. Helium nám alespoň kapalní, takže se nám do našich úvah opět vrací. Opět budeme brát jako důležitější periodu daného

<sup>26</sup> Wikipedie: Otevřená encyklopedie: Ostrov stability [online]. c2017 [citováno 18. 10. 2017]. Dostupný z WWW: <[https://cs.wikipedia.org/w/index.php?title=Ostrov\\_stability&oldid=14776718](https://cs.wikipedia.org/w/index.php?title=Ostrov_stability&oldid=14776718)>



Obr. 8: Extrapolace teploty tání Og – dle protonových čísel prvků

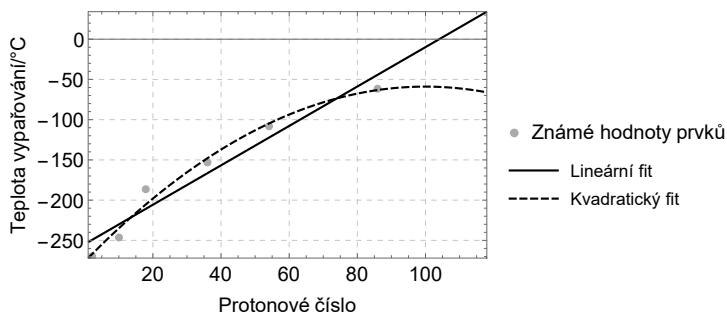


Obr. 9: Extrapolace teploty vypařování Og – dle periody

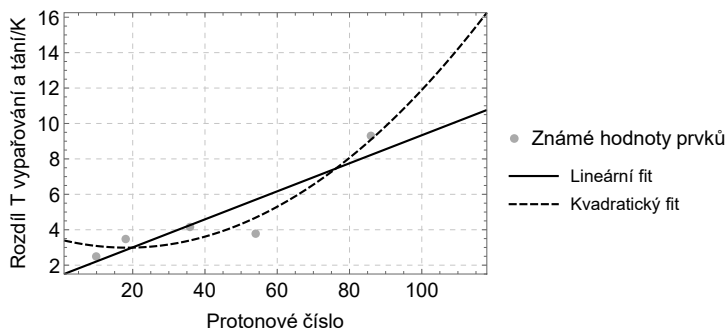
prvku. Pokud si to tedy proložíme přímkou dostáváme teplotu varu oganessonu  $-22\text{ }^{\circ}\text{C}$  ( $R = 0,99$ ) s lineárním proložením, respektive  $-14\text{ }^{\circ}\text{C}$  ( $R = 0,99$ ) s kvadratickým proložením. Tentokrát bude jistě rozumnější vzít lineární proložení, a to prohlásit za náš odhad.

Co kdybychom se ale chtěli zajímat o nějakou „složenu“ dvou fyzikálních vlastností? Hned první, která by nás mohla napadnout, by mohl být rozdíl mezi teplotou tání a teplotou varu. To je zajímavá vlastnost, protože například široké rozpětí kapalné vody umožňuje život na Zemi. Vlastnost se ale pro nás ukáže až tak zajímavou, že z toho nedokážeme nic odhadnout. Proč? Protože pro neon je to 2,5 K, pro argon 3,5 K a krypton 4,1 K. To by bylo ještě dobré, ale pak nám tento rozdíl klesne na 3,8 K u xenonu a zase docela dramaticky vzroste na 9,3 K u radonu. Takže z těchto málo bodů se dá říci, že se závislost nedá nijak jednoduše odhadnout (maximálně bychom mohli čekat, že půjde řádově o jednotky kelvinů) a ukazuje to, že nějaké složitější vlastnosti asi půjde odhadnout jenom velice špatně. Proložené křivky, jejichž výsledné hodnoty ani neuvádíme, můžete vidět jako ukázkou v grafu na obr. č. 11.

Co srovnání se současnými teoriemi? Vypadá to, že se právě u tohoto prvku



Obr. 10: Extrapolace teploty vypařování Og – dle protonových čísel prvků



Obr. 11: Extrapolace rozdílu teploty vypařování a tání Og – dle protonových čísel prvků

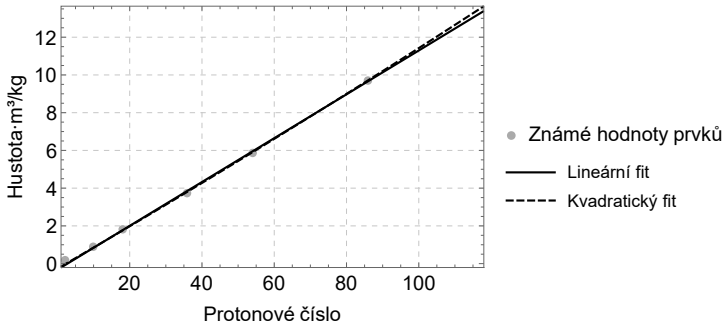
už má nejspíše začít projevovat „nová fyzika“, protože podle jednoho článku<sup>27</sup> by teplota varu měla být kolem 80 °C s tím, že při pokojové teplotě by tento prvek měl být pevný. Nicméně jsou to také zatím pouze neověřené teorie.

### Hustota

Sice je možné, že oganesson by byl pevná látka, nicméně my zkusíme odhadnout, jakou by měl hustotu, byl-li by to plyn za standardních podmínek. Hustotu bychom mohli odhadnout také z již odhadnuté relativní atomové hmotnosti a z rovnice pro ideální plyn, ale my to opět zkusíme na základě extrapolace známých hodnot. Tentokrát rovnou vybereme variantu, kde prokládáme hodnoty hustot plynů, kde na ose  $x$  máme protonové číslo prvku. Hustota jednoatomového plynu za standardních podmínek bude opravdu spíše odpovídat hmotnosti jednotlivých atomů než řádku periodické tabulky. Graf vidíme na obrázku 12. Vidíme, že na data velice

<sup>27</sup>Nash, Clinton S. (2005). *Atomic and Molecular Properties of Elements 112, 114, and 118*. Journal of Physical Chemistry A. 109 (15): 3493–3500. doi:10.1021/jp050736o. PMID 16833687.

dobře sedí už samotná přímka. Pro lineární fit dostáváme odhad  $\rho = 13,4 \text{ kg}\cdot\text{m}^{-3}$  a pro kvadratický fit je to  $13,6 \text{ kg}\cdot\text{m}^{-3}$ . U obou fitů dostáváme velice vysoké  $R$  blížící se 1. Vzhledem k tomu, že některé hustoty vzácných plynů známe jenom na dvě platné cifry, tak bychom měli prohlásit, že hustotu Og bychom čekali mezi  $13 \text{ kg}\cdot\text{m}^{-3}$  a  $14 \text{ kg}\cdot\text{m}^{-3}$ . To by platilo, pokud by Og netvořil víceatomové molekuly a byl jednoatomovým plynem. Pokud by tvořil dvouatomové molekuly (což asi nemůžeme úplně vyloučit, i když je ve sloupci vzácných plynů), pak by jeho hustota byla zhruba dvojnásobná. Pokud by tedy Og byl stabilní a byl to plyn, tak by za standardních podmínek byl nejhustším plynným prvkem a jednou z nejhustších plynných látek vůbec.



Obr. 12: Extrapolace hustoty Og jako plynné látky za standardních podmínek

### Úloha III.1 ... zpomalená

Představme si, že na kameru se snímkovou frekvencí 24 snímků za sekundu (uvažujeme časově rovnoměrně rozložené a dokonale ostré snímky) natočíme let vrtulníku s otáčkami hlavního rotoru 2900 ot./min. Následně si záznam přehrajeme. Jaká bude zdánlivá frekvence otáček rotoru na záznamu?

Frekvencia otáčania rotoru je

$$f_r = 2900 \text{ min}^{-1} = \frac{145}{3} \text{ s}^{-1}.$$

Za čas, ktorý uplynie medzi dvoma po sebe vytvorenými snímkami, sa rotor otočí  $N$ krát

$$N = \frac{f_r}{f_k} = \frac{145}{72},$$

kde  $f_k$  je snímková frekvencia kamery  $24 \text{ s}^{-1}$ . Na snímkach nie je vidieť celé otáčky, iba posun voči predošlej polohe, preto sa zdá, že rotor medzi jednotlivými snímkami vykoná otočenie len o  $\Delta N$ . To získame ako

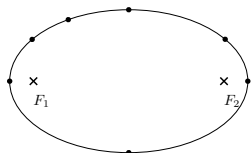
$$\Delta N = N - k,$$

kde  $k$  je také celé číslo, aby  $|\Delta N|$  bola čo najmenšia (teda  $k$  je celá časť  $N$ ). Potom zdanlivá frekvencia otáčania rotora je

$$f' = \Delta N f_k = \frac{1}{72} f_k = \frac{1}{3} \text{ s}^{-1}.$$

Zdanlivá frekvencia otáčania rotora na snímkoch je  $1/3 \text{ s}^{-1} = 20 \text{ ot./min.}$

### Úloha III.2 ... zrychleníčko, zrychlení



Na obrázku vidíte náčrt elipsy s ohnisky  $F_1$  a  $F_2$  a niekoľko vyznačenými body na ní. Uvažujte, že elipsa znázorňuje trajektoriu nějakého hmotného bodu. Znázorněte do obrázku zrychlení, ktorá pôsobí na hmotný bod v jednotlivých vyznačených bodech dráhy pro dvě situace (jde o směry a vzájemné poměry zrychlení (které je

větší/menší) v různých bodech v rámci jednoho náčrtu).

- V ohnisku  $F_1$  je umístěno hmotné těleso, kolem kterého hmotný bod obíhá. Uvažujeme, že platí 2. Keplerův zákon.
- Těleso má konstantní velikost rychlosti, pouze se pohybuje po elipse.

a)

Ako prvé si musíme uvedomiť, čo za druh zrýchlenia pôsobí v prípade klasickej Keplerovej úlohy Slnko-jedna planéta. Vezmeme si na počiatku teleso v nejakej vzdialenosti od Slnka a udelíme mu rýchlosť, menšiu ako je kruhová rýchlosť v tejto vzdialenosti smerom kolmo od spojnice teleso-Slnko. Zo základov nebeskej mechaniky vieme, že teleso sa bude pohybovať po elipse, ktorej afélium (najvzdialenejší bod dráhy od Slnka) bude práve v bode, odkiaľ sme teleso vypustili. Jediná sila, ktorá bude v takomto prípade pôsobiť na nami vypustené teleso, bude príťažlivá gravitačná sila medzi telesom a Slnkom. Pre ľubovoľnú silu pôsobiacu na teleso s konštantnou hmotnosťou platí, že táto sila uvádza teleso do pohybu so zrýchlením, ktoré je rovné

$$a = \frac{F}{m},$$

kde  $F$  je sila pôsobiaca na teleso a  $m$  je hmotnosť daného telesa. Môžeme si všimnúť, že z tohto vzťahu si vieme prirodzene definovať hmotnosť ako schopnosť telesa klásť odpor voči pôsobeniu vonkajších síl. Sila pôsobiaca na toto teleso je v našom prípade rovná Newtonovskej gravitačnej sile

$$F_g = -\frac{GMm}{r^3} \mathbf{r},$$

kde  $G$  je gravitačná konštanta,  $M$  je hmotnosť centrálneho telesa (v našom prípade Slnka) a  $\mathbf{r}$  je polohový vektor voči Slnku. Ďalej je vhodné spraviť si transformáciu kartézskych súradníc s počiatkom v strede elipsy, na kartézske súradnice so stredom v Slnku. Potom  $y$ ová súradnica Slnka je totožná s  $y$  súradnicou stredu elipsy, avšak  $z$ ová súradnicu musíme posunúť o

$$f = \sqrt{a^2 - b^2},$$



kde  $a$ ,  $b$  sú hlavné a vedľajšie polosy elipsy. Potom nové súradnice budú:

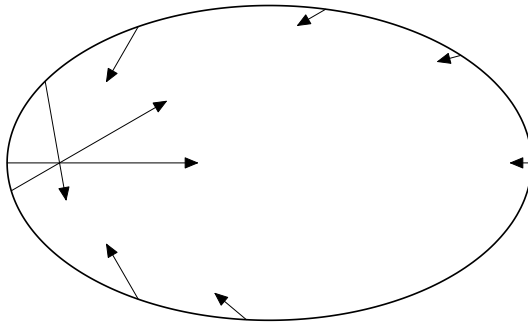
$$\begin{aligned}x' &= x + f, \\y' &= y.\end{aligned}$$

Ak do vzorca pre zrýchlenie dosadíme náš vzorec pre silu a doň dosadíme za  $\mathbf{r}$  polohu bodu v nových súradniciach, tak výsledný vzťah pre veľkosť zrýchlenia pre ľubovoľný bod ležiaci na elipse bude

$$a_d = \frac{GM}{x'^2 + y'^2} = \frac{GM}{(x + \sqrt{a^2 - b^2})^2 + y^2}.$$

Po dosadení do tohto vzorca vieme určiť presnú veľkosť. Avšak našim cieľom je toto zrýchlenie zakresliť. Keďže smer zrýchlenia je od daného bodu na elipse smerom k Slnku, v našom prípade smeruje vektor zrýchlenia z daného bodu na elipse, do ohniska  $F_1$ .

Ak si vhodne zvolíme mierku obrázku tak s týmito informáciami je jednoduché vkresliť vektory zrýchlenia do obrázku. Vyzeráť by to mohlo asi ako na Obrázku 13.



Obr. 13: Výsledok prípadu a).

b)

Predpokladajme teraz že teleso sa znova hýbe po tej istej elipse, a to napriek prítomnosti akejkoľvek gravitačnej alebo inej sily. Môžeme si predstaviť napríklad vesmírnu loď ktorá niekde v priestore jazdí dookola po elipse, spaľujúc pritom palivo na to, aby si udržala požadovanú dráhu. Vieme že zrýchlenie pôsobiace na teleso si vieme vždy rozdeliť na normálovú zložku, ktorá spôsobuje zakrivenie dráhy, a na tangenciálnu zložku, ktorá spôsobuje zmenu rýchlosti telesa. Keďže teleso si po celý čas udržiava rovnakú rýchlosť, je tangenciálna zložka zrýchlenia nulová, a smer zrýchlenia je teda totožný s normálou k dotyčnici v danom bode.

Úlohu si rozdelíme na dve časti. Najprv budeme hľadať, ako vyzerá smerový vektor zrýchlenia, t.j. normála k danej krivke. Potom si zrátame, akú veľkosť má zrýchlenie v závislosti na poloha. Keďže sa jedná o geometrickú úlohu, nemusíme

nájsť explicitný predpis pre priamku, na ktorej leží normálový vektor v danom bode. Stačí nám vedieť, že daná priamka bude prechádzať cez nami zvolený bod, a bude rovnobežná s vektorom gradientu elipsy (ako implicitnej funkcie), ktorý zo svojej prirodzenej definície je vnímaný ako vektorový operátor určujúci normálu k nejakej krivke či ploche. Bez znalostí vyššej matematiky, skrátka vezmeme pravítko s ryskou a nájdeme kolmicu na dotyčnicu v nejakom bode. Poznámka pre prípadných záujemcov, matematicky-analyticky by sme to spočítali takto:

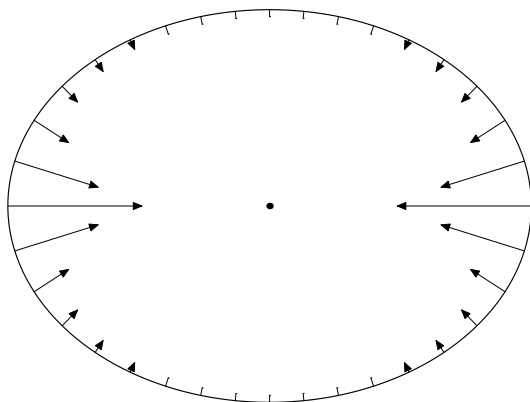
Elipsa je popísaná implicitne krivkou

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Gradientom tejto krivky, a teda nami hľadaným vektorom je

$$\left[ \frac{2x}{a^2}, \frac{2y}{b^2} \right].$$

Na základe týchto informácií vieme ľubovoľnému bodu priradiť smer zrýchlenia.



Obr. 14: Výsledok prípadu b).

Veľkosť dostredivého zrýchlenia ("šípka" vektora bude teda smerovať vždy smerom dovnútra elipsy) je potom daná vzťahom

$$a_d = \frac{v^2}{R},$$

kde rýchlosť  $v$  je po celý čas konštantná a polomer krivosti elipsy  $R$  v nejakom bode, je daný vzťahom

$$R = a^2 b^2 \left( \frac{x^2}{a^4} + \frac{y^2}{b^4} \right)^{\frac{3}{2}},$$

kde  $x$  a  $y$  sú súradnice daného bodu. Dosadením vzťahu pre polomer krivosti v nejakom bode do vzťahu pre veľkosť dostredivého zrýchlenia máme vzorec, z ktorého vieme dopočítať veľkosť zrýchlenia v ľubovoľnom bode na elipse, pri daných počiatočných podmienkach. Po zakreslení do obrázku s použitím takto odvodených informácií by mal vyzerať ako Obrázek 14.

### Úloha III.3 ... IDKFA

Vypálili jste na impa z plazmové pušky, která střílí stabilní shluk částic s rovnoměrným rozdělením podélné rychlosti v intervalu  $\langle v_0, v_0 + \delta v \rangle$  (příčná rychlost je nulová) a s celkovou energií  $E_0$ . Hlaveň pušky má průřez  $S$  a pulz trvá nekonečně krátký čas. Jak daleko musí imp stát, aby se mu nic nestalo? Předpokládejte, že jeho kůže bez problémů uchládí na malém prostoru tepelný tok  $q$ .

Kinetická energie částic je přímo úměrná druhé mocnině jejich rychlosti. Rozdělme tedy celé spektrum rychlostí na úseky o délce  $\Delta v$ . Protože délku jednotlivých úseků můžeme zvolit libovolně malou, můžeme předpokládat, že všechny částice z daného úseku mají stejnou rychlost  $v$ . Kinetická energie daného úseku potom bude  $\Delta E_k = kv^2 \Delta v$  ( $\Delta v$  tady vyjadřuje počet částic s rychlostí  $v$  v daném úseku), kde  $k$  je nějaká konstanta, ktorou dále spočítáme.

Pokud sečteme  $\Delta E_k$  ze všech úseků, dostaneme energii celého svazku částic, neboli  $E_0$ . Pokud umíme integrovat, nahradíme konečné rozdíly  $\Delta$  diferenciály  $d$  a čeká nás jednoduchý výpočet

$$E_0 = \int_{v_0}^{v_0+\delta v} dE_k = \int_{v_0}^{v_0+\delta v} kv^2 dv = \frac{k}{3} ((v_0 + \delta v)^3 - v_0^3),$$

$$k = \frac{3E_0}{(v_0 + \delta v)^3 - v_0^3}. \quad (9)$$

Pokud zatím integrovat neumíme, můžeme se pokusit integrál nahradit sumou. Označíme-li počet všech úseků

$$n = \frac{\delta v}{\Delta v},$$

přičemž rychlost  $v$   $j$ -tém úseku (čísľujeme od 1) bude  $v_j = v_0 + j\Delta v$ , dostávame rovnici

$$E_0 = \sum_{j=1}^n \Delta E_{k,j} = \sum_{j=1}^n k (v_0 + j\Delta v)^2 \Delta v = k\Delta v \sum_{j=1}^n (v_0^2 + 2jv_0\Delta v + j^2\Delta v^2) =$$

$$= k\Delta v \left( nv_0^2 + 2v_0\Delta v \frac{n(n+1)}{2} + \Delta v^2 \frac{n(n+1)(2n+1)}{6} \right),$$

kde po dosazení za  $n$  a za předpokladu libovolně malého  $\Delta v$  (kdy můžeme využít odhad  $n+1 \approx n$ ) dostávame

$$E_0 = k \left( v_0^2 \Delta v + v_0 \Delta v^2 + \frac{1}{3} \Delta v^3 \right) = \frac{k}{3} ((v_0 + \Delta v)^3 - v_0^3),$$

což je to samé, co jsme odvodili výše.

Nyní uvažujme, že imp stojí ve vzdálenosti  $x$  od pušky. V čase  $t$  na něj za nějaký malý časový úsek  $\Delta t$  dopadnou částice z rychlostního intervalu  $\langle v, v + \Delta v \rangle$ , pro které platí

$$\begin{aligned} v &= \frac{x}{t}, \\ v + \Delta v &= \frac{x}{t - \Delta t}, \\ \Delta v &= \frac{x}{t - \Delta t} - \frac{x}{t} = x \frac{\Delta t}{t(t - \Delta t)}. \end{aligned}$$

Za předpokladu, že  $\Delta t$  zvolíme dostatečně malé, můžeme  $\Delta$  nahradit diferenciálem a psát

$$dv = x \frac{dt}{t^2} = \frac{v^2}{x} dt.$$

Kinetickou energii těchto částic můžeme spočítat jako  $dE_k = kv^2 dv$ . Pro výkon svazku potom platí

$$P = \frac{dE_k}{dt} = kv^2 \frac{dv}{dt} = \frac{kv^4}{x}.$$

Z tohoto vzorce vyplývá, že největší výkonu bude dosaženo pro největší rychlost, což je  $v_0 + \delta v$ .

Můžeme předpokládat, že částice dopadají rovnoměrně na část povrchu impa o obsahu  $S$ . Maximální výkon, který imp zvládne absorbovat, tak bude  $P = qS$ . Dostáváme rovnici

$$qS = \frac{k(v_0 + \delta v)^4}{x},$$

ze které po vyjádření  $x$  a dosazení za  $k$  z (9) vyplývá

$$x = \frac{3E_0}{qS} \frac{(v_0 + \delta v)^4}{(v_0 + \delta v)^3 - v_0^3}.$$

Spočítali jsme, v jaké nejmenší vzdálenosti od zbraně může imp stát, aby mu plazmový výstřel nic neudělal. Kdo by čekal, že fyzika DOOMa bude *takhle* zajímavá?

### Úloha III.4 ... upuštěná propiska

*Propisku (tuhou tyč) upustíme na stůl tak, že během svého letu svírá úhel  $\alpha$  s vodorovnou rovinou. Jakou rychlostí dopadne její druhý konec (ten, co se stolu dotkne jako druhý), jestliže jsme těžiště upustili z výšky  $h$ ? Všechny srážky jsou nepružné a tření mezi stolem a koncem propisky dostatečně velké.*

*Bonus* Spočítejte, jaký musíme zvolit úhel  $\alpha$ , aby druhý konec dopadl s co nejvyšší rychlostí. Pro jaké výšky se vyplatí propisku naklonit?

Propiska nejdříve padá volným pádem (bez jakékoliv rotace). Potom její dolní konec dopadne na zem, přičemž zůstane zachován její moment hybnosti kolem bodu dopadu. Jelikož uvažujeme nepružnou srážku, tak se propiska neodrazí. Pohyb propisky se tak změní z posuvného na otáčivý. Protože uvažujeme velké tření, spodní konec neproklouzne, bude se dotýkat pořád stejné části stolu a propiska se tak bude otáčet kolem tohoto bodu. Při tomto otáčení jí bude pořád urychlovat tíhové zrychlení. Řešení si rozdělíme na tyto tři části.

Pro výpočet potřebujeme znát i délku propisky. Tu si označíme  $l$ .

### Volný pád

Původní výška spodního konce je

$$h - \frac{l}{2} \sin \alpha.$$

Ta bude rovna  $gt^2/2$ , kde  $t$  je čas pádu. Nás ale zajímá rychlost, kterou tužka během pádu nabere,  $v = gt$ . Dosazením dostáváme výsledek známý i ze zákona zachování energie

$$h - \frac{l}{2} \sin \alpha = \frac{v^2}{2g},$$

$$v = \sqrt{2g \left( h - \frac{l}{2} \sin \alpha \right)}.$$

### Srážka

Nyní hledáme úhlovou rychlost, kterou se tužka začne otáčet kolem bodu dotyku po dopadu na stůl. Srážka je nepružná, takže nemůžeme použít zákon zachování energie (nevíme, kolik se přeměnilo na teplo, vibrace). Zároveň také nemůžeme předpokládat, že si druhý konec zachová svou rychlost. Musíme použít zákon zachování momentu hybnosti.

Na tužku působí dvě různé vnější síly. Jedna je v bodě nárazu. Tímto bodem, ale prochází osa otáčení tužky. Rameno této síly je tedy nulové, takže na tužku působí nulovým momentem síly.<sup>28</sup> Druhá síla je tíhová. Srážka nastane ale v jednom okamžiku, takže tato síla nestihne nějak změnit hybnost nebo moment hybnosti tužky. To započítáme až později, během otáčivého pádu. Z toho vyplývá, že její moment hybnosti  $L$  je během srážky konstantní.

Vycházíme tedy z toho, že moment hybnosti těsně před nárazem je stejný jako po nárazu.

Moment hybnosti hmotného bodu spočítáme jako

$$L = pr = mvr,$$

kde  $m$  je hmotnost bodu,  $v$  je jeho rychlost a  $r$  je vzdálenost od osy otáčení (tedy od bodu dotyku se stolem) měřena kolmo na směr rychlosti. Pokud je těleso tuhé

<sup>28</sup>Tato úvaha není dokonalá, protože nefunguje pro „nekonečné“ síly. Ty se v dostatečně zjednodušeném modelu s nulovým časem srážky a nárazem v jednom bodě skutečně vyskytují. To, že kontakt s podložkou neovlivňuje moment hybnosti, je ale rozumný předpoklad.

a nerotuje (tzn. všechny jeho body mají stejnou velikost i směr rychlosti), tak si ho pro výpočet momentu hybnosti můžeme nahradit hmotným bodem v jeho těžišti. Rychlost tužky má pouze svislou složku, kolmá vzdálenost od osy otáčení je tedy vodorovná vzdálenost těžiště od osy  $r = \frac{l}{2} \cos \alpha$ . Počáteční moment hybnosti je

$$L = mv \frac{l}{2} \cos \alpha.$$

Pro otáčivý pohyb tužky těsně po srážce platí  $L = J\omega$ , kde  $\omega$  je úhlová rychlost otáčení kolem dané osy a  $J$  je moment setrvačnosti tělesa vzhledem k této ose. Pro tyč, otáčející se kolem osy na ni kolmé a procházející jejím koncem, platí vzoreček  $J = ml^2/3$ , proto

$$\begin{aligned} \frac{1}{2} mvl \cos \alpha &= \frac{1}{3} ml^2 \omega_0, \\ \omega_0 &= \frac{3v}{2l} \cos \alpha, \end{aligned}$$

kde  $\omega_0$  je okamžitá úhlová rychlost hned po srážce.

### Otáčivý dopad

Tužka ale dopadne s vyšší úhlovou rychlostí než jakou měla hned po srážce. Tuto rychlost už můžeme vypočítat ze zákona zachování energie. Během tohoto otáčivého pádu se poloha jejího těžiště sníží o  $\Delta h = \frac{l}{2} \sin \alpha$ . Kinetická energie se tedy zvýší o

$$\Delta E = \frac{1}{2} lmg \sin \alpha.$$

Podle vzorce pro kinetickou energii otáčivého pohybu dostáváme

$$\Delta E = \frac{1}{2} J\omega_1^2 - \frac{1}{2} J\omega_0^2 = \frac{1}{6} ml^2 (\omega_1^2 - \omega_0^2),$$

kde  $\omega_1$  je úhlová rychlost při dopadu druhého konce na stůl. Ze zákona zachování energie

$$\begin{aligned} \frac{1}{6} ml^2 (\omega_1^2 - \omega_0^2) &= \Delta E = \frac{1}{2} l \sin \alpha mg, \\ \omega_1 &= \sqrt{\frac{3g \sin \alpha}{l} + \omega_0^2}. \end{aligned}$$

Dosadíme předchozí výsledky

$$\begin{aligned} \omega_1 &= \sqrt{\frac{3g \sin \alpha}{l} + \frac{9v^2}{4l^2} \cos^2 \alpha}, \\ &= \sqrt{\frac{3g \sin \alpha}{l} + \frac{9g}{2l^2} \cos^2 \alpha \left( h - \frac{l}{2} \sin \alpha \right)} \end{aligned}$$

a vyjádříme rychlost  $v_1$ , se kterou dopadá druhý vrchol na stůl

$$v_1 = l\omega_1 = \sqrt{3gl \sin \alpha + \frac{9g}{2} \cos^2 \alpha \left( h - \frac{l}{2} \sin \alpha \right)}.$$

**Bonus**

Hledáme úhel  $\alpha$  tak, abychom maximalizovali rychlost dopadu  $v_1$ . Rychlost  $v_1$  je maximální, když je výraz pod odmocninou maximální. To musí nastat buď pro některý hraniční úhel ( $\alpha = 0$  je nejmenší možný, pro  $h \geq l/2$  je největší možný  $90^\circ$  a pro  $h < l/2$  je to  $\arcsin(2h/l)$ ) nebo nulovou derivaci

$$\begin{aligned} \frac{dv_1}{d\alpha} &= 0, \\ 3l \cos \alpha - 9h \cos \alpha \sin \alpha + \frac{9l}{2} \cos \alpha \sin^2 \alpha - \frac{9l}{4} \cos^3 \alpha &= 0, \end{aligned}$$

Jedním řešením je  $\cos \alpha = 0$ , tedy  $\alpha_1 = 90^\circ$ .

$$\begin{aligned} 1 - \frac{3h}{l} \sin \alpha + \frac{3}{2} \sin^2 \alpha - \frac{3}{4} + \frac{3}{4} \sin^2 \alpha &= 0, \\ 1 - \frac{12h}{l} \sin \alpha + 9 \sin^2 \alpha &= 0, \\ \sin \alpha &= \frac{2h}{3l} \pm \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1}. \end{aligned}$$

Vidíme, že řešení kvadratické rovnice existuje, jen pokud  $h \geq \frac{l}{2}$ . To dává smysl, ale otázkou zůstává, co je pro nás výhodnější pro  $h < l/2$ , nechat propisku padat horizontálně ( $\alpha = 0$ ) nebo ji položit jedním koncem na zem ( $\alpha = \arcsin(2h/l)$ )?<sup>29</sup> Pro volný horizontální pád máme  $v_a = \sqrt{9gh/2}$  a pro otáčivý pád  $v_b = \sqrt{3gl \sin \alpha}$ , kde  $2h = l \sin \alpha$ , tedy  $v_b = \sqrt{6gh}$ . Pro  $h < l/2$  dopadne druhý konec nejvyšší rychlostí, když zvolíme maximální úhel  $\alpha$ , tedy tužku opřeme jedním koncem o stůl.

Pro  $h \geq l/2$  může existovat třetí nebo čtvrtý extrém, pokud

$$1 \geq \frac{2h}{3l} \pm \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1} \geq 0.$$

Nyní, pokud bude před odmocninou mínus, tedy

$$1 \geq \frac{2h}{3l} - \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1} \geq 0,$$

lze snadno ověřit, že výraz bude kladný, protože

$$\frac{2h}{3l} \geq \frac{1}{3} \sqrt{\left(\frac{2h}{l}\right)^2 - 1}$$

a zároveň hodnota výrazu nepřekročí 1, protože funkce je klesající a její počáteční hodnota v bodě  $h = \frac{l}{2}$  je  $\frac{1}{3}$ .

<sup>29</sup>Řešení  $\cos \alpha = 0$  sice matematicky smysl dává, ale pro  $h < l/2$  to fyzikální smysl nedává, protože by na začátku byl spodní konec tužky pod úrovní stolu.

V případě, že před odmocninou je plus, je druhá nerovnost splněna vždy, ale první nemusí být. Úpravami dostaneme

$$\begin{aligned} 1 - \frac{2h}{3l} &\geq \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1}, \\ 1 - \frac{4h}{3l} + \frac{4h^2}{9l^2} &\geq \frac{4h^2}{9l^2} - \frac{1}{9}, \\ h &\leq \frac{5l}{6}. \end{aligned}$$

Čtvrtý extrém je možný pouze pokud  $h \leq \frac{5}{6}l$ . Pokud ale rozepíšeme výraz

$$\begin{aligned} &3gl \left( \frac{2h}{3l} \pm \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1} \right) + \\ &+ \frac{9g}{2} \left( 1 - \left( \frac{2h}{3l} \pm \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1} \right)^2 \right) \left( h - \frac{l}{2} \left( \frac{2h}{3l} \pm \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1} \right) \right) \end{aligned}$$

a hledáme, kdy je větší pro znaménko plus, téměř vše se vykrátí a dopracujeme se k nerovnosti

$$4 \frac{h^2}{l^2} - 1 < 0.$$

To neplatí nikdy (předpokládáme  $2h \geq l$ ), proto znaménko plus nikdy nevede na globální maximum.

Lze ověřit, že derivace  $\frac{dv_1}{d\alpha}$  v bodě  $\alpha = 0$  bude kladná, tedy funkce bude rostoucí a v tomto bodě taky nebude extrém.

Zbývá vyšetřit, jestli se vyplatí tužku naklonit pod úhlem  $\alpha$ , pro který platí  $\sin \alpha = \frac{2h}{3l} - \frac{1}{3} \sqrt{4 \frac{h^2}{l^2} - 1}$ , nebo  $\alpha = 90^\circ$ . Jelikož bychom se při porovnávání příslušných rychlostí dopadu dostali k rovnicím vysokých stupňů, můžeme využít Wolfram Alpha. Zjistíme, že pro  $h < (4/\sqrt{3} - 1)l/2$  je maximální rychlost  $v_1 = \sqrt{3gl}$  s  $\alpha = 90^\circ$  a pro větší výšky je to

$$v_1 = \sqrt{3gl \sin \alpha + \frac{9g}{2} \cos^2 \alpha \left( h - \frac{l}{2} \sin \alpha \right)}$$

s nakloněnou tužkou.

To dává celkem dobrý smysl – pro  $\alpha = 90^\circ$  nezávisí rychlost dopadu na výšce  $h$ , takže pro velké výšky nebude maximální, zatímco pro malé výšky ( $h \approx l/2$ ) se neoplatí nechat tužku nejdřív volně padat.

### Úloha III.5 . . . rozpad sem, rozpad tam

Máme  $A_0$  částic typu  $A$ , které se s rozpadovou konstantou  $\lambda_A$  rozpadají na částice typu  $B$ . Ty se zase s rozpadovou konstantou  $\lambda_B$  rozpadají na částice typu  $A$  a na začátku jich je  $B_0$ . Najděte funkci udávající poměr počtů částic typů  $A$  a  $B$  v čase.



Označme  $dA_r$  počet částic typu  $A$ , které se rozpadnou za čas  $dt$ . Derivací rovnice radioaktivního (nebo chemického) rozpadu dostaneme

$$\frac{dN}{dt} = \frac{d(N_0 e^{-\lambda t})}{dt} = -\lambda N_0 e^{-\lambda t} = -\lambda N,$$

což můžeme dále upravit na

$$dN = -\lambda N dt.$$

Z toho vyplývá  $dA_r = \lambda_A A dt$ . Analogicky bude platit  $dB_r = \lambda_B B dt$ . Nyní je potřeba si uvědomit, že celkovou změnu počtu částic typu  $A$  za čas  $dt$  (kterou si označíme  $dA$ ) spočítáme jako rozdíl počtu částic, které se z  $B$  rozpadnou na  $A$ , a částic, které se z  $A$  rozpadnou na  $B$ . Jinými slovy

$$dA = dB_r - dA_r = (\lambda_B B - \lambda_A A) dt = -dB. \quad (10)$$

Pro další řešení úlohy musíme zredukovat počet proměnných. Můžeme to udělat dvěma způsoby – buď si jednu proměnnou vyjádříme pomocí druhé, nebo budeme počítat s jejich podílem. První je mírně pracnější, druhý naopak vyžaduje řešení náročnějšího integrálu. Pro názornost si ukážeme oba postupy.

Označme  $M = A_0 + B_0$ . Protože se celkový počet částic zachovává (to vidíme z  $dA + dB = 0$ ), platí  $M = A + B$ . Díky tomu si můžeme  $A$  vyjádřit jako  $A = M - B$ . Po dosazení do rovnice (10) dostáváme

$$\frac{dB}{dt} = \lambda_A A - \lambda_B B = \lambda_A (M - B) - \lambda_B B = \lambda_A M - (\lambda_A + \lambda_B) B.$$

Tuto diferenciální rovnici můžeme řešit separací proměnných

$$\int dt = \int \frac{dB}{\lambda_A M - (\lambda_A + \lambda_B) B}.$$

Integrál levé strany je zřejmý. Pravá strana je ve tvaru derivace nějaké funkce od  $B$ , dělené tou samou funkcí od  $B$ . Integrálem výrazů tohoto typu je vždy logaritmus dané funkce:

$$t = -\frac{1}{\lambda_A + \lambda_B} \int \frac{-(\lambda_A + \lambda_B) dB}{\lambda_A M - (\lambda_A + \lambda_B) B} = -\frac{1}{\lambda_A + \lambda_B} \ln(\lambda_A M - (\lambda_A + \lambda_B) B) + C.$$

Z počátečních podmínek víme, že v čase  $t = 0$  máme  $B = B_0$ . To vede na vztah pro výpočet integrační konstanty

$$C = \frac{1}{\lambda_A + \lambda_B} \ln(\lambda_A M - (\lambda_A + \lambda_B) B_0) = \frac{1}{\lambda_A + \lambda_B} \ln(\lambda_A A_0 - \lambda_B B_0),$$

což můžeme rovnou dosadit do předchozí rovnice, kde využijeme pravidla pro rozdíl logaritmů a dostáváme

$$t = \frac{1}{\lambda_A + \lambda_B} \ln \frac{\lambda_A A_0 - \lambda_B B_0}{\lambda_A M - (\lambda_A + \lambda_B) B}.$$

Vyjádríme si  $B$ :

$$B = \frac{\lambda_A M - (\lambda_A A_0 - \lambda_B B_0) e^{-(\lambda_A + \lambda_B)t}}{\lambda_A + \lambda_B}.$$

Nyní už nám jen zbývá spočítat hledaný poměr. Jednoduchou algebru můžeme přeskočit a dostáváme se rovnou k výsledku

$$\frac{A}{B} = \frac{M - B}{B} = \frac{\lambda_B (A_0 + B_0) e^{(\lambda_A + \lambda_B)t} + (\lambda_A A_0 - \lambda_B B_0)}{\lambda_A (A_0 + B_0) e^{(\lambda_A + \lambda_B)t} - (\lambda_A A_0 - \lambda_B B_0)},$$

kteřý je řešením této úlohy.

Při využití druhého postupu nejprve označme  $K = A/B$ . Pro derivaci  $K$  podle času po dosazení z (10) dostáváme

$$\frac{dK}{dt} = \frac{d}{dt} \frac{A}{B} = \frac{\frac{dA}{dt} B - A \frac{dB}{dt}}{B^2} = (\lambda_B - \lambda_A K) (1 + K).$$

Zavedeme substituci  $k = \lambda_A/\lambda_B$ . Rovnici separujeme a vzniklý integrál řešíme pomocí parciálních zlomků

$$\begin{aligned} t &= \frac{1}{\lambda_B} \int \frac{dK}{(1 - kK)(1 + K)} = \\ &= \frac{1}{\lambda_B(1+k)} \int \frac{k}{1 - kK} + \frac{1}{1 + K} dK = \frac{-\ln(1 - kK) + \ln(1 + K)}{\lambda_B(1+k)} + C. \end{aligned}$$

V čase  $t = 0$  jsme měli  $K = K_0 = A_0/B_0$ . Dosazením do rovnice si můžeme vyjádřit integrační konstantu

$$C = -\frac{1}{\lambda_B + \lambda_A} \ln \frac{1 + K_0}{1 - kK_0}.$$

Po úpravě můžeme vyjádřit funkci

$$\begin{aligned} K &= \frac{e^{(t-C)(\lambda_A + \lambda_B)} - 1}{ke^{(t-C)(\lambda_A + \lambda_B)} + 1} = \frac{(1 + K_0) e^{(\lambda_A + \lambda_B)t} - (1 - kK_0)}{(1 + K_0) ke^{(\lambda_A + \lambda_B)t} + (1 - kK_0)} = \\ &= \frac{\lambda_B (A_0 + B_0) e^{(\lambda_A + \lambda_B)t} + (\lambda_A A_0 - \lambda_B B_0)}{\lambda_A (A_0 + B_0) e^{(\lambda_A + \lambda_B)t} - (\lambda_A A_0 - \lambda_B B_0)}, \end{aligned}$$

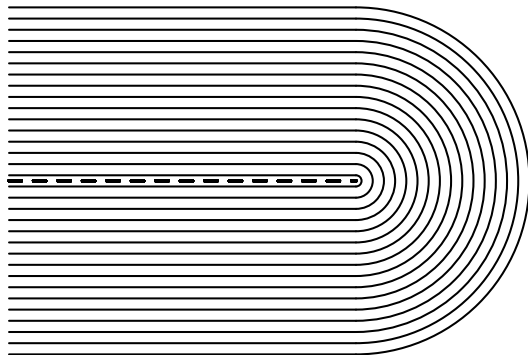
kteřá je opět řešením této úlohy. Vidíme, že oba postupy vedly na stejný výsledek.

Všimněte si, že oba postupy fungují jenom za předpokladu  $\lambda_A A - \lambda_B B \neq 0$ . Můžeme ale ověřit, že pokud tato podmínka platí pro počáteční hodnoty, bude platit vždy (a jinak je úloha triviální).

## Úloha III.P . . . složený papír

Každý to jistě někdy slyšel a určitě i zkusil: „List papíru nelze na půlku přeložit více než sedmkrát.“ Je to ale skutečně pravda? Najděte hraniční podmínky.

Je dobré si uvědomit, že hlavní problém je se stále se zvyšující tloušťkou, která narůstá exponenciálně.<sup>30</sup> Uvažujme několik vrstev papíru položených na sobě a přeložme celou hromadu napůl, výsledek by měl vypadat přibližně jako na obrázku 15.



Obr. 15: Obrázek ilustrující jak vypadá přeložení, když překládáme více vrstev papíru najednou.

Pokud si domyslíme třetí rozměr, bude mít každá vrstva tvar povrchu válce. Ten má nulovou Gaussovu křivost, díky čemuž máme možnost tento tvar složit z listu papíru. Nyní můžeme provést další přehyb okolo osy ležící v zatím rovinné části papíru. Tuto osu můžeme volit více způsoby. Buď osu přehybu zvolíme rovnoběžně s první osou, nebo k ní kolmo, nebo s ní svírající obecný úhel. Pokud zvolíme osu rovnoběžně, vytvoříme další válec s nulovou křivostí, kdežto pokud zvolíme kolmo k původní, vytváříme povrch toru, který má křivost nenulovou, a proto klade papír mnohem větší odpor, protože se musí deformovat i v ploše papíru.<sup>31</sup> Proto budeme uvažovat papír efektivně jednodimenzionální, jako bychom napůl přehýbali dlouhý a úzký pás papíru.

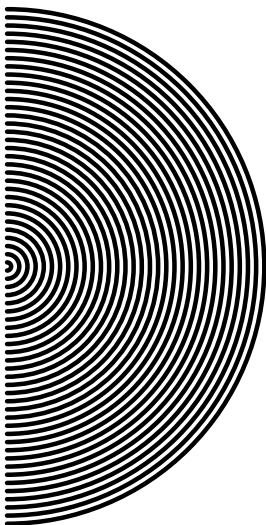
Dále si všimněme, že při přehybu napůl polovina každé vrstvy skončí nad střední rovinou<sup>32</sup> a druhá pod ní, což znamená, že touto rovinou musí procházet tolik vrstev, kolik vrstev přehýbáme. Z toho rovnou můžeme vyvodit že papír nelze přehýbat donekonečna, protože kvůli zachování celkového objemu bychom potřebovali šířku paklíku ztenčovat donekonečna, což ale evidentně nelze.<sup>33</sup>

<sup>30</sup>Můžete si sami spočítat, že běžný papír přeložený dvaatřicetkrát napůl by byl vysoký až na oběžnou dráhu měsíce.

<sup>31</sup>Při vnějším okraji se natahuje a při vnitřním se křadí, jak si můžete sami doma vyzkoušet.

<sup>32</sup>V obrázku vyznačena čárkovaně.

<sup>33</sup>Pokud bychom skládali papír do harmoniky, toto omezení není a můžeme ho přeložit vícekrát než u překládání napůl.



Obr. 16: Výsledný tvar, který už vícekrát nejde přeložit

Lze tedy udělat horní odhad pro maximální počet přehybů. Představme si stav jako na obrázku 16. Pak můžeme spočítat celkovou délku papíru potřebnou pro vytvoření takového obrazce pro daný počet vrstev. Označme  $d$  tloušťku papíru a  $l$  celkovou délku potřebného papíru. Uvažujme situaci rovnou s  $2^{n-1}$  vrstvami, což je stav po  $n - 1$  přehybech napůl. Uvědomme si, že protože délka oblouku roste s jeho poloměrem lineárně, součet obvodů vnějšího a vnitřního oblouku je stejný jako dvojnásobek délky uprostřed vrstvy. Totéž platí pro další vrstvy. Tedy celková délka  $2^{n-1}$  vrstev o poloměrech od 0 do  $2^{n-1}d$  je stejná jako délka  $2^{n-1}$  vrstev o poloměru  $2^{n-1}d/2$ . Dosazením dostáváme vzorec pro délku pro  $n$  vrstev papíru

$$l = \frac{\pi}{2} 2^{2n-2} d \frac{\pi}{8} 4^n d, \quad (11)$$

z čehož dokážeme vyjádřit maximální počet přehybů  $n$  pro fixní  $l$

$$n_{\max} = \log_4 \left( \frac{8l}{\pi d} \right) = \log_4 \left( \frac{8}{\pi} \eta \right), \quad (12)$$

kde jsme zavedli  $\eta = \frac{l}{d}$  jako bezrozměrný parametr překládaného papíru.

Pokud bychom tento odhad chtěli zpřesnit, musíme uvažovat i papír spotřebovaný na všechny předchozí přehyby. Získáme tak model přehybů odpovídající obrázku 17.

Celkovou délku papíru získáme sumou přes všechny přehyby, kde délka každého přehybu je dána vzorcem (11), takže pro  $n$  přehybů dostáváme

$$l = \sum_{k=1}^n \frac{\pi}{8} 2^{2k} d.$$



Obr. 17: Výsledný tvar, který již nejde vícekrát přeložit se zakreslením předchozích přehybů

Tuto sumu ale dokážeme převést na součet geometrické řady

$$l = \sum_{k=1}^n \frac{\pi}{8} 2^{2k} d = \frac{\pi d}{8} \sum_{k=1}^n 4^k = \frac{\pi d}{2} \frac{4^n - 1}{4 - 1} \approx \frac{\pi d 4^n}{6}.$$

Poslední aproximaci lze použít s rozumnou přesností pro  $n \geq 2$ . Ze vzorce vidíme, že na přidání dalšího přehybu potřebujeme nikoliv dvojnásobný pás papíru, ale dokonce čtyřnásobný.<sup>34</sup> Maximální počet přehybů je pak dán vzorcem

$$n_{\max} \approx \frac{1}{2} \log_2 \left( \frac{6}{\pi} \eta \right). \quad (13)$$

Dosazením hodnot pro papír formátu A4, pokud děláme přehyby jen po tom samém směru (tak, abychom přehýbali tu delší hranu), které jsme odhadli na  $d = 0,1$  mm,  $l = 300$  mm,  $\eta = 3000$ , získáváme  $n \approx 6,24$ . To se nám skutečně potvrdilo (konzistentně s tím, že se jedná o horní odhad), když jsme překládali papír a tímto způsobem ho skutečně přehli pětikrát, tak přehyb pošesté vypadal sice obtížně, ale nezdál se nemožný. Ale pak by rozhodně nešel papír přehýbat v druhém směru, protože by nám nedržely vrstvy na sobě.

Nyní ale máme teoreticky ještě druhý směr, ve kterém je 20 cm nevyužitého papíru. Bohužel ale už tloušťka není tloušťka papíru, ale součet všech vrstev z předchozího přehýbání. V našem příkladu je to  $d = 0,1 \cdot 2^5$  mm = 3,2 mm, tudíž  $\eta = 62,5$ . Potom  $n \doteq 3,45$ , což se také potvrdilo, protože jsme papír přehli ještě třikrát ve druhém směru.<sup>35</sup> Vidíme, že kancelářský papír formátu A4 lze přeložit napůl minimálně osmkrát, tedy často opakované tvrzení o sedmi přeloženích neplatí.

<sup>34</sup>To ale vyplývá už ze (12), pokud se nad ním trochu zamyslíme.

<sup>35</sup>Nutno podotknout, že po pěti přehybech jsme naměřili šuplerou tloušťku jen 3 mm, což je nižší hodnota než ta použitá v odhadech.

Pokud překládáme vrstvy střídavě nebo nějak jinak, musíme začít uvažovat ne o délce papíru, ale o celkové ploše papíru ztracené v přehybech. Ta je ale, protože papír má tvar válce, dána vzorcem (11) přenásobeném zbývající délkou v druhém směru. Proto, pokud chceme snížit množství plochy ztracené v posledních přehybech (kdy se ztrácí nejvíce papíru), musíme mít co nejmenší délku ve směru kolmém na přehýbání. Toho dosáhneme právě tehdy, když nejprve přehneme podél jedné strany maximální počet přehybů, které nám papír dovolí, a pak překládáme ve druhém směru. Z toho je vidět, že daný papír nemůžeme obecně přeložit vícekrát, pokud střídáme přehyby, než pokud přehýbáme papír nejdříve v jednom směru a pak v druhém směru. Jen pokud bychom měli papír s takovými rozměry, že maximální počet přehybů v obou směrech<sup>36</sup> podle vzorce (13) vychází těsně menší, než nejbližší vyšší celé číslo, můžeme nějakým konkrétním proházením pořadí přehybů dosáhnout toho, že maximální počet zvýšíme o trochu v jednom směru na úkor směru druhého. Tím sice celkový součet nezaokrouhlených čísel klesne, ale po zaokrouhlení (samozřejmě dolů) získáme jeden přehyb navíc tím, že jeden ze směrů dostaneme přes celé číslo, zatímco to druhé si „nezkazíme“. Ale správné pořadí je nutné najít pro každé rozměry papíru zvlášť.

Ještě musíme říct, že všechny uvedené vzorce jsou jen horním odhadem, protože jsme neuvažovali papír ztrácející se jinak než ve válcových přehybech. Jeden ze zdrojů dalších ztrát byl popsán na začátku tohoto textu, tedy deformace papíru a z toho vyplývající menší doléhání papíru při přehybech. Z tohoto důvodu je též výhodné nestřídat směry a začít kratší stranou. Druhý problém souvisí s rozjížděním papíru, který překládáme. Jednak, když překládáme více vrstev, ve vrstvách na vnitřní straně se ztrácí mnohem méně, než na těch vnějších, čímž se vůči sobě konce jednotlivých vrstev posunou. Tomu se lze vyhnout nejlépe tak, že odhadneme posunutí v jednotlivých vrstvách a budeme dělat příslušné přehyby mírně asymetricky.<sup>37</sup> Zároveň, pokud po překládání v jednom směru bude zbylý proužek příliš úzký, vrstvy při překládání ve směru druhém nebudou držet na sobě, což se stane například tehdy, pokud A4 překládáme nejdříve podél kratší strany. Co se týče síly potřebné k přeložení vrstvy, ta sice narůstá, ale je samozřejmě možné překládanou vrstvu rozložit na dílčí části, které lze přeložit zvlášť.<sup>38</sup> Proto jsme tento problém nepovažovali za omezující faktor.

Na závěr se sluší podotknout, že se tímto přístupem proslavila americká studentka Britney Gallivanová, která má díky tomu vlastní článek na anglické Wikipedii. S řešením, včetně experimentálního potvrzení hypotézy a nového rekordu v počtu přehnutí přišla na střední škole ve svých sedmnácti letech. My jsme získali trochu jiný vzorec, protože jsme řešení hledali nezávisle, takže jsme použili jiný předpoklad, kde bereme poloměr vrstvy. U nás to byl střed, u Gallivanové pak vnější okraj.

<sup>36</sup>V druhém směru ale musíme použít celkovou tloušťku po provedení přehybů ve směru prvním.

<sup>37</sup>Bohužel musíme pro každý přehyb počítat s posunutím ve všech dalších, což vyžaduje opravdu dokonalé plánování.

<sup>38</sup>U prvního přehybu to sice nejde, ale na druhou stranu nepřekládáme železný plech.

## Úloha IV.1 . . . zmrzlina

Odhadněte, kolik gramů zmrzliny dokážeme vyrobit, pokud máme k dispozici 5 l kapalného dusíku o teplotě  $-196\text{ }^\circ\text{C}$  a neomezené množství mléka a smetany o pokojové teplotě  $22\text{ }^\circ\text{C}$ ? Předpokládejme, že požadovaná zmrzlina se skládá jen z mléka a smetany (hmotnostně půl na půl) a měla by mít teplotu  $-5\text{ }^\circ\text{C}$ . Protože se tepelné kapacity mléka a smetany v tomto intervalu teplot značně mění, počítejte s jejich průměrnými hodnotami  $c_m = 3,45\text{ kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$  pro mléko a  $c_s = 4,45\text{ kJ}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$  pro smetanu. Zbylé potřebné údaje si dohledejte na internetu.

V tomto příkladu budeme řešit tepelné rovnováhy, kdy teplo přijaté dusíkem se bude rovnat teplu odevzdanému zmrzlinou

$$Q_{\text{dod}} = Q_{\text{od}}.$$

Označme si zadané teploty  $t_1 = 22\text{ }^\circ\text{C}$ ,  $t_2 = -196\text{ }^\circ\text{C}$  a  $t_3 = -5\text{ }^\circ\text{C}$  a podívejme se nejprve na druhou část. Použijeme známý vzoreček  $Q = cm\Delta T$ , kde  $c$  je tepelná kapacita,  $m$  je hmotnost a  $\Delta T$  změna teploty. Odtud, protože zmrzlina je napůl mléko a napůl smetana, nám vychází

$$Q_{\text{od}} = c_z m_z (t_1 - t_3) = \frac{c_s + c_m}{2} m_z (t_1 - t_3),$$

kde  $m_z$  je celková hmotnost zmrzliny a  $c_z$  je její celková tepelná kapacita .

Kapalný dusík se při kontaktu s teplejším tělesem začne odpařovat. Nalezli jsme měrné skupenské teplo varu<sup>39</sup>  $l_N = 198\text{ kJ}\cdot\text{kg}^{-1}$ . Všimněme si, že vynecháváme ohřívání dusíku na teplotu  $-5\text{ }^\circ\text{C}$ . To je z toho důvodu, že plynný dusík samovolně vyprchá z nádoby, protože jeho objem je několikanásobně vyšší. Počítejme tedy jeho ohřev před vyprchááním za tepelné ztráty.

Protože známe pouze objem kapalného dusíku, budeme si muset jeho hmotnost dopočítat sami. Nejedná se o 5 kg plynného dusíku, proto si musíme nalézt jeho hustotu v kapalném stavu,<sup>40</sup> která činí  $\rho = 808\text{ kg}\cdot\text{m}^{-3}$ . Nyní už můžeme psát

$$Q_{\text{dod}} = l_N m_N = l_N \rho V_N,$$

kde  $m_N$  značí hmotnost dusíku.

Spojením všech tří rovnic dostáváme

$$\frac{c_s + c_m}{2} m_z (t_1 - t_3) = l_N \rho V_N,$$

odkud si můžeme vyjádřit hledanou hmotnost zmrzliny

$$m_z = \frac{2l_N \rho V_N}{(c_s + c_m)(t_1 - t_3)}.$$

Dosažením zjistíme, že jsme vyrobili  $m_z \doteq 7,5\text{ kg} = 7\,500\text{ g}$  zmrzliny.

I přes zanedbání veškerého ohřevu dusíku je výsledek nereálný kvůli mnoha dalším tepelným ztrátám.

<sup>39</sup>[http://www.ivt.mzf.cz/wp-content/uploads/fyzika/tep\\_konst\\_latek-zmeny\\_skupenstvi.pdf](http://www.ivt.mzf.cz/wp-content/uploads/fyzika/tep_konst_latek-zmeny_skupenstvi.pdf)

<sup>40</sup><https://en.wikipedia.org/wiki/Nitrogen>

## Úloha IV.2 . . . autisti

Kolik nejméně dětí by muselo roztočit svůj fidget spinner, aby se tak den na Zemi prodloužil o 1 ms? Všechny neznámé veličiny odhadněte.

Řešení úlohy je založeno na zákonu zachování momentu hybnosti. Máme-li izolovanou soustavu (v našem případě je to Země s fidget spinnery), její celkový moment hybnosti se zachovává. Pro velikost momentu hybnosti vzhledem k ose otáčení platí vztah

$$L = J\omega,$$

kde  $\omega$  je úhlová rychlost rotace kolem dané osy a  $J$  je moment setrvačnosti vzhledem k této ose. Pro moment setrvačnosti Země platí  $J_Z = \frac{2}{5}MR^2$ , kde  $M$  je hmotnost Země a  $R$  je její poloměr.

Abychom pozemský den prodloužili, musíme zpomalit zemskou rotaci. To uděláme tak, že jí „ukradneme“ část momentu hybnosti tím, že fidget spinnery roztočíme stejným směrem. Protože součet momentů hybnosti musí zůstat konstantní, moment hybnosti samotné Země se sníží. Označíme-li moment setrvačnosti fidget spinneru  $J_s$  a maximální úhlovou rychlost, kterou je obyčejné dítě schopné svůj spinner roztočit,  $\omega_s$ , pro celkový moment hybnosti všech spinnerů dostaneme

$$L_s = N J_s \omega_s,$$

kde  $N$  je hledaný počet dětí. Je třeba zdůraznit, že všechny spinnery se musí točit stejným směrem a ideálně i jejich roviny otáčení musí být rovnoběžné. Autisti na rovníku je musí nechat rotovat kolmo na povrch Země, autisti na pólech zase rovnoběžně s povrchem Země. Ostatní děti musí držet spinner pod určitým náklonem vzhledem k povrchu podle toho, v jaké zeměpisné šířce se nacházejí.

V našem případě se moment setrvačnosti Země má změnit z

$$L_0 = J_Z \omega_Z$$

na

$$L = J_Z \omega'_Z,$$

kde

$$\omega_Z = \frac{2\pi}{T_Z} = \frac{2\pi}{1 \text{ den}}$$

a

$$\omega'_Z = \frac{2\pi}{T_Z + \Delta T_Z} = \frac{2\pi}{1 \text{ den} + 1 \text{ ms}}.$$

Aproximací  $\Delta T_Z \ll T_Z$  dostáváme

$$L_0 - L = J_Z(\omega_Z - \omega'_Z) = J_Z \left( \frac{2\pi}{T_Z} - \frac{2\pi}{T_Z + \Delta T} \right) \approx J_Z \frac{2\pi \Delta T}{T_Z^2}.$$

Tato změna momentu je rovna momentu hybnosti spinnerů

$$L_0 - L = L_s,$$



$$J_Z \frac{2\pi\Delta T}{T_Z^2} = N J_s \omega_s,$$

$$N = \frac{4\pi M R^2 \Delta T}{5 J_s \omega_s T_Z^2}.$$

Nyní zbývá jen odhadnout neznámé veličiny fidget spinneru ( $J_s$  a  $\omega_s$ ).

Ačkoliv některé kvalitní kovové spinnery mohou vážit i přes 120 g, nejčastěji používaný typ fidget spinneru má obvykle hmotnost kolem 50 g a poloměr zhruba 35 mm.<sup>41</sup> Moment setrvačnosti by bylo poměrně komplikované spočítat přesně, protože spinnery většinou nejsou homogenní. Proto je vhodné použít aproximaci. Můžeme například „od oka“ odhadnout, jaký poloměr by měl disk se stejnými parametry (tím bychom dostali správný řádový odhad). My použijeme sofistikovanější metodu. Všimneme si, že většina hmoty se nachází v prstencových kovových závažích, které se nacházejí v každé ze tří hlavic (někdy jsou jako závaží použita ložiska). Tato závaží aproximujeme kružnicemi s poloměrem 10 mm a se středem ve vzdálenosti 25 mm od středu spinneru.

Moment setrvačnosti nezávisí na tom, kolik má fidget spinner hlavic, protože moment setrvačnosti každé samotné hlavy je stejný a přímo úměrný její hmotnosti, která je zase nepřímo úměrná počtu hlav. Moment setrvačnosti kružnice vzhledem k její ose je stejný jako moment setrvačnosti hmotného bodu ve stejné vzdálenosti od osy

$$J_k = m r_k^2.$$

Moment vůči středu spinneru spočítáme pomocí Steinerovy věty

$$J_s = m r^2 + J_k,$$

kde  $m = 50$  g,  $r_k = 10$  mm a  $r = 25$  mm jsou výše odhadnuté hodnoty. Po dosazení dostáváme  $J_s \doteq 4 \cdot 10^{-5} \text{ kg}\cdot\text{m}^2$ .<sup>42</sup>

Zbývá odhadnout, jak rychle lze spinner roztáčet. Pokud nemáme k dispozici stlačený vzduch, vrtačku či jiný mechanismus, může se nám podařit spinner roztáčet až na 30 otáček za sekundu (to už chce zkušenost s točením a silné prsty). Tedy  $\omega_s \doteq 190 \text{ rad}\cdot\text{s}^{-1}$ .

Dosazením těchto hodnot do vztahu pro  $N$  dostáváme

$$N \doteq 1 \cdot 10^{28},$$

což je tak velké číslo, že se o změny délky dne kvůli lidem s ADHD vůbec nemusíme bát.

<sup>41</sup>Pokud nějaký máte, můžete si ho zkusit zvážit a změřit.

<sup>42</sup>Také jsme mohli spinner aproximovat homogenním diskem (usnadnilo by to výpočet na úkor přesnosti)  $J'_s = \frac{1}{2} m r_s^2 \doteq 3 \cdot 10^{-5} \text{ kg}\cdot\text{m}^2$ , kde  $r_s$  je poloměr disku.

### Úloha IV.3 . . . divně tvarovaná nádobka

Máme válcovou skleničku, která má z boku u dna malou díru o ploše  $S$ . Tato nádoba je naplněná vodou, která samovolně přetéká do druhé nádoby, která je tentokrát již bez díry. Jaký tvar by musela mít druhá nádoba, aby v ní hladina rostla rovnoměrně? Předpokládejte, že má být válcově symetrická.

Bonus Dna obou nádob jsou ve stejné výšce a nádoby jsou dírou spojené.

Předpokládejme kvazistacionární proudění, abychom mohli sestavit Bernoulliho rovnici pro proudnici od hladiny válcové nádoby v daném okamžiku až po výtokový otvor. Nechť  $\rho$  je hustota vody,  $p_a$  atmosférický tlak,  $v$  výtoková rychlost vody z otvoru,  $h_1$  výška hladiny ve válcové nádobě a  $v_1 = -\frac{dh_1}{dt} > 0$  je rychlost klesání hladiny. Bernoulliho rovnice pak má tvar

$$\frac{1}{2}\rho v_1^2 + p_a + h_1\rho g = \frac{1}{2}\rho v^2 + p_a.$$

Po odečtení atmosférického tlaku  $p_a$  a vydělení rovnice hustotou pak dostaneme

$$\frac{1}{2}v_1^2 + h_1g = \frac{1}{2}v^2. \quad (14)$$

Za předpokladu nestlačitelnosti vody plyne z rovnice kontinuity rovnost

$$S_1v_1 = Sv,$$

kde  $S_1$ , resp.  $S$ , je obsah příčného průřezu válcové nádoby, resp. jejího otvoru. Dosazením za  $v$  do rovnice (14) dostaneme

$$\frac{1}{2}v_1^2 + h_1g = \frac{1}{2}\left(\frac{S_1}{S}\right)^2 v_1^2.$$

Odtud můžeme vyjádřit rychlost klesání hladiny

$$v_1 = \sqrt{\frac{2h_1g}{\left(\frac{S_1}{S}\right)^2 - 1}}.$$

Dosazením  $v_1 = -\frac{dh_1}{dt}$  získáme diferenciální rovnici, kterou řešíme separací proměnných

$$-\frac{dh_1}{\sqrt{h_1}} = \sqrt{\frac{2g}{\left(\frac{S_1}{S}\right)^2 - 1}} dt.$$

Integrujeme od  $h_0$  do  $h_1$ , tedy od počáteční výšky hladiny do její výšky v čase  $t$

$$\int_{h_0}^{h_1} -\frac{dh_1}{\sqrt{h_1}} = \int_0^t \sqrt{\frac{2g}{\left(\frac{S_1}{S}\right)^2 - 1}} dt,$$

$$\left[-2\sqrt{h_1}\right]_{h_0}^{h_1} = \left[\sqrt{\frac{2g}{\left(\frac{S_1}{S}\right)^2 - 1}} t\right]_0^t.$$

Dosazením integračních mezí dostáváme

$$-2\sqrt{h_1} + 2\sqrt{h_0} = \sqrt{\frac{2g}{\left(\frac{S_1}{S}\right)^2 - 1}} t,$$

odkud vyjádříme

$$h_1 = h_0 - \sqrt{\frac{2h_0g}{\left(\frac{S_1}{S}\right)^2 - 1}} t + \frac{g}{\left(\frac{S_1}{S}\right)^2 - 1} \frac{t^2}{2}.$$

Derivací a drobnými algebraickými úpravami získáme vztah pro rychlost klesání hladiny v první nádobě

$$v_1 = \sqrt{\frac{2h_0g}{\left(\frac{S_1}{S}\right)^2 - 1}} - \frac{g}{\left(\frac{S_1}{S}\right)^2 - 1} t.$$

Odtud můžeme dosadit do rovnice kontinuity ve tvaru  $S_1v_1 = S_2v_2$ , kde  $v_2$  je rychlost růstu hladiny ve druhé nádobě a  $S_2$  je obsah jejího plošného průřezu. Ze zadání víme, že  $v_2$  je konstantní, tedy  $v_2 = h_2/t$ . Odtud si vyjádříme čas. Také platí  $S_2 = \pi r_2^2$ , kde  $r_2$  je hledaný poloměr druhé nádoby. Dostáváme pro něj

$$r_2 = \sqrt{\frac{S_1}{\pi v_2} \left( \sqrt{\frac{2h_0g}{\left(\frac{S_1}{S}\right)^2 - 1}} - \frac{g}{\left(\frac{S_1}{S}\right)^2 - 1} \frac{h_2}{v_2} \right)}.$$

Tím máme analyticky zadaný tvar druhé nádoby, tedy závislost jejího poloměru na výšce ode dna. Závislost výšky na poloměru je

$$h_2 = \frac{v_2}{g} \sqrt{\left(\frac{S_1}{S}\right)^2 - 1} \left( \sqrt{2h_0g} - \frac{\pi r_2^2 v_2}{S_1} \sqrt{\left(\frac{S_1}{S}\right)^2 - 1} \right).$$

Jedná se o rovnici paraboly s maximem v jejím vrcholu, jejíž osa leží na ose nádobky. Nádobka má tedy tvar části rotačního paraboloidu.

V případě, že  $S \ll S_1$ , můžeme v Bernoulliho rovnici (14) zanedbat první člen. Pro výtokovou rychlost potom platí

$$v = \sqrt{2h_1g}$$

a pro rychlost klesání hladiny v první nádobě tak dostáváme

$$v_1 = \frac{S}{S_1} \sqrt{2h_1g}.$$

Podobným postupem jako výše bychom se dostali k výsledku

$$r_2 = \sqrt{\frac{S}{\pi v_2} \left( \sqrt{2h_0g} - \frac{Sg}{S_1} \frac{h_2}{v_2} \right)}.$$

**Bonus**

Bernoulliho rovnice pro hladiny válcové a divně tvarované nádoby má tvar

$$\frac{1}{2}\rho v_1^2 + p_a + h_1 \rho g = \frac{1}{2}\rho v_2^2 + p_a + h_2 \rho g,$$

kde všechny veličiny jsou definovány jako dříve. Jednoduchými úpravami ji dostaneme do tvaru

$$\frac{1}{2}v_1^2 + h_1 g = \frac{1}{2}v_2^2 + h_2 g. \quad (15)$$

Derivací podle času dostáváme

$$v_1 \frac{dv_1}{dt} - v_1 g = v_2 g,$$

kde jsme první člen zderivovali podle pravidla pro derivaci složené funkce a o  $v_2$  víme, že je konstantní. Nyní separujeme proměnné a obě strany rovnice integrujeme

$$\int \frac{v_1}{v_1 + v_2} dv_1 = \int g dt.$$

První integrál vyřešíme částečným dělením integrandu, neboli

$$\int \frac{v_1}{v_1 + v_2} dv_1 = \int \frac{v_1 + v_2 - v_2}{v_1 + v_2} dv_1 = \int \left(1 - \frac{v_2}{v_1 + v_2}\right) dv_1 = v_1 - v_2 \ln(v_1 + v_2) + C_1.$$

Druhý integrál je triviální, celkově vychází

$$v_1 - v_2 \ln(v_1 + v_2) = gt + C, \quad (16)$$

kde  $C$  je již jediná integrační konstanta (zde jsme výhodněji použili neurčitě integrály s integračními konstantami místo určitých integrálů), kterou určíme z počátečních podmínek. V čase  $t = 0$  je výška hladiny válcové nádoby  $h_0$  a klesá rychlostí  $v_0$ . Odtud dostáváme

$$v_0 - v_2 \ln(v_0 + v_2) = C. \quad (17)$$

Z rovnice (15) si vyjádříme rychlost

$$v_0 = \sqrt{v_2^2 - 2h_0g},$$

kteřou dosadíme do (17), a tím získáme vztah pro hledanou integrační konstantu

$$C = \sqrt{v_2^2 - 2h_0g} - v_2 \ln\left(\sqrt{v_2^2 - 2h_0g} + v_2\right).$$

Odtud můžeme dosadit do (16). Stejně tak dosadíme za  $v_1 = \frac{\pi r_2^2}{S_1} v_2$  a za  $t = \frac{h_2}{v_2}$ . Celkově píšeme

$$\frac{\pi r_2^2}{S_1} v_2 - v_2 \ln\left(\frac{\pi r_2^2}{S_1} v_2 + v_2\right) = \frac{h_2 g}{v_2} + \sqrt{v_2^2 - 2h_0g} - v_2 \ln\left(\sqrt{v_2^2 - 2h_0g} + v_2\right).$$

Nyní můžeme konečně vyjádřit

$$h_2 = \frac{v_2^2}{g} \left( \frac{\pi r_2^2}{S_1} - \sqrt{1 - \frac{2h_0 g}{v_2^2}} + \ln \left( \frac{\sqrt{1 - \frac{2h_0 g}{v_2^2}} + 1}{\frac{\pi r_2^2}{S_1} + 1} \right) \right),$$

což je vztah závislosti výšky divně tvarované nádoby na poloměru  $r_2$ . Bezpochyby by byla zajímavější opačná závislost, tedy závislost poloměru na výšce. Ta bohužel analyticky vyjádřit nejde, protože rovnice je transcendentní vzhledem k proměnné  $r_2$ . Ještě dodejme, že při volbě počátečních podmínek jsme mohli zvolit kromě výšky  $h_0$  např. i obsah dna divně tvarované nádoby nebo rychlost klesání hladiny ve válcové nádobě v čase  $t = 0$ . Dále si povšimneme, že při řešení bonusové úlohy již nemůžeme zanedbat první člen levé strany Bernoulliho rovnice.

#### Úloha IV.4 . . . vymyslete si sami

Máme černou skříňku se třemi výstupy ( $A$ ,  $B$  a  $C$ ). Víme, že obsahuje  $n$  rezistorů se stejným odporem, ale nevíme jak jsou zapojeny. Změříme tedy odpory mezi dvojicemi bodů  $AB$ ,  $BC$  a  $CA$  a zjistíme, že  $R_{AB} = 3\Omega$ ,  $R_{BC} = 5\Omega$  a  $R_{CA} = 6\Omega$ . Zjistěte, kolik nejméně rezistorů může skříňka obsahovat a určete příslušný odpor jednoho rezistoru.

Při řešení úlohy lze postupovat několika různými způsoby. Jelikož máme jen jeden druh součástky, kterou můžeme použít (víme, že všechny odpory uvnitř jsou stejné), můžeme zkoušet sestavovat různá zapojení mezi třemi výstupy a zjišťovat, jaké budou výsledné odpory. Neznámý počet odporů uvnitř nám ale bude práci značně ztěžovat. Dalším způsobem je nahradit si celý vnitřní obvod třemi různými odpory zapojenými do trojúhelníku nebo do hvězdy. My si ukážeme oba postupy.

##### *brute-force*

Označme si nejprve počet odporů  $n$ . Toto číslo budeme po jednom postupně zvyšovat a hledat, zda existuje dané zapojení.

Zároveň nás nebude zajímat konkrétní poloha bodů  $A$ ,  $B$  a  $C$ , protože ty můžeme přeznačením libovolně vyměňovat. Budeme se zajímat pouze o poměr výstupních odporů, protože ten je lineárně závislý na odporu  $R$  jednoho rezistoru. Hledáme poměr 3:5:6.

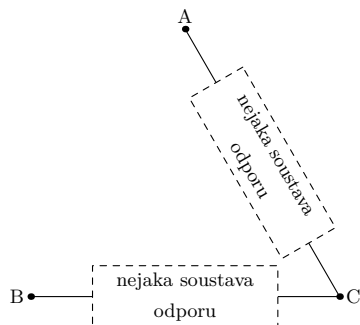
$$n = 1$$

Z jednoho odporu skříňku zřejmě nepostavíme.

$$n = 2$$

Povšimneme si, že se největší hodnota odporu mezi dvěma výstupy se nerovná součtu zbývajících dvou odporů. Dva odpory mezi třemi výstupy lze ale zapojit pouze sériově, tedy jeden výstupní odpor by musel být součtem předchozích dvou.

Toto je klíčové pozorování, které nám dále usnadní práci. Zjišťujeme tu, že obvod nemůže být zapojen tak, že každá cesta mezi dvěma vzdálenějšími výstupy



Obr. 18: Zapojení přes uzel.

vede přes prostřední výstup, jak je zobrazeno na obrázku 18. Proto už dále nebudeme takovou možnost zapojení uvažovat.

$$n = 3$$

Pro tři odpory máme pouze dvě možná zapojení: trojúhelník nebo hvězda (ne-uvážujeme zapojení vyloučená výše). Snadno nahlédneme, že obě tato zapojení jsou symetrická a vedla by tak k naměření stejných hodnot pro jakoukoliv dvojici výstupů.

$$n = 4$$

Víme, že každé uvažované zapojení tří odporů má tři osy symetrie. Přidáním jednoho dalšího rezistoru do obvodu se zbavíme pouze dvou z nich. Vždy nám jedna osa symetrie zůstane, což implikuje naměření stejných hodnot na dvou různých výstupech. Jelikož jsme naměřili tři různé hodnoty odporu, nelze vnitřní obvod sestavit ani ze čtyř rezistorů.

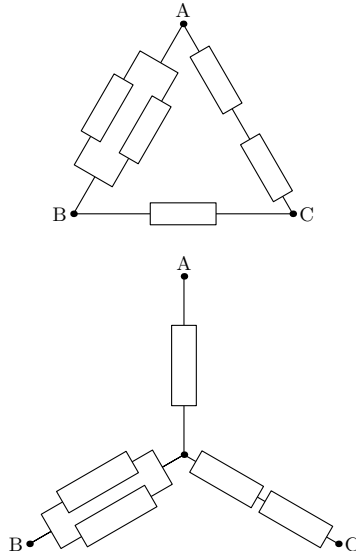
$$n = 5$$

Přidáním dalšího rezistoru už jsme schopni sestavit nesymetrický obvod. Začneme-li si vypisovat všechna možná zapojení pěti rezistorů mezi třemi výstupy, podaří se nám najít pouze tři nesymetrická zapojení.<sup>43</sup>

Jednotlivé výstupní odpory pro zapojení ve tvaru hvězdy vypočítáme jednoduše sečtením dvou „ramen“ spojujících dané vrcholy

$$\begin{aligned} R_{AB}^* &= R^* + \frac{R^{*2}}{R^* + R^*} = 1\frac{1}{2}R^* \text{ ,} \\ R_{BC}^* &= \frac{R^{*2}}{R^* + R^*} + R^* + R^* = 2\frac{1}{2}R^* \text{ ,} \\ R_{CA}^* &= 2R^* + R^* = 3R^* \text{ ,} \end{aligned}$$

<sup>43</sup>Důkaz zde uvádět nebudeme. Podařilo-li se vám najít nějaké další zapojení, zkuste si ho překreslit do co nejjednodušší podoby a zjistíte, že je buď symetrické, nebo je to jedno ze tří uvedených zapojení.



Poměr odporů mezi výstupy je tedy 3 : 5 : 6.

Výstupní odpory v zapojení do trojúhelníku spočítáme jako paralelní zapojení částí spojující dané body a větve procházející přes třetí vrchol

$$R_{AB}^{\Delta} = \frac{\frac{1}{2}R^{\Delta} \cdot 3R^{\Delta}}{R^{\Delta} + 2R^{\Delta} + \frac{1}{2}R^{\Delta}} = \frac{\frac{3}{2}R^{\Delta}}{\frac{7}{2}} = \frac{3}{7}R^{\Delta},$$

$$R_{BC}^{\Delta} = \frac{R^{\Delta} \cdot \left(2\frac{1}{2}\right)R^{\Delta}}{R^{\Delta} + 2R^{\Delta} + \frac{1}{2}R^{\Delta}} = \frac{\frac{5}{2}R^{\Delta}}{\frac{7}{2}} = \frac{5}{7}R^{\Delta},$$

$$R_{CA}^{\Delta} = \frac{2R^{\Delta} \cdot \left(1\frac{1}{2}\right)R^{\Delta}}{R^{\Delta} + 2R^{\Delta} + \frac{1}{2}R^{\Delta}} = \frac{\frac{6}{2}R^{\Delta}}{\frac{7}{2}} = \frac{6}{7}R^{\Delta}.$$

U tohoto zapojení nám vyšel také poměr 3 : 5 : 6.

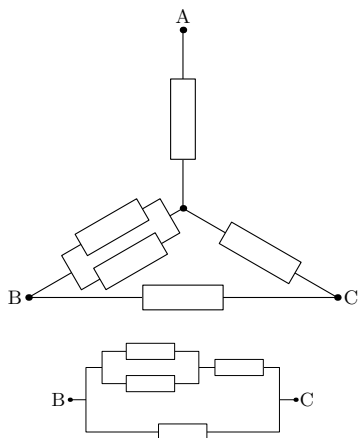
Výstupní odpory třetího zapojení spočítáme následovně (pomoci může přehlednější překreslení jednotlivých situací a použití základních pravidel pro dva sériově nebo paralelně zapojené rezistory)

$$R_{CA}^{\nabla} = R^{\nabla} + \frac{R^{\nabla} \frac{3}{2}R^{\nabla}}{R^{\nabla} + \frac{3}{2}R^{\nabla}} = \frac{8}{5}R^{\nabla}$$

$$R_{AB}^{\nabla} = R^{\nabla} + \frac{2R^{\nabla} \frac{1}{2}R^{\nabla}}{2R^{\nabla} + \frac{1}{2}R^{\nabla}} = \frac{7}{5}R^{\nabla}$$

$$R_{BC}^{\nabla} = \frac{R^{\nabla} \frac{3}{2}R^{\nabla}}{R^{\nabla} + \frac{3}{2}R^{\nabla}} = \frac{3}{5}R^{\nabla}.$$

Teď jsou odpory rozděleny v poměru 3 : 8 : 7, což není požadovaný poměr.



$$n \geq 6$$

Nemá smysl hledat dál, protože jsme již našli zapojení s menším počtem použitých rezistorů.

### Závěr

Zjišťujeme, že úloha má dvě možná řešení, přičemž oba obvody sestávají z pěti rezistorů. Odpor jednoho rezistoru zjistíme jednoduše. Stačí vzít např. odpor mezi body A a B a položit ho roven  $3\Omega$

$$1\frac{1}{2}R^* = 3\Omega \quad \Rightarrow \quad R^* = 2\Omega,$$

$$\frac{3}{7}R^\Delta = 3\Omega \quad \Rightarrow \quad R^\Delta = 7\Omega.$$

### Přímý výpočet

Další metodou výpočtu je převedení úlohy na hledání jednotlivých dílčích odporů. Začneme tím, co jsme ukázali v prvním obrázku, tzn. že obvod nemůže mít jediný uzel v některém výstupním bodě, a obvody tohoto typu nebudeme uvažovat.

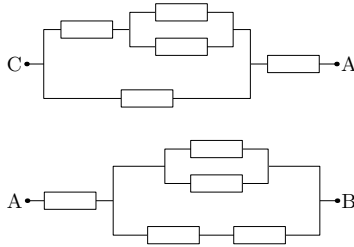
Složitějším zapojením tří bodů jsou sestavy trojúhelník a hvězda. Zkusíme tedy spočítat, jaké odpory by tato hvězda nebo trojúhelník musely obsahovat, a poté se je pokusíme seskládat ze samotných stejných odporů. Označme  $R_{AB} = 3\Omega$ ,  $R_{BC} = 5\Omega$  a  $R_{CA} = 6\Omega$ . Dále jednotlivé odpory, ze kterých se skládá hvězda, budeme značit  $R_A$ ,  $R_B$  a  $R_C$ . Můžeme si napsat soustavu tří rovnic

$$R_{AB} = R_A + R_B,$$

$$R_{BC} = R_B + R_C,$$

$$R_{CA} = R_C + R_A.$$





Řešením této soustavy je

$$R_A = \frac{R_{AB} - R_{BC} + R_{CA}}{2} = 2\Omega,$$

$$R_B = \frac{R_{BC} - R_{CA} + R_{AB}}{2} = 1\Omega,$$

$$R_C = \frac{R_{CA} - R_{AB} + R_{BC}}{2} = 4\Omega.$$

Vidíme, že vycházejí „pěkné“ hodnoty. Dále řešíme jen úvahou. Jestliže jeden odpor má  $2\Omega$ , sériové zapojení dvou takových rezistorů má  $4\Omega$  a paralelní zapojení těchto dvou rezistorů má  $1\Omega$ , což nám dává řešení úlohy v podobě prvního obrázku hvězdy s pěti použitými rezistory.

Kdybychom se na začátku rozhodli místo hvězdy počítat zapojení pro trojúhelník, dostali bychom značně komplikované rovnice,<sup>44</sup> ale jejich vyřešením také dostaneme poměr tří dílčích odporů 1:2:4.

I v tomto případě je potřeba ověřit, že žádné zapojení jednoho až čtyř rezistorů nevede na požadované odpory mezi výstupy. To provedeme stejně jako v metodě brute-force.

## Úloha IV.5 ... nemožnost nakažení

Představme si, že roztlačíme nějakou bakterii obvyklé velikosti na rychlost  $v = 50 \text{ km}\cdot\text{h}^{-1}$  ve vodorovném směru a necháme ji volně letět ve vzduchu. Jakou vzdálenost zhruba urazí, než se zastaví?

Výsledek vás možná překvapí. Jak je tedy možné se infikovat tímto způsobem bakteriální infekcí? Diskutujte, proč je to možné i přes takový výsledek.

Bakterie se v čase  $t$  bude nacházet na vodorovné souřadnici  $x(t)$  a bude mít vodorovnou rychlost  $\dot{x}(t)$ . Vzhledem k rychlostem a velikosti bakterie můžeme proudění vzduchu okolo považovat za laminární. Odporovou sílu působící na bakterii spočítáme ze Stokesova vztahu

$$F = -6\pi\eta r\dot{x},$$

<sup>44</sup>Výrazně jednodušší je nejprve spočítat hvězdu a poté jí převést na trojúhelník (to lze buď podle známých převodních vztahů nebo vlastním výpočtem).

kde  $\eta$  je dynamická viskozita a  $r$  je charakteristický rozměr bakterie. Po dosazení do pohybové rovnice  $F = m\ddot{x}$  dostaneme lineární homogenní diferenciální rovnici druhého řádu

$$\ddot{x} + \frac{6\pi\eta r}{m}\dot{x} = 0,$$

kteřou řešíme určením charakteristického polynomu.<sup>45</sup> Jeho kořeny jsou

$$\begin{aligned}\lambda_1 &= 0 \\ \lambda_2 &= -k,\end{aligned}$$

kde

$$k = \frac{6\pi\eta r}{m}.$$

Řešení rovnice potom hledáme ve tvaru

$$x = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t} = C_1 + C_2 e^{-kt}.$$

V čase  $t = 0$  je celková uražená dráha nulová, tedy

$$C_1 + C_2 = 0.$$

Rychlost v čase  $t$  získáme derivací

$$\dot{x} = -C_2 k e^{-kt},$$

což v kombinaci s druhou počáteční podmínkou  $\dot{x}(0) = v_0$  vede na rovnici

$$v_0 = -C_2 k.$$

Jednoduchou algebrou vyjádříme jednotlivé konstanty

$$\begin{aligned}C_1 &= \frac{v_0}{k}, \\ C_2 &= -\frac{v_0}{k}.\end{aligned}$$

Dosazením získáme parametrickou rovnici pohybu

$$x(t) = \frac{v_0}{k} (1 - e^{-kt}).$$

Z rovnice je vidět, že bakterie ve skutečnosti nikdy nezastaví, ale bude se limitně blížit do bodu

$$x_{\max} = \lim_{t \rightarrow \infty} x(t) = \frac{v_0}{k},$$

který tak představuje maximální vzdálenost, kterou může bakterie urazit, než se zastaví.

<sup>45</sup>Můžeme použít i separaci proměnných, ale užití charakteristického polynomu je v tomto případě jednodušší.

Dále bychom mohli uvažovat pohyb bakterie ve svislé ose. Díky tomu, že odporová síla je v rychlosti lineární, můžeme oba pohyby počítat zvlášť a výsledek získáme součtem jednotlivých pohybů.

Označme svislou souřadnici bakterie  $y(t)$ . Tíhová síla potom bude  $F_g = -mg$ , pro odporovou sílu platí  $F = -mk\dot{y}$ . V úlohách jako je tato těleso typicky velmi rychle dosáhne terminální rychlosti,<sup>46</sup> kterou můžeme z podmínky nulové výslednice sil určit jako

$$\dot{y}_{\text{term}} = -\frac{g}{k},$$

Dále už bakterie zrychlovat nebude. Nic nám však nebrání tento odhad ověřit výpočtem. Dosazením do pohybové rovnice dostáváme nehomogenní diferenciální rovnici

$$\ddot{y} + k\dot{y} = -g,$$

kteřou opět řešíme pomocí charakteristického polynomu. Jak už asi tušíte, získáme tím řešení homogenní rovnice, které je stejné jako u rovnice vodorovného pohybu. K tomu ještě musíme přičíst partikulární řešení, které v tomto případě najdeme snadno,<sup>47</sup> bude to lineární funkce  $t$ . Řešení hledáme ve tvaru

$$y = C_1 + C_2 e^{-kt} + C_3 t.$$

Opět vyjdeme z počátečních podmínek,  $y(0)$  a  $\dot{y}(0) = 0$ , tedy  $C_1 + C_2 = 0$ . Pro derivaci  $y$  podle času platí

$$\dot{y} = -C_2 k e^{-kt} + C_3,$$

což vede na vztah  $-C_2 k + C_3 = 0$ . Poslední podmínku získáme spočítáním druhé derivace  $y$  a dosazením do původní diferenciální rovnice

$$0 = \ddot{y} + k\dot{y} + g = C_2 k^2 e^{-kt} - C_2 k e^{-kt} + C_3 + g,$$

ze které vyplývá  $C_2 k(k-1) + C_3 + g = 0$ . Nyní už trošku pracnější, ale stále jednoduchou algebrou vyjádříme

$$\begin{aligned} C_1 &= \frac{g}{k^2} \\ C_2 &= -\frac{g}{k^2} \\ C_3 &= -\frac{g}{k}. \end{aligned}$$

Konečně tak můžeme napsat parametrickou rovnici pro pohyb ve svislém směru

$$y(t) = \frac{g}{k^2} (1 - e^{-kt}) - \frac{g}{k} t.$$

---

<sup>46</sup>Teoreticky nebude terminální rychlosti dosaženo nikdy, nicméně to můžeme považovat za dobrou aproximaci.

<sup>47</sup>Jedná se o rovnici se speciální pravou stranou.

Vidíme, že náš odhad byl správný. Rychlost se velmi brzy přiblíží hodnotě  $\dot{y}_{\text{term}}$ , protože hodnota  $k$  je vysoká (jak ukážeme níže) a člen  $e^{-kt}$  tedy velmi rychle konverguje k nule. Bakterie předtím sice urazí jistou vzdálenost, ta je však vzhledem k velikosti  $k^{-2}$  téměř zanedbatelná.

Nyní už nezbývá než dosadit číselné hodnoty. Dynamická viskozita vzduchu je při normálních podmínkách přibližně  $\eta = 1,8 \cdot 10^{-5}$  Pa · s. Bakterie jsou co do velikosti velmi rozmanité, ale můžeme odhadnout  $r = 1 \mu\text{m} = 1 \cdot 10^{-6}$  m. Hmotnost typické bakterie odhadneme jako  $m = 4 \text{ pg} = 4 \cdot 10^{-15}$  kg. Dosazením číselných hodnot zjistíme, že  $k \doteq 8,5 \cdot 10^4 \text{ s}^{-1}$ .

Pro vodorovnou vzdálenost, kterou bakterie urazí, dostáváme  $x_{\text{max}} \doteq 1,6 \cdot 10^{-4}$  m. To skutečně není daleko. Na druhou stranu, velikost terminální rychlosti ve svislém směru bude  $\dot{y}_{\text{term}} \doteq -1,2 \cdot 10^{-4} \text{ m} \cdot \text{s}^{-1}$ . Bakterie tedy po výstřelu nedoletí téměř nikam, ale také prakticky nebude padat na zem. Může se tak pohybovat díky větru, čímž dokáže urazit daleko větší vzdálenosti.

Ve skutečnosti to však není tak jednoduché, protože bakterie se typicky šíří v malých kapičkách s poloměrem řádově  $r' = 100 \mu\text{m} = 1 \cdot 10^{-4}$  m. Pokud předpokládáme, že se jedná o koule s hustotou podobnou hustotě vody, pro jejich hmotnost platí

$$m' = \frac{4}{3} \pi r_k^3 \rho,$$

z čehož pro neznámou konstantu dostáváme

$$k' = \frac{9\eta}{2r_k^2 \rho}.$$

Po dosazení číselných hodnot vychází  $k' \doteq 8 \text{ s}^{-1}$ , což vede na vzdálenost  $x' \doteq 1,7$  m. To už je docela reálný odhad toho, jak daleko běžně kýcháme. Pro rychlost klesání pak dostáváme  $\dot{y}'_{\text{term}} \doteq -1,2 \text{ m} \cdot \text{s}^{-1}$ .

Shrneme-li výsledky předchozích výpočtů, dojdeme k závěru, že jednotlivé bakterie ve stojícím vzduchu urazí jen minimální vzdálenost. Díky tomu se ale dokáží efektivně šířit větrem. Oproti tomu větší kapičky brzy spadnou na zem, ale předtím se stihnou dostat do vzdálenosti, která bezpečně stačí k nakažení všech lidí v blízkém okolí.

## Úloha IV.P ... Voyager II a Voyager I žijí!

*Máme nějaký satelit, který chceme vypustit ven ze Sluneční soustavy. Vypouštíme ho z oběžné dráhy Země tak, že po nějakých korekcích dráhy získá rychlost, která je vyšší než úniková rychlost ze Sluneční soustavy. Jaká je pravděpodobnost, že dojde ke kolizi sondy s nějakým kosmickým materiálem s průměrem větším než  $d = 1$  m před opuštěním Sluneční soustavy?*

Eudstvo za svoju existenciu vypustilo do vesmíru približne stovku medziplanetárnych sond,<sup>48</sup> z pomedzi ktorých päť (Voyager 1 a 2, Pioneer 10 a 11 a New Horizons) opúšťa Slnecnú sústavu. Najvzdialenejšia z nich je sonda Voyager 1, na začiatku

<sup>48</sup>[https://en.wikipedia.org/wiki/List\\_of\\_interplanetary\\_voyages](https://en.wikipedia.org/wiki/List_of_interplanetary_voyages)

roka 2018 vzdialená od Slnka asi 140 AU. V žiadnom prípade (aspoň pokiaľ je nám známe) nebola potvrdená zrážka s cudzím telesom. Niektoré sondy boli na kolízny kurz navedené zámerné, no mnoho sond, hlavne na začiatku kozmických letov, bolo stratených. Mohli za to technické problémy, alebo je vesmír plný „zabijáckych“ asteroidov?

### Pravdepodobnosť zrážky

Pustme sa do riešenia úlohy. Predpokladajme, že pozíciu planét poznáme a dokážeme sa im vyhnúť. Ďalej nech naša sonda letí v rovine ekliptiky ako väčšina sond. Nájďme najprv vzťah pre pravdepodobnosť zrážky našej sondy s jediným asteroidom. Pre jednoduchosť predpokladajme guľový tvar asteroidov a sondy s polomerami  $r_a$ ,  $r_s$ . Zrážka nastane, ak je vzdialenosť sondy od asteroidu menšia ako súčet ich polomerov. V sústave pevne spojenjej so sondou opíše asteroid obklopený „ochrannou zónou“ a pohybujúci sa rýchlosťou  $v'$  za čas  $\Delta t$  objem  $V_a = \pi (r_a + r_s)^2 v' \Delta t$ . Ak sa bude asteroid nachádzať v náhodnom mieste objemu  $V$  (teda jeho polohu nepoznáme, len vieme, že tam niekde je), dostávame pre pravdepodobnosť zrážky

$$p_0 = \frac{V_a}{V} = \frac{\pi (r_a + r_s)^2 v' \Delta t}{V}.$$

Opäť pre jednoduchosť predpokladajme, že asteroidy obiehajú okolo Slnka po kruhových trajektóriách a naša sonda sa vzdaluje zo Slnčnej sústavy po priamke prechádzajúcej cez Slnko s práve únikovou rýchlosťou v rovine ekliptiky. Potom

$$p_0 = \frac{\pi (r_a + r_s)^2 \sqrt{v_s^2 + v_a^2} \Delta t}{V}.$$

Pre kruhovú a únikovú rýchlosť platí  $v_k^2 = \frac{GM}{R}$ ,  $v_u^2 = \frac{2GM}{R}$ , ďalej  $v_s = \frac{\Delta R}{\Delta t}$ , teda

$$p_0(R, r_a) = \frac{\pi (r_a + r_s)^2 \sqrt{\frac{3}{2}} \Delta R}{V}.$$

### Viac zrážok

Pre celkovú pravdepodobnosť  $P$ , že našu sondu niečo zasiahne, máme

$$1 - P = \prod_i (1 - p_i),$$

čo po zlogaritmovaní a použitia odhadu  $\ln(1 + x) = x$  pre  $|x| \ll 1$  dá

$$P = \sum_i p_i,$$

ak predpokladáme, že všetky pravdepodobnosti sú malé (teda hlavne výsledná). Index  $i$  prebieha všetky telesá, ktoré nás zaujímajú. Príspevok telies vo vzdialenostiach  $R$  až  $R + \Delta R$  s polomerami  $r_a$  až  $r_a + \Delta r_a$  do celkovej pravdepodobnosti je

$$p(R, r_a) = \pi (r_a + r_s)^2 \sqrt{\frac{3}{2}} \frac{(r_a + r_s)^2}{V(R)} \frac{\partial N(R, r_a)}{\partial r_a} \Delta r_a \Delta R,$$

kde  $\frac{\partial N(R, r_a)}{\partial r_a} \Delta r_a$  je počet takýchto telies v objeme  $V(R)$ . Tento objem môžeme odhadnúť ako toroid s obdĺžnikovým prierezom, ktorý má polomer  $R$ , hrúbku  $\Delta R$  a výšku  $2R \sin i$ , kde  $i$  je odhad istej strednej inklinácie dráh asteroidov. Preto

$$V(R) = 4\pi R^2 \Delta R \sin i,$$

$$p(R, r_a) = \sqrt{\frac{3}{2}} \frac{(r_a + r_s)^2}{4R^2 \sin i} \frac{\partial N(R, r_a)}{\partial r_a} \Delta r_a,$$

Celkovú pravdepodobnosť dostaneme súčtom/integrálom cez všetky prípustné  $R$ ,  $r_a$ .

### Distribúcia veľkostí asteroidov

Zostáva teda určiť funkciu  $N(R, r_a)$ . Určíme najprv  $P(R)$ , pravdepodobnosť zásahu asteroidom ľubovoľnej veľkosti v intervale vzdialeností od Slnka  $R$ ,  $R + \Delta R$ . Pre závislosť počtu asteroidov od ich veľkosti približne platí vzťah

$$N(r_a > r) = Ar^B,$$

kde koeficient  $-2 > B > -3$  v závislosti na skupine a veľkosti objektov.<sup>49</sup> Pravdepodobne najväčší vplyv na parameter  $B$  majú vlastnosti materiálu, z ktorého sú asteroidy zložené. Koeficient  $A$  určíme pomocou polomeru desiateho najväčšieho asteroidu  $r_{10}$  v danej oblasti, čo je dohľadateľný údaj a zároveň už má istú štatistickú významnosť

$$10 = Ar_{10}^B \quad \Rightarrow \quad A = \frac{10}{r_{10}^B},$$

$$N(r_a > r) = 10 \left( \frac{r}{r_{10}} \right)^B.$$

Pre počet asteroidov s polomerami  $r_a$  až  $r_a + dr_a$  dostávame diferencovaním

$$\frac{\partial N(R, r_a)}{\partial r_a} dr_a = 10B \left( \frac{r_a}{r_{10}} \right)^B \frac{dr_a}{r_a}.$$

Preto vieme  $P(R)$  určiť ako integrál

$$P(R) = \int_{r_0}^{r_1} \sqrt{\frac{3}{2}} \frac{(r_a + r_s)^2}{4R^2 \sin i} \frac{\partial N(R, r_a)}{\partial r_a} dr_a,$$

kde  $r_0 = 0,5m$  je polomer najmenších asteroidov, ktoré nás zaujímajú,

$$P(R) = \int_{r_0}^{r_1} \frac{(r_a + r_s)^2}{r_a^{3+b}} \frac{5}{2} \sqrt{\frac{3}{2}} \frac{Br_{10}^{2+b}}{R^2 \sin i} dr_a,$$

<sup>49</sup>[https://commons.wikimedia.org/wiki/File:Asteroids\\_by\\_size\\_and\\_number.svg](https://commons.wikimedia.org/wiki/File:Asteroids_by_size_and_number.svg), orbit.psi.edu/~tricaric/pdf/skads.pdf alebo heslo *meteoroid* na en.wiki

$$P(R) = \frac{5}{2} \sqrt{\frac{3}{2}} \frac{B r_{10}^2}{R^2 \sin i} \int_{r_0}^{r_1} \frac{r_{10}^b (r_a + r_s)^2}{r_a^{3+b}} dr_a,$$

kde sme na niektorých miestach nahradili  $B = -2 - b$ ,  $0 < b < 1$ . Označme integrál ako  $-\alpha$ , po jeho vypočítaní a dosadení medzi ( $r_1 = 10^{-\frac{1}{B}} r_{10}$ ) s prihliadnutím na  $r_s \approx r_0 \ll r_{10}$  dostávame

$$\alpha = - \left( \frac{r_1^{2+B} - r_0^{2+B}}{2+B} - r_0^B \left( \frac{2}{1+B} r_0 r_s + \frac{1}{B} r_s^2 \right) \right) r_{10}^b.$$

Pre danú oblasť a veľkosť sondy je  $\alpha$  len bezrozmerná konštanta. Konečne pre pravdepodobnosť, že v oblasti, ktorej desiaty najväčší objekt má polomer  $r_{10}$ , dôjde k zrážke, dostávame

$$P(R) = -B \frac{5}{2} \sqrt{\frac{3}{2}} \frac{r_{10}^2}{R^2 \sin i} \alpha.$$

Celkovú pravdepodobnosť určíme súčtom dielčích pravdepodobností pre všetky oblasti.

### Medziplanetárna hmota

S rastúcou vzdialenosťou od Slnka musí naša sonda postupne prekonať:

- asteroidy skupín Apollo a Amor v priestore medzi dráhami Zeme a Marsu,
- križiče Marsu,
- hlavný pás asteroidov,
- skupinu Trójanov, ktorí sa nachádzajú v okolí libračných bodov L4, L5 Jupitera, teda približne  $60^\circ$  pred a za ním na jeho orbite,
- skupinu Centaurov, ktorí sa nachádzajú medzi dráhami jovialnych planét a majú výrazne excentrické orbity,
- Kuiperov pás za dráhou Neptuna, v ktorom sa nachádza aj Pluto,
- objekty rozptýleného disku,
- Oortov oblak komét.

Prvé dve menované skupiny však obsahujú len málo objektov<sup>50</sup> v porovnaní s hlavným pásom. Pre hlavný pás odhadneme  $R_{\min} = 2,2 \text{ km}$ ,  $R_{\max} = 3,2 \text{ km}$ ,  $i = 10^\circ$ <sup>51</sup> Pre náš výpočet odhadnime  $R = 2,5 \text{ AU}$ . Desiaty najväčší objekt v hlavnom páse je 15 Eunomia s polomerom  $r_{10} = 134 \text{ km}$ .

Skupine Trójanov sa väčšina sond vyhýba, keďže využíva prelet okolo Jupitera na zvýšenie rýchlosti. Vystavuje sa tým ale riziku zrážky s Jupiterovými satelitmi.

Pre Centaurov  $r_{10} \doteq 80 \text{ km}$ ,<sup>52</sup>  $i = 15^\circ$ ,  $R = 5,5 \dots 30 \text{ AU} \doteq 10 \text{ AU}$ <sup>53</sup> pričom pri poslednom odhade berieme do úvahy, že hľadáme priemer obrátených štvorcov, ktorý je vždy menší ako aritmetický.

<sup>50</sup>obrázok na <http://faculty.washington.edu/trq/hpcc/stawarz/orbres.html>

<sup>51</sup>[https://en.wikipedia.org/wiki/Asteroid\\_family#/media/File:AsteroidIncAu.png](https://en.wikipedia.org/wiki/Asteroid_family#/media/File:AsteroidIncAu.png)

<sup>52</sup><http://www.johnstonsarchive.net/astro/tnodiam.html>

<sup>53</sup>[https://en.wikipedia.org/wiki/Centaur\\_\(minor\\_planet\)#/media/File:TheKuiperBelt\\_42AU\\_Centaurs.svg](https://en.wikipedia.org/wiki/Centaur_(minor_planet)#/media/File:TheKuiperBelt_42AU_Centaurs.svg)

Pre objekty za dráhou Neptúna  $r_{10} \doteq 420 \text{ km}$ ,<sup>54</sup>  $i = 20^\circ$ ,  $R \doteq 45 \text{ AU}$ <sup>55</sup>

O tom, ako vyzerá Slnecná sústava vo vzdialenostiach väčších ako 100 AU, veľa nevieme, preto náš výpočet ukončíme práve tu. Dokonca sa špekuluje o existencii deviatej planéty.<sup>56</sup> Vo vzdialenostiach väčších ako niekoľko tisíc AU sa predpokladá existencia Oortovho oblaku komét s miliardami objektov s priemerom väčším ako 20 km.<sup>57</sup>

### Finálny výpočet

Tab. 1: Závislosť parametra  $\alpha$  na  $B$ ,  $r_{10}$

$\frac{r_{10}}{\text{km}}$	$B = -2.1$	$B = -2.5$	$B = -2.9$
80	60	4 000	400 000
134	60	5 000	600 000
420	70	9 000	1 600 000

Najprv sa bližšie pozrieme na hodnoty, ktoré nadobúda  $\alpha$ . Pre  $r_0 = 0,5 \text{ m}$ ,  $r_s = 1,5 \text{ m}$  a rôzne hodnoty  $B$  dostávame pre jednotlivé pásma hodnoty v tabuľke 1. Vidíme, že  $\alpha$  sa v závislosti od parametrov nachádza v intervale 50 až 2 000 000, pričom dominantný vplyv má parameter  $B$ . Pôjde teda o extrémne hrubý odhad. Vďaka tomu sú aj naše pôvodné zanedbania modelu oprávnené. Pre zaujímavosť sú v tabuľke 2 uvedené hodnoty  $\alpha$  pre rôzne hodnoty  $r_0$  pre  $r_{10} = 100 \text{ km}$ . Môžeme si všimnúť, že zatiaľ čo v prípade  $r_s \ll r_0$  je koeficient pomerne malý, pre  $r_0 \ll r_s$  nadobúda obrovské hodnoty. Sonda totiž „vymetá“ priestor vyplnený obrovským množstvom malých telies.

Tab. 2: Závislosť parametra  $\alpha$  na  $B$ ,  $r_0$

$r_0$	$B = -2.1$	$B = -2.5$	$B = -2.9$
1 km	7	19	70
100 m	11	60	600
10 m	17	200	5 000
1 m	30	1 500	110 000
1 dm	600	110 000	20 000 000
1 cm	60 000	30 000 000	1 600 000 000
1 mm	7 000 000	9 000 000 000	12 000 000 000 000

<sup>54</sup>[https://en.wikipedia.org/wiki/List\\_of\\_trans-Neptunian\\_objects](https://en.wikipedia.org/wiki/List_of_trans-Neptunian_objects)

<sup>55</sup>[https://en.wikipedia.org/wiki/Trans-Neptunian\\_object#/media/File:TheTransneptunians\\_73AU.svg](https://en.wikipedia.org/wiki/Trans-Neptunian_object#/media/File:TheTransneptunians_73AU.svg)

<sup>56</sup><http://www.findplanetnine.com/>

<sup>57</sup>[https://en.wikipedia.org/wiki/Oort\\_cloud](https://en.wikipedia.org/wiki/Oort_cloud)



Čiastkovú pravdepodobnosť môžeme odhadnúť ako

$$P(R) = 50 \frac{r_{10}^2}{R^2} \alpha.$$

Pre parameter  $\frac{r_{10}^2}{R^2}$  dostávame postupne  $1,3 \cdot 10^{-13}$  pre hlavný pás,  $3 \cdot 10^{-15}$  pre Centurov a  $4 \cdot 10^{-15}$  pre transneptunické objekty. Teda hlavný pás je dominantný zdroj rizika.

Pre celkovú pravdepodobnosť zrážky s telesom väčším ako 1 m dostávame

$$P \doteq 4 \cdot 10^{-10} \text{ až } 4 \cdot 10^{-6}.$$

Môžeme vidieť, že inžinieri sa nemusia ničoho obávať, pokiaľ ide o zásah sondy objektom väčším ako meter pri voľnom prelete Slnecnou sústavou. Aby sa riziko pohybovalo na úrovni percent, musí mať koeficient  $\alpha$  hodnotu v ráde miliárd. To ale nastáva len pre milimetrové telesá v prípade nepriaznivejšieho koeficienta  $B$ .

Inou otázkou sú ale blízke prelety popri objektoch, ktoré majú prstence, keďže v nich je koncentrácia hmoty vyššia. Touto otázkou sa zaoberali vedci v prípade finálnej časti misie Cassini, ktorá prelietavala medzerou v prstencoch Saturna a kľúčová bola u sondy New Horizons pri prelete okolo Pluta, o ktorom sa dovtedy nevedelo, že prstence nemá. V dnešnej dobe sa rovnako problematickým stáva odletieť preč od Zeme, ktorú sme si obklopili množstvom satelitov, ale i odpadu.

## Úloha V.1 ... schodišťa na Měsíci

*Pokud bychom jednou kolonizovali Měsíc, bylo by vhodné na něm používat schody? Představte si na Měsíci klesající schodišťa s výškou schodu  $h = 15$  cm a délkou  $d = 25$  cm. Odhadněte počet schodů  $N$ , které by přeletěl člověk, jestliže by před vstupem na schody šel rychlostí  $v = 5,4 \text{ km} \cdot \text{h}^{-1} = 1,5 \text{ m} \cdot \text{s}^{-1}$ . Tíhové zrychlení na povrchu Měsíce je šestkrát slabší než na povrchu Země.*

Na Mesiaci pôsobí gravitačné zrýchlenie

$$a = \frac{g}{6},$$

kde  $g$  je pozemské tiažové zrýchlenie. Umiestnime počiatok súradnicovej sústavy na hranu najvyššieho schodu (koniec rovnej plochy). Nech os  $x$  je vodorovná v smere od roviny cez schodisko a os  $y$  je zvislá v smere zhora nadol. Potom pre letiaceho pozemšťana platí

$$\begin{aligned} x_p &= vt, \\ y_p &= \frac{at^2}{2}, \end{aligned}$$

pre vrcholy schodov

$$\begin{aligned} x_s &= Nd, \\ y_s &= Nh. \end{aligned}$$

V okamihu dopadu platí

$$Nd = x_s = x_p = vt,$$

$$Nh = y_s = y_p = \frac{at^2}{2},$$

z čoho dostávame

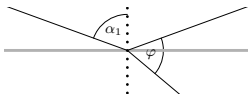
$$N = \frac{2hv^2}{ad^2}$$

a po dosadení

$$N = \frac{12hv^2}{gd^2}.$$

Po vyčíslení dostávame  $N \doteq 6,6$ , teda človek pretne priamku spájajúcu vrcholy schodov niekde medzi  $N = 6$  a  $N = 7$ . Čiže po preletení šiestich schodov dopadne na siedmy. Pre Zem by sme dostali, že preletíme jeden schod, čo je v súlade s realitou. Prvý schod totiž zvykneme buď preskočiť, alebo pred schodiskom pribrzdíme. Problematický je totiž práve prechod z pohybu vo vodorovnom smere do pohybu v šikmom smere, nie pohyb šikmo. Na Mesiaci by teda nemusel byť problém používať dlhšie schodisko, ak by pred ním používatelia pribrzdili. Krátke schodisko sa nevyplatí, pri skoku z výšky jedného poschodia nadobudneme rovnakú rýchlosť ako zo šesťkrát menšej výšky na Zemi, teda asi z pol metra. A to sa nám určite nič nestane.

## Úloha V.2 ... paprsky smrti na skle



Na skleněnou desku s absolutním indexem lomu  $n = 1,5$  dopadá světelný paprsek. Stanovte jeho úhel dopadu  $\alpha_1$ , jestliže paprsek odražený od rozhraní svírá s lomeným paprskem úhel  $60^\circ$ . Deska je uložena ve vzduchu.

Index lomu vzduchu je  $n_0 = 1$ , uhol lomu označíme  $\alpha_2$ . Zo Snellovho zákona vieme, že platí

$$n_0 \sin \alpha_1 = n \sin \alpha_2.$$

Zo zákona odrazu a z geometrie situácie vidíme, že

$$\alpha_1 + \alpha_2 + 60^\circ = 180^\circ.$$

Teda

$$\alpha_1 + \alpha_2 = 120^\circ.$$

Z tejto rovnosti vyjadríme  $\alpha_2$  a dosadíme do Snellovho zákona

$$\sin \alpha_1 = \frac{n}{n_0} \sin \alpha_2 = \frac{n}{n_0} \sin (120^\circ - \alpha_1).$$

Použijeme súčtový vzorec pre sínus rozdielu dvoch uhlov  $\varphi_1$  a  $\varphi_2$

$$\sin (\varphi_1 - \varphi_2) = \sin \varphi_1 \cos \varphi_2 - \cos \varphi_1 \sin \varphi_2.$$

Dostáváme

$$\sin \alpha_1 = \frac{n}{n_0} (\sin 120^\circ \cos \alpha_1 - \cos 120^\circ \sin \alpha_1) .$$

Úpravami vyjádříme uhol  $\alpha_1$

$$\begin{aligned} \sin \alpha_1 \left( 1 + \frac{n}{n_0} \cos 120^\circ \right) &= \frac{n}{n_0} \sin 120^\circ \cos \alpha_1 , \\ \operatorname{tg} \alpha_1 &= \frac{\frac{n}{n_0} \sin 120^\circ}{1 + \frac{n}{n_0} \cos 120^\circ} . \end{aligned}$$

Po dosazení číselných hodnot

$$\operatorname{tg} \alpha_1 = 3\sqrt{3} .$$

Teda

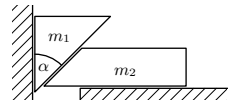
$$\alpha_1 \doteq 79^\circ .$$

Uhol dopadu lúča je  $79^\circ$ .

### Úloha V.3 ... klín

Máme dva klíny o hmotnostech  $m_1, m_2$  a úhel  $\alpha$  (viz obrázek). Vypočítejte zrychlení levého klínu. Předpokládejte, že nikde nedochází ke tření.

Bonus Uvažujte tření s koeficientem  $f$ .



Levý klín se může pohybovat pouze ve svislém směru a pravý klín po podložce ve vodorovném směru. Označme zrychlení levého klínu  $a_1$  a zrychlení pravého klínu  $a_2$ . Na levý klín působí ve svislém směru tíhová síla  $m_1g$  a v opačném směru složka  $F_n \sin \alpha$  reakční síly  $F_n$ , která působí mezi našimi klíny a je kolmá k jejich styčné ploše. Celkově tak z druhého Newtonova zákona dostáváme

$$m_1g - F_n \sin \alpha = m_1a_1 . \tag{18}$$

Na pravý klín působí ve vodorovném směru složka  $F_n \cos \alpha$  reakční síly, a proto platí

$$F_n \cos \alpha = m_2a_2 . \tag{19}$$

Dále využijeme vazbu mezi klíny. Představme si, že se levý klín posune svisle dolů o vzdálenost  $s_1$ . Protože na sebe i nadále klíny musí přiléhat, posune se pravý klín doprava o vzdálenost  $s_2 = s_1 \operatorname{tg} \alpha$ . Dvojitým zderivováním podle času získáme  $a_2 = a_1 \operatorname{tg} \alpha$ . Když tento vztah dosadíme do rovnice (19), dostaneme

$$F_n \cos \alpha = m_2a_1 \operatorname{tg} \alpha . \tag{20}$$

Nyní máme soustavu dvou rovnic o dvou neznámých. Rovnici (20) vynásobíme výrazem  $\operatorname{tg} \alpha$  a sečteme ji s rovnicí (18). Tak se zbavíme členu s neznámou silou  $F_nT$ ,

$$m_1g = m_1a_1 + m_2a_1 \operatorname{tg}^2 \alpha .$$

Odtud už snadno vyjádříme

$$a_1 = g \frac{1}{1 + \frac{m_2}{m_1} \operatorname{tg}^2 \alpha}.$$

Jestliže se tedy levý klín bude pohybovat směrem dolů bez tření, jeho výsledné zrychlení bude  $a_1$ .

### Bonus

Kromě sil zmíněných výše působí na klíny ještě třecí síly s koeficientem tření  $f$ . V případě levého klínu se budeme opět zajímat o velikost svislé komponenty výsledné síly. Připomeňme, že na levý klín působí ve svislém směru tíhová síla  $m_1 g$  směrem dolů a síla  $F_n \sin \alpha$  směrem nahoru. Levý klín působí na pravý klín silou  $F_n$ , která je kolmá na jejich styčnou plochu. Tření s pravým klínem způsobí třecí sílu  $f F_n$ , která míří ve směru styčné plochy vzhůru a má svislou komponentu  $f F_n \cos \alpha$ . Celková síla, kterou působí levý klín na svislou stěnu, je součtem síly  $F_n \cos \alpha$  a vodorovné komponenty třecí síly  $-f F_n \sin \alpha$ . Třecí sílu mezi levým klínem a svislou stěnou pak spočítáme jako  $f(F_n \cos \alpha - f F_n \sin \alpha)$ . Celkově tak z druhého Newtonova zákona dostáváme

$$m_1 g - F_n \sin \alpha - f F_n \cos \alpha - f(F_n \cos \alpha - f F_n \sin \alpha) = m_1 a_1.$$

Na pravý klín působí normálová síla  $F_n$  a třecí síla  $f F_n$ . Jejich vodorovné komponenty jsou  $F_n \cos \alpha$  a  $-f F_n \sin \alpha$ . Dále potřebujeme zjistit velikost výsledné normálové síly působící na podložku, abychom z ní určili třecí sílu mezi podložkou a pravým klínem. Tato normálová síla se skládá ze svislé složky  $F_n \sin \alpha$  síly  $F_n$ , ze svislé složky  $f F_n \cos \alpha$  třecí síly mezi oběma klíny a z tíhové síly  $m_2 g$ . Třecí síla od vodorovné stěny je pak  $f(F_n \sin \alpha + f F_n \cos \alpha + m_2 g)$ . Z druhého Newtonova zákona pro pohyb pravého klínu v horizontálním směru platí

$$F_n \cos \alpha - f F_n \sin \alpha - f(F_n \sin \alpha + f F_n \cos \alpha + m_2 g) = m_2 a_1 \operatorname{tg} \alpha,$$

kde jsme již dosadili za  $a_2$  z rovnosti  $a_2 = a_1 \operatorname{tg} \alpha$ , protože pohyb klínů je na sebe navázaný stejně jako v případě bez tření. Vzniklou soustavu převedeme vhodnými algebraickými úpravami do tvaru

$$\begin{aligned} m_1 g - F_n (\sin \alpha + 2f \cos \alpha - f^2 \sin \alpha) &= m_1 a_1, \\ -f m_2 g + F_n (\cos \alpha - 2f \sin \alpha - f^2 \cos \alpha) &= m_2 a_1 \operatorname{tg} \alpha. \end{aligned}$$

Obě rovnice vhodně přenásobíme a sečteme

$$\begin{aligned} m_1 g (\cos \alpha - 2f \sin \alpha - f^2 \cos \alpha) - f m_2 g (\sin \alpha + 2f \cos \alpha - f^2 \sin \alpha), \\ = a_1 (m_1 (\cos \alpha - 2f \sin \alpha - f^2 \cos \alpha) + m_2 \operatorname{tg} \alpha (\sin \alpha + 2f \cos \alpha - f^2 \sin \alpha)). \end{aligned}$$

Dalšími algebraickými úpravami najdeme řešení

$$a_1 = g \frac{1 - f^2 - 2f \operatorname{tg} \alpha - \frac{m_2}{m_1} f (2f + (1 - f^2) \operatorname{tg} \alpha)}{1 - f^2 - 2f \operatorname{tg} \alpha + \frac{m_2}{m_1} \operatorname{tg} \alpha (2f + (1 - f^2) \operatorname{tg} \alpha)}.$$

Všimněme si, že dosazením  $f = 0$  dostaneme stejný vzorec jako v základní úloze.

## Úloha V.4 ... tepelné ztráty

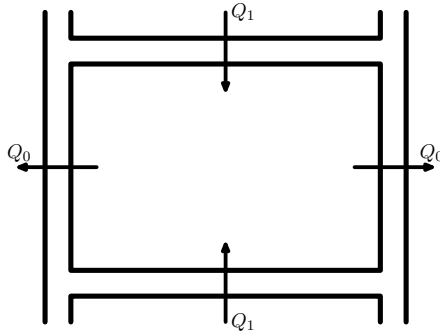
Na jaké teplotě se ustálí vnitřní prostředí bytu v panelovém domě? Uvažujte, že náš byt sousedí delšími stěnami, stropem a podlahou s dalšími byty, ve kterých je udržována teplota  $22^\circ\text{C}$ . Kratšími stěnami sousedí s okolím, kde je teplota  $-5^\circ\text{C}$ . Vnitřní rozměry bytu jsou: výška  $h = 2,5\text{ m}$ , šířka  $a = 6\text{ m}$  a délka  $b = 10\text{ m}$ . Součinitel měrné teplotní vodivosti stěn je  $\lambda = 0,75\text{ W}\cdot\text{K}^{-1}\cdot\text{m}^{-1}$ . Vnější stěny a stropy jsou tlusté  $D_{\text{out}} = 20\text{ cm}$  a vnitřní  $D_{\text{in}} = 10\text{ cm}$ .

Jak se změní výsledek, pokud budovu zvenku zateplíme polystyrenem o tloušťce  $d = 5\text{ cm}$  s měrnou tepelnou vodivostí  $\lambda' = 0,04\text{ W}\cdot\text{K}^{-1}\cdot\text{m}^{-1}$ ?

Úloha je zameraná na vedenie tepla. Základná rovnica, ktorú využijeme, je

$$Q = \lambda S \frac{T_1 - T_2}{d} t, \quad (21)$$

kde  $\lambda$  je koeficient tepelnej vodivosti,  $S$  je plocha steny,  $T_1 - T_2$  je rozdiel teplôt a  $t$  je doba tepelnej výmeny. Konštrukcia je na obrázku 19.



Obr. 19: Pohľad na byt zhora.

Označme  $Q_0$  teplo prenesené do bytu cez vonkajšiu stenu. Ďalej  $Q_1$  sú tepelné prenosy cez steny spoločnými so susednými bytmi a  $Q_2$  sú tepelné prenosy cez podlahu a strop. Po ustálení situácie musí platiť, že za ľubovoľný časový úsek  $t$  bude celkový prírastok energie bytu nulový, teda

$$2Q_0 + 2Q_1 + 2Q_2 = 0,$$

čiže po dosazení z (21) dostávame

$$2\lambda ah \frac{T_{\text{out}} - T}{D_{\text{out}}} t + 2\lambda bh \frac{T_{\text{in}} - T}{D_{\text{in}}} t + 2\lambda ab \frac{T_{\text{in}} - T}{D_{\text{out}}} t = 0,$$

kde  $T_{\text{out}}$  je teplota vonku a  $T_{\text{in}}$  je teplota v okolitých bytoch. Odtiaľ si už môžeme vyjadriť teplotu, na ktorej sa ustáli prostredie vo vnútri bytu

$$T = \frac{ahD_{\text{in}}T_{\text{out}} + bhD_{\text{out}}T_{\text{in}} + abD_{\text{in}}T_{\text{in}}}{ahD_{\text{in}} + bhD_{\text{out}} + abD_{\text{in}}} \doteq 291,91\text{ K} = 18,76^\circ\text{C}.$$

### Pre zateplené

Zateplením vonkajšej steny sa zmení len hodnota  $Q_0$ . Konkrétne sa zmení hodnota  $R$ , čo je tepelný odpor, pre ktorý platí  $R = \frac{d}{\lambda}$ . Pôvodná rovnica má potom tvar

$$Q = S \frac{T_1 - T_2}{R}.$$

Po zateplení sa výsledný tepelný odpor rovná súčtu<sup>58</sup> tepelných odporov polystyrénu a steny, teda

$$R = \frac{D_{\text{out}}}{\lambda} + \frac{d}{\lambda'}.$$

Čiže dostávame vzťah

$$2ah \frac{T_{\text{out}} - T}{\frac{D_{\text{out}}}{\lambda} + \frac{d}{\lambda'}} t + 2\lambda bh \frac{T_{\text{in}} - T}{D_{\text{in}}} t + 2\lambda ab \frac{T_{\text{in}} - T}{D_{\text{out}}} t = 0,$$

z ktorého si opäť môžeme vyjadriť teplotu, na ktorej sa ustáli prostredie vo vnútri bytu

$$T = \frac{\lambda' ah \frac{T_{\text{out}}}{\lambda' D_{\text{out}} + \lambda d} + bh \frac{T_{\text{in}}}{D_{\text{in}}} + ab \frac{T_{\text{in}}}{D_{\text{out}}}}{\lambda' ah \frac{1}{\lambda' D_{\text{out}} + \lambda d} + bh \frac{1}{D_{\text{in}}} + ab \frac{1}{D_{\text{out}}}} \doteq 294,52 \text{ K} = 21,37^\circ \text{ C}.$$

Vidíme, že teplota v byte už bude veľmi podobná teplote, na ktorú vykurejú susedi svoje byty.

## Úloha V.5 . . . záludná kapka

Mějme kulatou kapku o poloměru  $r_0$  tvořenou vodou o hustotě  $\rho_v$ , která shodou okolností padá v mlze v homogenním tíhovém poli  $g$ . Uvažujme vhodnou mlhu se speciálními předpoklady. Tvoří ji vzduch o hustotě  $\rho_{\text{vd}}$  a vodní kapičky s průměrnou hustotou  $\rho_r$ , když uvážíme, že se rozptýlí zcela rovnoměrně. Jestliže kapka propadne nějakým objemem takové mlhy, vysbírá všechnu vodu, která se v tomto objemu nachází. Na místě zůstane pouze vzduch. Jaká je závislost hmotnosti kapky na vzdálenosti uražené v takovéto mlze?

Bonus Řešte pohybové rovnice.

Při řešení bonusu vyjdeme z pohybové rovnice  $F = \dot{p} = \dot{m}\dot{x} + m\ddot{x}$ . Na kapku působí jediná síla, a sice tíhová,<sup>59</sup> kterou spočítáme jako  $F = mg$ . Dále můžeme spočítat časovou derivaci hmotnosti

$$\dot{m} = km^{\frac{2}{3}} \dot{x}.$$

<sup>58</sup>V praxi sa pri výpočte tepelného odporu zohľadňujú aj odpory pri prestupe tepla na vnútornom a vonkajšom povrchu konštrukcie.

<sup>59</sup>Zanedbávame vztlakovou sílu a odpor vzduchu. Pokud bychom je chtěli uvažovat, stačilo by pouze změnit některé konstanty, ale rovnice jako takové by zůstaly stejné. Vztlakovou sílu totiž přidáme členem úměrným pouze objemu, ten je zase přímo úměrný hmotnosti. Člen  $mg$  už je ale obsažen díky tíhové síle. Podle Newtonova zákona odporu bychom zase museli brát v úvahu nový člen úměrný druhé mocnině rychlosti, ten je zase shodný se členem, který získáme z časové derivace hybnosti.

Pro pohyb kapky tak platí

$$mg = m\dot{x} + m\ddot{x} = km^{\frac{2}{3}}\dot{x}^2 + m\ddot{x},$$

$$g = \frac{k}{\frac{1}{3}kx + \left(\frac{4}{3}\pi\rho_v\right)^{\frac{1}{3}}r_0}\dot{x}^2 + \ddot{x}.$$

Jednoduchými algebraickými úpravami a zjednodušením<sup>60</sup>  $x_0 = \frac{3}{k} \left(\frac{4}{3}\pi\rho_v\right)^{\frac{1}{3}} r_0 = \frac{4g_v}{g_r} r_0$  dostáváme

$$g(x + x_0) = 3\dot{x}^2 + \ddot{x}(x + x_0) \quad (22)$$

Nyní využijeme substituci  $x = y^2 - x_0$ , pro kterou platí

$$\dot{x} = 2y\dot{y},$$

$$\ddot{x} = 2\dot{y}^2 + 2y\ddot{y}.$$

Snadno si můžete ověřit, že dosazení do rovnice (22) vede na

$$g = 14\dot{y}^2 + 2y\ddot{y}.$$

Nyní už konečně využijeme rovnost

$$\ddot{y} = \frac{d\dot{y}}{dt} = \frac{d\dot{y}}{dt} \frac{dy}{dy} = \frac{d\dot{y}}{dt} \frac{dy}{dy} = \dot{y} \frac{d\dot{y}}{dy},$$

která nám umožní předchozí diferenciální rovnici separovat. Řešíme integrál

$$\int \frac{dy}{y} = \int \frac{2y\dot{y}}{g - 14\dot{y}^2},$$

$$\ln y = -\frac{1}{14} \int \frac{-28y\dot{y}}{g - 14\dot{y}^2} = -\frac{1}{14} \ln |g - 14\dot{y}^2| + C.$$

Dosazením z předchozích vztahů ukážeme, že

$$g - 14\dot{y}^2 = \frac{1}{6} (7\ddot{x} - g),$$

tedy že podmínka  $g > 14\dot{y}^2$  je ekvivalentní s podmínkou  $7\ddot{x} > g$ . Jelikož na počátku platí  $\ddot{x} = g$ , podmínka je (alespoň na začátku) zřejmě splněna. Později však budeme muset ověřit, zda platí po celou dobu.

Z počátečních podmínek si vyjádříme integrační konstantu

$$C = \frac{1}{14} \ln(gx_0^7),$$

<sup>60</sup>Ze vzorce (??) vyplývá, že pokud otočíme směr času, kapka se postupně dostane do jednoho konkrétního bodu, ve kterém je její poloměr nulový. Můžeme si povšimnout, že hodnota  $x_0$  má význam vzdálenosti tohoto bodu od polohy kapky s poloměrem  $r_0$ .

tedy po zpětném dosazení do výsledku integrálu máme

$$\begin{aligned}\ln y^{14} &= -\ln(g - 14\dot{y}^2) + \ln(gx_0^7), \\ \dot{x} &= \sqrt{\frac{2g}{7}} \left( (x + x_0) - x_0^7 (x + x_0)^{-6} \right)^{\frac{1}{2}}, \\ \ddot{x} &= \frac{g}{7} \left( 1 + 6x_0^7 (x + x_0)^{-7} \right).\end{aligned}$$

Druhou časovou derivací jsme získali  $\ddot{x}$ , pomocí čehož jsme si dosazením do (22) ověřili, že jsme postupovali správně. Navíc zřejmě platí  $7\ddot{x} > g$ , takže podmínka z předchozí části řešení bude splněna vždy.

Další postup je jasný, předchozí rovnice pro  $\dot{x}$  je separovatelná diferenciální rovnice, kterou stačí pouze zintegrovat

$$t = \sqrt{\frac{7}{2g}} \int \left( (x + x_0) - x_0^7 (x + x_0)^{-6} \right)^{-\frac{1}{2}} dx.$$

Bohužel, tato lahůdka jen pro ty největší integrální gurmány nemá analytické řešení. Lze ho vyjádřit pomocí hypergeometrické funkce, ale to je jen nekonečná suma, kterou stejně musíme spočítat numericky. Nicméně si můžeme povšimnout, že pro dostatečně velké  $x$  jde výraz  $x_0^7 (x + x_0)^{-7}$  velmi rychle k nule, tedy zrychlení se shora blíží hodnotě  $\frac{g}{7}$ . Zvolme nějaký dostatečně velký čas  $t_0$ , potom můžeme pro čas  $t > t_0$  pohyb kapky aproximovat rovnoměrně zrychleným pohybem

$$\begin{aligned}\ddot{x}(t) &\approx \frac{g}{7}, \\ \dot{x}(t) &\approx \frac{gt}{7} + \dot{x}(t_0), \\ x(t) &\approx \frac{gt^2}{14} + \dot{x}(t_0)t + x(t_0),\end{aligned}$$

kde  $x(t_0)$  a  $\dot{x}(t_0)$  jsou dráha a rychlost kapky v čase  $t_0$ . Jejich hodnoty musíme samozřejmě určit numericky, ale všechny další  $x(t)$  už pak z předchozích rovnic snadno spočítáme analyticky. Jedná se sice pouze o aproximaci, ale pro dostatečně velké  $t_0$  můžeme dosáhnout libovolné přesnosti. Navíc, vzhledem k tomu, jak vysoké mocniny se v původních rovnicích vyskytují, dostáváme i pro relativně malé hodnoty  $t_0$  velmi přesné výsledky.

## Úloha V.P ... plovoucí rtuť

Vymyslete co nejvíce fyzikálních „figlů“, díky kterým by rtuť, alespoň po omezenou dobu, plavala na kapalně vodě. Čím trvalejší řešení naleznete, tím lépe.

Nejprve uvedeme několik zajímavých konstant. Podle publikace Matematické, fyzikální a chemické tabulky Státního nakladatelství v Praze (ano, ty stařícké tabulky, které většina z vás má doma) je hustota rtuti  $\rho_{\text{Hg}} = 13\,546 \text{ kg}\cdot\text{m}^{-3}$ , hustota vody  $\rho_{\text{Aq}} = 998 \text{ kg}\cdot\text{m}^{-3}$ , teplota tání rtuti  $t_t = -38,8^\circ\text{C}$  a teplota varu rtuti  $t_v = 356,6^\circ\text{C}$ , součinitel teplotní objemové roztažnosti rtuti při  $20^\circ\text{C}$



$\beta = 0,18 \cdot 10^{-3} \text{ K}^{-1}$ , povrchové napětí rtuti při  $20^\circ\text{C}$   $\sigma_{\text{Hg}} = 491 \cdot 10^{-3} \text{ N}\cdot\text{m}^{-1}$ , povrchové napětí vody při  $20^\circ\text{C}$   $\sigma_{\text{Aq}} = 73 \cdot 10^{-3} \text{ N}\cdot\text{m}^{-1}$ .

Zadaný problém rozebereme z několika úhlů pohledu. Snížení hustoty rtuti tak, aby plavala na vodě, je nejpřímochařejší řešení, které napadne snad každého. To ovšem není nijak jednoduché. Mohli bychom rtuť dát do (vhodně velké, přesné parametry necháme na čtenáři) nádoby se vzduchem a ten potom všechen odsát, aby vznikl podtlak. Rtuť by pak izotermicky vyplnila nádobu a tím pádem by jí klesla hustota. Po návratu do prostředí s běžným tlakem by se ale hustota velmi rychle vrátila zpět na běžnou hodnotu, ani bychom ji na vodu nestihli nalít. Ekvivalent se stlačením vody by nefungoval ze stejného důvodu, pokud by něco takového bylo vůbec možné. Stejně tak se můžeme pokusit izobaricky změnit hustotu tím, že ji ohřejeme. Může se nám povést ji přivést na požadovanou hustotu dříve, než začne vařit a vypaří se? Tentokrát precizně ověříme, že se nám to povést nemůže, v ostatních ukázkách takto precizní nebudeme a pouze nastíníme postup.

Součinitel teplotní objemové roztažnosti se při malých změnách teploty chová skoro konstantně. Uvažujme tedy, že se nemění. Zhruba odhadneme potřebnou teplotu zdola. Aby rtuť měla menší hustotu než voda, potřebujeme její objem při dané hmotnosti zvětšit více než třináctkrát. Je-li  $\delta t$  rozdíl teplot, potom se počáteční objem  $V_0$  změní na  $V_1 = V_0(1 + \beta\delta t)$ . Aby se takto objem rtuti zvětšil více než třináctkrát, musí určitě platit  $\beta\delta t > 12$ , tedy  $\delta t > 12/\beta > 66\,000 \text{ K}$ , což je vysoko nad teplotou varu rtuti, i kdyby počáteční teplota soustavy byla  $0^\circ\text{C}$ .

Člověka také napadne rtuť nebo vodu s něčím smíchat (např. vodu osolit a tím jí zvýšit hustotu), ale to bohužel nespĺňuje parametry zadání (pak už to není voda, ale nějaká směs).

Přímochařým řešením je rtuť hodně ochladit. Ona pak kolem sebe vytvoří led, na kterém chvíli poplave. Na podobném principu funguje řešení rtuť rozžhavit. Díky vodním párám, které se okolo začnou tvořit, se chvíli udrží na hladině. Je diskutabilní, zda toto řešení splňuje zadání. Podle nás spíše nespĺňuje, ale nějaké body jsme přesto dávali, protože tento způsob napadl mnoho z vás.

Další hodně přímochařé řešení je rtuť zmrazit třeba ve tvaru lodičky. Rtuť pak bude plavat, dokud neroztaje. Abychom dodrželi zadání (vizte předchozí odstavec), musíme vodu pod lodičku ohřívát, což trvanlivost lodičky značně snižuje.

Jiné řešení je soustavu dát do skleněné nádoby ve tvaru úzkého válce, přímo kapiláry, který bude tak tenký, že rtuť se díky svému odstávání od stěn nádoby prostě nestlačí natolik, aby se potopila (tzn. aby kolem ní pronikla voda, která sice vzlíná, ale ani to jí nepomůže). Zároveň ale nádoba musí být tak široká, abychom rtuť ještě dostali dovnitř. Díky vazbám mezi atomy vody (vzlínavá vrstva musí mít nějakou tloušťku a snaží se být souvislá) tyto dvě hranice nejsou stejné, ale budeme muset měřit hodně přesně.

Další už ne tak zjevný postup je rtuť roztržít na malíčké kuličky, které se díky povrchovému napětí vody udrží na hladině. Toto řešení může být trvalé, pokud hladina vody vydrží v klidu a kuličky se k sobě nebudou přibližovat. Každá kulička kolem sebe totiž vytvoří prohlubeň, které způsobí, že pokud se kuličky dostanou moc blízko k sobě, seběhnou se, spojí a vytvoří tak větší kuličku. Příliš velké kuličky se kvůli své hmotnosti na hladině neudrží a potopí se. Velikosti kuliček lze

určit jednoduchým výpočtem, který je cvičením na vzájemné působení sil a který ponecháme laskavému čtenáři. Hlavní myšlenka je ale poměrně jednoduchá. Vazby mezi atomy rtuti jsou silnější než vazby mezi atomy vody (rtuť díky tomu bude víceméně držet tvar kuličky) a nanočástice se zřejmě nepotopí (povrchové napětí vody je silnější, to vidíme třeba na tom, že na hladině může plavat ocelová jehla).

Další možností je rtuti stále pomáhat, aby se udržela na hladině. Třeba ji mechanicky šoupat po hladině (např. do ní z boku foukat) nebo ji shora vysávat. Nebo zařídít, aby kapalina proudila tak rychle, že se rtuť nemá šanci potopit.

Můžeme také pod kapkou rtuti vytvořit trysku, která bude nahoru stále posílat proud vody a kapka, pokud z něj nesklouzne a bude dost malá, aby se nerozpadla, se na něm udrží. Tady řešíme ten problém, že kapička je jednak křehká, jednak je tato konfigurace hodně nestabilní a stačí malý podnět, aby objekt z fontánky sklouzl. Částečným řešením může být zmrazení kapičky. Bude-li proud vody dost silný, nebude se ani tvořit led (kvůli spoustě kinetické energie). Také můžeme kapičku na místě nějak přidržovat.

Relativní magnetická permeabilita rtuti je za normálních podmínek nižší než u vody (rtuť má něco jako  $1 - 31 \cdot 10^{-6}$ , voda  $1 - 9 \cdot 10^{-6}$ ), takže ji magnetické pole odpuzuje více než vodu. Proto můžeme udržovat kapičku rtuti na hladině pomocí magnetu umístěného pod nádobou. Rozdíl relativních magnetických permeabilit je ale malíčký, takže by se muselo jednat o hodně silný magnet, aby to fungovalo.

Rtuť podchlazená na teplotu blízkou teplotě absolutní nuly je supravodivá, takže s tím lze také pracovat. Tato metoda má ale extrémní až nesplnitelné nároky na dodávání/odebírání tepla, má-li voda zůstat kapalná.

Při řešení této úlohy se objevilo ještě mnohem více nápadů, některé dokonce vypadaly proveditelně. Abychom to uzavřeli, figlů je spousta, ale většinou se jedná o výměnu mezi množstvím a energetickou náročností. Které řešení je nejlepší, to necháme na názoru každého z vás.

## Úloha VI.1 . . . asi se urazila

*Máme dva hmotné body o stejných hmotnostech  $m$  ve vzdálenosti  $d$  od sebe volně v prostoru bez žádných vnějších gravitačních sil. Jakou minimální rychlost ve směru spojnice obou bodů musíme udělit jednomu hmotnému bodu, aby se od sebe stále vzdalovaly?*

Úlohu nelze řešit tak, že zafixujeme jeden bod a druhému udělíme únikovou rychlost z gravitačního pole prvního bodu. Problém je v tom, že „zafixovaný“ hmotný bod se začne pohybovat směrem za vzdalujícím se, s čímž ale vzorec pro únikovou rychlost nepočítá.

Potenciální energie jednoho hmotného bodu v poli druhého bodu je<sup>61</sup>

$$E_p = -\frac{Gm^2}{d},$$

<sup>61</sup>Nulovou hladinu potenciální energie volíme v nekonečnu.

což je tedy počáteční energie soustavy. Pokud bychom nyní každému bodu udělili rychlost  $u$  směrem od druhého bodu, každý z nich by získal kinetickou energii  $E_k = \frac{1}{2}mu^2$ . Pro celkovou energii systému by potom platilo

$$E = E_p + 2E_k = -\frac{Gm^2}{d} + mu^2.$$

Pokud bychom zvolili jednu určitou hodnotu  $u$ , body by se začaly postupně vzdalovat až do nekonečna, čili jejich vzájemná potenciální energie by se blížila k nule. Zároveň by jejich rychlost, a tedy i jejich kinetická energie, klesala k nule. Díky pohybu do nekonečna je splněn požadavek na neustálé vzdalování se, díky poklesu rychlosti až k nule je zároveň splněn požadavek minimální udělené rychlosti.

Celková energie se musí zachovávat. Jestliže mají body v nekonečnu dohromady nulovou energii, musely ji mít i na počátku, tedy po urychlení na rychlost  $u$ . Dostáváme tak

$$E = 0, \\ u = \sqrt{\frac{Gm}{d}}.$$

Nyní si jen stačí uvědomit, že udělit každému bodu rychlost  $u$  je stejné, jako udělit jednomu bodu rychlost  $v = 2u$ . Hledaná minimální rychlost, kterou musíme jednomu bodu udělit, aby se od sebe body navzády vzdalovaly, je

$$v = 2\sqrt{\frac{Gm}{d}}.$$

## Úloha VI.2 ... horký drát

Vypočítejte proud, který by měl procházet kovovým vláknem s průměrem  $d = 0,10\text{ mm}$  nacházejícím se ve vakuové baňce, aby teplota vlákna měla stálou hodnotu  $T = 2600\text{ K}$ . Předpokládejte, že povrch vlákna září jako ideální černé těleso. Zanedbejte ztráty tepla způsobené vedením tepla. Rezistivita materiálu vlákna při dané teplotě je  $\rho = 2,5 \cdot 10^{-4}\ \Omega\cdot\text{cm}$ .

Nápověda Použijte Stefanův-Boltzmannův zákon.

Ak sa má vlákno udržiavať na stálej teplote, je mu potrebné dodávať v každom okamihu výkon rovnako veľký ako výkon vláknom vyžarovaný. Vyžarovaný výkon  $P_{vyz}$  je priamo úmerný ploche povrchu vlákna  $S$  (uvažujeme valcové vlákno s dĺžkou  $l$ ) a intenzite vyžarovania  $M_0$

$$P_{vyz} = SM_0.$$

$M_0$  súvisí so Stefan-Boltzmannovou konštantou  $\sigma = 5,67 \cdot 10^{-8}\text{ W}\cdot\text{m}^{-2}\cdot\text{K}^{-4}$  a teplotou povrchu telesa  $T$  podľa Stefan-Boltzmannovho zákona

$$M_0 = \sigma T^4.$$

Vlákno teda vyžaruje výkon

$$P_{\text{vyz}} = \pi dl \sigma T^4 .$$

Výkon dodaný vláknu pretekajúcim prúdom  $I$  je úmerný odporu vlákna. Ten spočítame z rezistivity  $\varrho$ , plochy priečného prierezu  $S_p$  a dĺžky vlákna  $l$  ako

$$R = \frac{l\varrho}{S_p} = \frac{4l\varrho}{\pi d^2} .$$

Potom je výkon dodaný vláknu

$$P_{\text{dod}} = UI = RI^2 = \frac{4l\varrho}{\pi d^2} I^2 .$$

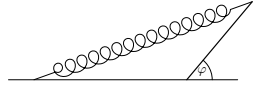
Z rovnosti  $P_{\text{vyz}} = P_{\text{dod}}$  vyjadríme prúd  $I$  a výraz upravíme

$$I = \frac{\pi T^2}{2} \sqrt{\frac{\sigma d^3}{\varrho}} \doteq 1,6 \text{ A} .$$

Vláknom teda musí prechádzať prúd s veľkosťou 1,6 A. V skutočnosti má na potrebný prúd vplyv aj vzájomné ožarovanie jednotlivých závitov vlákna žiarovky, ktoré tak zvyšuje teplotu vlákna v niektorých miestach. Náš výsledok je teda skôr horný odhad.

### Úloha VI.3 . . . neanalytická pružinka

Představme si tyč s délkou  $b = 5$  cm a hmotností  $m = 1$  kg a pružinku s klidovou délkou  $c = 10$  cm, s tuhostí  $k = 200 \text{ N} \cdot \text{m}^{-1}$  a se zanedbatelnou hmotností, které jsou na koncích spojeny. Druhé konce tyče a pružinky jsou upevněny ve stejné výšce ve vzdálenosti  $a = 10$  cm od sebe. Kolem obou upevnění i kolem spoje lze libovolně rotovat. Označme  $\varphi$  sklon tyče od horizontálního směru. Najděte všechny hodnoty  $\varphi$ , pro které je soustava v rovnovážné poloze. Které z těchto poloh jsou stabilní a které labilní?



Délku pružinky určíme z kosinusovej vety ako

$$d = \sqrt{a^2 + b^2 - 2ab \cos(\pi - \varphi)} = \sqrt{a^2 + b^2 + 2ab \cos \varphi}.$$

V rovnovážnej polohe musí byť splnená rovnováha momentov síl v bode uchytenia tyče. Tiažová sila tyče pôsobí momentom

$$M_1 = -mg \frac{b}{2} \cos \varphi,$$

kde znamienko  $-$  značí, že ide o moment spôsobujúci pohyb v smere hodinových ručičiek. Pre silu, ktorou pôsobí pružinka, máme

$$F_2 = k\Delta l = k(d - c),$$

teda pre moment dostávame

$$M_2 = F_2 b \sin \vartheta,$$

kde  $\vartheta$  je uhol medzi tyčou a pružinkou. Veľkosť sínusu tohto uhla môžeme pomocou sínusovej vety vyjadriť ako

$$\sin \vartheta = \frac{d}{a} \sin(\pi - \varphi) = \frac{d}{a} \sin \varphi.$$

Po dosadení máme pre moment

$$M_2 = k(d - c)b \frac{a}{d} \sin \varphi.$$

Pre rovnováhu musí byť výsledný moment síl  $M$  rovný nule, teda

$$0 = M = M_1 + M_2 = -mg \frac{b}{2} \cos \varphi + k(d - c)b \frac{a}{d} \sin \varphi.$$

Po úprave a dosadení za  $d$  máme rovnosť

$$mg \sqrt{a^2 + b^2 + 2ab \cos \varphi} \cos \varphi = 2k \left( \sqrt{a^2 + b^2 + 2ab \cos \varphi} - c \right) a \sin \varphi,$$

ktorá sa nedá riešiť analyticky, no napr. pomocou WolframAlpha alebo niektorej z metód diskutovanej v seriáli dostávame na intervale 0 až  $2\pi$  riešenia

$$\varphi_1 \doteq 0,69,$$

$$\varphi_2 \doteq 2,05,$$

$$\varphi_3 \doteq 2,86,$$

$$\varphi_4 \doteq 4,55.$$

Zostáva rozhodnúť, ktoré z riešení predstavujú stabilnú, a ktoré labilnú rovnováhu. Toto sa dá zistiť napríklad z hodnoty potenciálnej energie - stabilnej rovnováhe zodpovedajú lokálne minimá, labilnej lokálne maximá. Inou možnosťou je zistiť znamienko výsledného momentu síl po malom vychýlení z rovnovážnej polohy. V prípade stabilnej rovnováhy bude výsledný moment pôsobiť proti smeru výchylky. Keďže je výsledný moment spojitá veličina vzhľadom na uhol  $\varphi$ , stačí určiť jej znamienka medzi bodmi rovnováhy:

$$M(0 < \varphi < \varphi_1) < 0,$$

$$M(\varphi_1 < \varphi < \varphi_2) > 0,$$

$$M(\varphi_2 < \varphi < \varphi_3) < 0,$$

$$M(\varphi_3 < \varphi < \varphi_4) > 0,$$

$$M(\varphi_4 < \varphi < 2\pi) < 0.$$

Ukážme, že poloha 1 je labilná. Ak sa vychýlime do nižších hodnôt  $\varphi$  (v smere hodinových ručičiek), bude mať výsledný moment záporné znamienko, bude teda výchylku zväčšovať. Podobne pri výchylke do vyšších hodnôt bude výsledný moment s kladným znamienkom výchylku zväčšovať. Rovnako preskúmame polohy 2, 3, 4. Zistíme, že polohy 1 a 3 sú labilné a polohy 2 a 4 sú stabilné.

Úloha sa dala riešiť aj energetickým prístupom. Náš systém má jeden voľný parameter, uhol  $\varphi$ . Pre celkovú energiu systému máme

$$E = E_e + E_g,$$

kde  $E_e$  je elastická energia pružiny a  $E_g$  je potenciálna energia v tiažovom poli. Pre energie máme

$$E_e = \frac{k(\Delta l)^2}{2},$$

$$E_g = mgh,$$

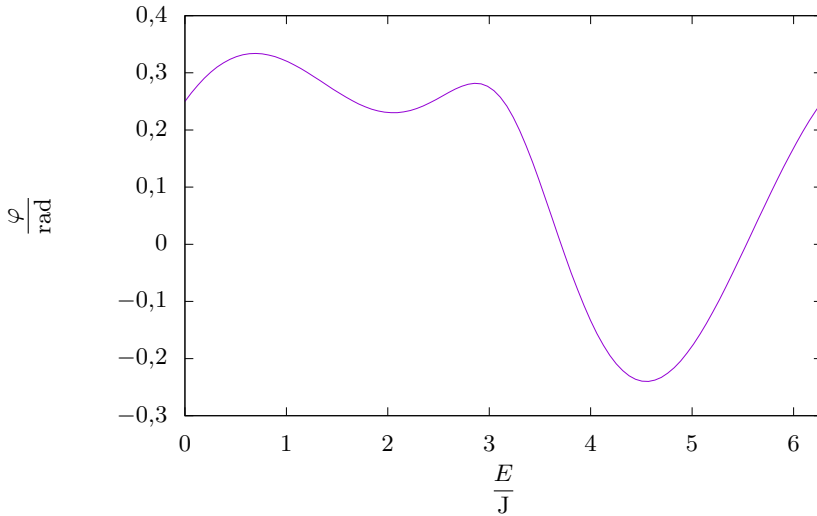
kde  $\Delta l$  je predĺženie pružiny a  $h$  je výška ťažiska tyče nad spojnicou bodov úchyto. Po dosadení dostávame

$$E = \frac{1}{2}k \left( \sqrt{a^2 + b^2 + 2ab \cos \varphi} - c \right)^2 + mg \frac{b}{2} \sin \varphi.$$

V rovnováze musí platit

$$\frac{\partial E}{\partial \varphi} = 0.$$

Po zderivování dostáváme rovnakú podmienku ako v prvom riešení prostredníctvom momentov. O stabilite sa môžeme presvedčiť hodnotou druhej derivácie v nulových bodoch prvej derivácie. Ak je druhá derivácia kladná, jedná sa o lokálne minimum energie, teda o stabilnú rovnovážnu polohu, a naopak. Najjednoduchší spôsob je však nechať si závislosť energie sústavy na uhle vykresliť, tak ako môžeme vidieť na obrázku 20.



Obr. 20: Závislosť celkovej energie systému  $E$  na uhle  $\varphi$ .

## Úloha VI.4 ... rozměrová analýza

Matěj si doma vyrobil střelnou zbraň a chce změřit, jakou rychlostí vystřeluje náboje. Bohužel nemá k dispozici jiný měřicí přístroj než pravítko. Našel ale kostku, jež je tvořena z poloviny ocelí a poloviny dřevem. Položí ji na kraj stolu (jehož výška je 100 cm a délka je 200 cm) a horizontálně do ní vystřelí. Kulka se od ocelové strany dokonale pružně odrazí přesně opačným směrem a dopadne do vzdálenosti 50 cm od stolu. Kostka se na stole posune o 5 cm. Potom Matěj kostku otočí a střelí do její dřevěné strany, v níž se kulka zaryje. Nyní naměřil posunutí jen 4 cm. Pomozte mu s výpočtem rychlosti výstřelu. Možná se vám bude hodit, že zjistil, že pohyb rozjeté kostky po stole se nezastaví, pokud jednu stranu stolu zvedne do výšky alespoň 20 cm.

Naměřené vzdálenosti si označíme po řadě (tak, jak jsou v zadání)  $h$ ,  $l$ ,  $d$ ,  $s_1$ ,  $s_2$  a  $\Delta h$ . Hmotnost kulky budeme značit  $m$ , hmotnost kostky  $M$  a třecí koeficient mezi kostkou a stolem  $f$ .

Okamžitou rychlost kostky po odrazu projektilu označme  $v_1$  a rychlost v okamžiku zarytí do dřevěné části označme  $v_2$ . Potom její kinetická energie bude  $\frac{1}{2}Mv_1^2$ , resp.  $\frac{1}{2}(M+m)v_2^2$ . Pohyb kostky se nezastaví, dokud všechnu energii neztratí vlivem smykového tření. Tíhová síla přitlačující kostku ke stolu je  $Mg$ , resp.  $(M+m)g$ , proto při posunu o vzdálenost  $s_1$ , resp.  $s_2$ , ztratí vlivem tření energii  $fMgs_1$ , resp.  $f(M+m)gs_2$ . Ze zákona zachování energie dostáváme

$$\begin{aligned}\frac{1}{2}Mv_1^2 &= fMgs_1 \quad \Rightarrow \quad v_1 = \sqrt{2fgs_1}, \\ \frac{1}{2}(M+m)v_2^2 &= f(M+m)gs_2 \quad \Rightarrow \quad v_2 = \sqrt{2fgs_2}.\end{aligned}$$

Okamžitou rychlost, kterou bude mít kulka hned po odrazu, označíme  $u$ . Potom řešíme horizontální vrh s počáteční rychlostí  $u$ . Ze znalosti výšky stolu a ze vzdálenosti, kam projektil dopadl, dostáváme

$$\begin{aligned}h &= \frac{1}{2}gt^2, \\ d &= ut, \\ u &= \sqrt{\frac{gd^2}{2h}},\end{aligned}$$

kde  $t$  je čas dopadu, který se nám z rovnic podařilo eliminovat.

Nyní použijeme zákon zachování hybnosti. Pro oba případy je počáteční hybnost celé soustavy  $mv$ . Po odrazu kulky má kostka hybnost  $Mv_1$  a kulka má hybnost  $-mu$ . Při pohlčení kulky nám zůstává hybnost  $(M+m)v_2$ . Dostáváme tedy jednu rovnici pro každý případ

$$\begin{aligned}mv &= Mv_1 - mu, \\ mv &= (M+m)v_2.\end{aligned}$$

Z obou rovnic vyjádříme poměr hmotností kostky a kulky, které následně eliminujeme

$$\begin{aligned}\frac{M}{m} &= \frac{v+u}{v_1}, \\ \frac{M}{m} &= \frac{v-v_2}{v_2}, \\ \frac{v+u}{v_1} &= \frac{v-v_2}{v_2}.\end{aligned}$$

Nyní už jen stačí vyjádřit původní rychlost projektilu. Všechny další rychlosti jsme spočítali výše, takže za ně můžeme rovnou dosadit

$$v = \frac{uv_2 + v_1v_2}{v_1 - v_2} = \frac{\sqrt{\frac{gd^2}{2h}}s_2 + \sqrt{2fgs_1}s_2}{\sqrt{s_1} - \sqrt{s_2}}.$$



Poslední neznámou je třecí koeficient  $f$ . Označme úhel, při kterém se kostka rozjede, jako  $\alpha$ . Kostku směrem dolů urychluje složka gravitační síly  $mg \sin \alpha$ , zatímco ji brzdí třecí síla  $fmg \cos \alpha$ . Obě síly se musí rovnat, tedy pro třecí koeficient vychází

$$f = \operatorname{tg} \alpha = \frac{\Delta h}{\sqrt{l^2 - \Delta h^2}}.$$

Dosazením do předchozího vztahu pro rychlost dostáváme

$$v = \frac{uv_2 + v_1v_2}{v_1 - v_2} = \frac{\sqrt{\frac{gd^2}{2h} s_2} + \sqrt{\frac{2gs_1 s_2 \Delta h}{\sqrt{l^2 - \Delta h^2}}}}{\sqrt{s_1} - \sqrt{s_2}}.$$

*Matěji, máme problém*

V předchozích výpočtech jsme však nikde nevyužili skutečnosti, že odraz má být dokonale pružný. To znamená, že při odrazu by se neměla ztratit žádná energie. Pokud si však zkusíme napsat zákon zachování energie pro odraz kulky

$$\frac{1}{2}mv^2 = \frac{1}{2}Mv_1^2 + \frac{1}{2}mu^2, \quad (23)$$

zjistíme, že jsme dostali další rovnici, ale žádnou novou neznámou. Všechny veličiny, které v této rovnici vystupují, už známe z předchozích výpočtů.<sup>62</sup> Můžeme tedy zkusit dosadit jejich skutečné hodnoty ze zadání a ověřit, zda rovnost platí. Oops! Rovnost neplatí. Co to znamená?

Neznamená to, že jsme počítali špatně. V zadání je však příliš mnoho údajů potřebných pro náš výpočet. V podstatě máme více rovnic než neznámých. Docházíme k závěru, že celé zadání je nekonzistentní, protože ve skutečnosti by nebylo možné tuto sadu hodnot naměřit.

To je také důvod, proč můžeme korektními výpočty dojít i k jiné hodnotě  $v$ . Stačí ignorovat libovolnou z výše uvedených rovnic a místo ní použít vztah (23) a dostaneme jiný výsledek.

## Úloha VI.5 . . . skok z letadla

Filip o hmotnosti 80 kg vyskočil z letadla, které je ve výšce  $h_1 = 500$  m nad zemí. Ve stejném okamžiku z druhého letadla skočila Danko o hmotnosti 50 kg, ale z výšky  $h_2 = 569$  m nad zemí. Předpokládejme, že oba mají stejný odporový koeficient  $C = 1,2$ , Filipova plocha příčného průřezu je  $S_F = 2,2 \text{ m}^2$  a Dančina je  $S_D = 1,5 \text{ m}^2$ . Hustota vzduchu  $\rho = 1,205 \text{ kg} \cdot \text{m}^{-3}$  se nemění s výškou. Za jakou dobu od výskoku bude Danko ve stejné výšce nad zemí jako Filip?

Nejdříve předpokládejme, že obě letadla jsou například vrtulníky a v okamžiku skoku stojí ve vzduchu se zanedbatelnou rychlostí vůči okolí. Potom bude na oba skokany působit odporová síla podle Newtonova zákona odporu

$$F_o = \frac{1}{2}CS\rho v^2.$$

<sup>62</sup>Hmotnosti  $M$  a  $m$  sice neznáme, ale stačí nám znát jejich poměr.

Označíme-li si dráhu, kterou urazila Danko (měřeno od letadla, ze kterého vyskočila) jako  $x_D$ , její rychlost potom bude  $\dot{x}_D$  (směrem dolů). Pro její zrychlení pak z druhého Newtonova zákona vyplývá

$$m_D \ddot{x}_D = F,$$

kde  $F$  je výslednice sil, které na ni působí. Nyní už jen stačí dosadit  $F = F_g - F_o$ , kde  $F_g = m_D g$ , abychom celkově dostali

$$\ddot{x}_D = g - \frac{C S_D \varrho}{2 m_D} \dot{x}_D^2. \quad (24)$$

Je vidět, že s postupně se zvyšující rychlostí bude klesat zrychlení. Rychlost tak bude postupně konvergovat k jisté terminální rychlosti  $v_D$ , pro kterou platí  $\ddot{x}_D = 0$ , tedy

$$v_D = \lim_{t \rightarrow \infty} \dot{x}_D(t) = \sqrt{\frac{2 m_D g}{C S_D \varrho}}.$$

Číselně pro terminální rychlosti Danky a Filipa dostaneme

$$\begin{aligned} v_D &\doteq 21,3 \text{ m} \cdot \text{s}^{-1}, \\ v_F &\doteq 22,2 \text{ m} \cdot \text{s}^{-1}. \end{aligned}$$

Vidíme, že  $v_D < v_F$ . Filip se tak bude v limitním případě  $t \rightarrow \infty$  pohybovat rychleji než Danko. Vzhledem k tomu, že Filip padá z menší výšky, by se dalo očekávat, že se s Dankou potká přesně ve chvíli jejího dopadu na zem, tedy ve výšce 0.

Zatím si však nemůžeme být jisti tím, co se děje před dosažením terminální rychlosti. Vraťme se tedy k rovnici (24). Dostáváme

$$\begin{aligned} \frac{d\dot{x}}{dt} &= g(1 - v^{-2} \dot{x}^2), \\ dt &= \frac{d\dot{x}}{g(1 - v^{-2} \dot{x}^2)}. \end{aligned}$$

Nyní jen stačí na obě strany rovnice dopsat znaménka pro integrály. Při jejich řešení rovnou použijeme substituci

$$y = \frac{\dot{x}}{v} \quad \Rightarrow \quad d\dot{x} = v dy,$$

která vede na

$$\begin{aligned} \int dt &= \int \frac{d\dot{x}}{g(1 - v^{-2} \dot{x}^2)} = \frac{v}{g} \int \frac{dy}{1 - y^2}, \\ t &= \frac{v}{g} \operatorname{argtgh} y + C = \frac{v}{g} \operatorname{argtgh} \left( \frac{\dot{x}}{v} \right) + C. \end{aligned}$$

Pro  $t = 0$  máme  $\dot{x} = 0$ , z čehož vyplývá  $C = 0$ . Celkově tak máme

$$\dot{x}(t) = v \operatorname{tgh} \left( \frac{gt}{v} \right). \quad (25)$$

Pro  $t \rightarrow \infty$  vychází  $\dot{x} \rightarrow v$ , což je očekávaná hodnota. Nyní bychom chtěli ukázat, že pro libovolný čas  $t$  platí  $\dot{x}_D(t) < \dot{x}_F(t)$ . Na první pohled však není zřejmé, jak tyto funkce porovnat, takže si pomůžeme jednoduchým trikem. Derivaci dostáváme

$$\ddot{x}(t) = g \left( 1 - \operatorname{tgh}^2 \left( \frac{gt}{v} \right) \right).$$

Jelikož je funkce  $\operatorname{tgh}$  ryze monotónní na celém  $\mathbb{R}$ , z nerovnosti  $v_D < v_F$  nutně plyne  $\ddot{x}_D(t) < \ddot{x}_F(t)$ . Obě rychlosti se navíc v nule rovnají, takže musí platit  $v_D(t) < v_F(t)$ .

Filip s Dankou se tedy potkají přesně ve chvíli, kdy Danka dopadne na zem (Filip už tam bude). V zadání úlohy je otázka, kdy přesně k tomu dojde. Mohli bychom samozřejmě pokračovat další integrací vztahu (25), ukážeme si však jiný postup. Vyjdeme z rovnice (24) a použijeme rovnost

$$\ddot{x} = \frac{d\dot{x}}{dt} \frac{dx}{d\dot{x}} = \frac{dx}{dt} \frac{d\dot{x}}{dx} = \dot{x} \frac{d\dot{x}}{dx}.$$

Dosazením do (24) dostáváme

$$\begin{aligned} \dot{x} \frac{d\dot{x}}{dx} &= g (1 - v^{-2} \dot{x}^2), \\ dx &= \frac{\dot{x} d\dot{x}}{g (1 - v^{-2} \dot{x}^2)}. \end{aligned}$$

Před obě strany rovnice dopíšeme integrály a využijeme faktu, že integrál z derivace funkce, dělené danou funkcí, je logaritmus z dané funkce, tedy

$$\int dx = \int \frac{\dot{x} d\dot{x}}{g (1 - v^{-2} \dot{x}^2)} = -\frac{v^2}{2g} \int \frac{-2v^{-2} \dot{x} d\dot{x}}{1 - v^{-2} \dot{x}^2} = -\frac{v^2}{2g} \ln (1 - v^{-2} \dot{x}^2) + C.$$

Pro  $\dot{x} = 0$  je  $x = 0$ , takže integrační konstanta je rovna nule. Dosazení za  $\dot{x}$  z (25) vede na

$$\begin{aligned} x &= -\frac{v^2}{2g} \ln \left( 1 - \operatorname{tgh}^2 \left( \frac{gt}{v} \right) \right) = \frac{v^2}{g} \ln \cosh \left( \frac{gt}{v} \right), \\ t &= \frac{v}{g} \operatorname{argcosh} \exp \left( \frac{gx}{v^2} \right). \end{aligned}$$

Při pádu až na zem se celková uražená dráha rovná počáteční výšce, ze které Danka s Filipem vyskočili. Dosazením  $h_2$  za  $x_D$  a  $h_1$  za  $x_F$  můžeme snadno zjistit, že

$$\begin{aligned} t_D &\doteq 28,3 \text{ s}, \\ t_F &\doteq 24,1 \text{ s}. \end{aligned}$$

Danka se tak s Filipem potká až na zemi, kam dopadne zhruba 4,2s po něm.

Dále bychom mohli rozebrat případ, ve kterém mají letadla (a tedy i Danka s Filipem) nějakou nenulovou rychlost. Pokud bude nenulová pouze vertikální složka rychlosti, stačí jen rovnici (25) opravit o integrační konstantu. Pokud by však byla kladná směrem nahoru, museli bychom ji pro první část pohybu (dokud se nezačnou Filip s Dankou pohybovat směrem dolů) znovu odvodit, protože rovnice (24) nebere v úvahu změnu znaménka. Ve výsledku bychom tak místo funkce tgh dostali funkci tg.

Pro jinou počáteční rychlost analytické řešení neexistuje. Pro konkrétní počáteční rychlosti však dokážeme z původní pohybové rovnice snadno spočítat numerické řešení.

## Úloha VI.P . . . kompenzace vesmírné expanze

*Podle současných pozorování a vesmírných modelů se zdá, že vesmír se rozpíná a rychlost rozpínání se zvyšuje. Co kdyby to tak nebylo? Co kdyby byl vesmír stále stejně velký, ale měnily by se fyzikální zákony/konstanty? Jak by se musely konstanty měnit, aby se nám zdálo, že se vesmír rozpíná, jak ukazují pozorování? Popište co nejvíce zákonů, které by se musely měnit.*

Na začiatku je potrebné uvedomiť si, ako dokážeme zo Zeme odmerať, že vesmír sa rozpína. Rozpínanie vesmíru objavil v roku 1930 americký astronóm Edwin Hubble. Hubble meral Dopplerov jav v spektrách vzdialených galaxií, aby určil ich rýchlosť voči nám. Ak vezmeme do úvahy predpoklad, že galaxie rovnomerne vypĺňajú statický vesmír, malo by platiť, že pri dostatočne veľkom počte meraní bude počet približujúcich sa galaxií a počet vzdalujúcich sa galaxií približne rovnaký. Hubble ale nameral niečo úplne iné. Galaxie, ktoré sa od nás vzdalovali tvorili výraznú väčšinu v ním skúmanej vzorke. Tento fakt mu vnukol myšlienku, že galaxie sa od seba vzdalujú preto, lebo sa zvyšuje ich vzájomná vzdialenosť, oproti ktorej sú ich rozmery zanedbateľné, a teda, že vesmír sa rozpína. Hubble neskôr experimentálne ukázal, že rýchlosť vzdalovania nejakej galaxie vo vzdialenosti  $R$  je rovná

$$v = HR,$$

kde  $H$  je Hubblova konštanta, ktorá udáva rýchlosť rozpínania vesmíru a jej prevrátená hodnota je dobrým odhadom pre vek vesmíru. Z toho vidíme, že Hubblova konštanta nie je úplne konštantná, ale jej veľkosť sa mení s časom. Vidíme teda, že meranie rýchlosti rozpínania vesmíru je vlastne meranie hodnoty konštanty úmernosti medzi vzdialenosťou nejakej galaxie a rýchlosťou, akou sa vzdaluje. Do roku 1930 si ale všetci uvedomovali, že niektoré objekty sa od nás vzdalujú a niektoré zase nie, no nikto tomu neprikladal váhu. Nie je teda ničím novým, že vo vesmíre existujú mechanizmy, ktoré spôsobujú pohyb objektov nejakým smerom. Poďme teda vymyslieť nejaký mechanizmus, ktorý zapríčiní, že sa budú od nás vzdialené galaxie vzdalovať, alebo aspoň, že my ich budeme pozorovať akoby sa vzdalovali. Ak by sme vymysleli nejakú silu, ktorá ťahá objekty smerom od nás tým silnejšie, čím sú ďalej, ale nijako inak ju nemôžeme pozorovať, malo by to hneď niekoľko problémov. Nedokázali by sme rozlíšiť tento stav od stavu, keď sa

vesmír rozpína, takže táto teória je z tohto hľadiska ekvivalentná teórii expanzie. Čo je ale horšie, ak by sa rýchlosť vzdalovania objektov zväčšovala lineárne so vzdialenosťou od nás, znamenalo by to, že sme vo vesmíre na určitom privilegovanom mieste, čo ale nedáva vôbec žiadny zmysel. A nakoniec by toto vysvetlenie nebolo v súlade so zadaním úlohy, nakoľko chceme nielen aby sa vesmír nerozpínal tak, ako to vysvetľuje súčasná fyzika, ale aj aby bol statický. To teda znamená, že rýchlosť vzdalovania vzdialených galaxií, ktorú pozorujeme, je len zdanlivá, a ony sa v skutočnosti nehýbu. Ako je to ale možné? Ako vieme, rýchlosť vzdalovania meriame tak, že meriame červený posun spektrálnych čiar v spektre daného objektu. Predpokladajme teda, že červený posun nie je spôsobený tým, že sa od nás galaxie pohybujú preč obrovskou rýchlosťou, ale tým, že sa vlnová dĺžka svetla zmení po ceste z veľkej diaľky. Alebo dokonca, že v čase, keď bolo vyžiarené svetlo z ďalekej galaxie, mala nejaká konkrétna spektrálna čiara inú vlnovú dĺžku ako má v našom čase. Potom keď my porovnávame vlnovú dĺžku pozorovanej čiary a vlnovú dĺžku tej istej čiary, ale vyžiarenej z pozemského alebo veľmi blízkeho zdroja v našom čase, tak vidíme, že majú rozličné hodnoty, a teda nameriame červený posun. Vlnová dĺžka svetla vo vákuu je úzko previazaná s jeho frekvenciou  $f$  vzťahom

$$\lambda = \frac{c}{f},$$

kde  $c$  je rýchlosť svetla. Pri vytváraní nášho nového modelu, samozrejme chceme, aby sa zachovalo to najfundamentálnejšie. čo vo fyzike máme, a to je zákon zachovania energie. Ak teda fotón vyžiarieme s energiou  $E$ , tak k nám dopadne tiež s energiou  $E$ , ak počas svojho letu nejakú časť energie nestratil. Energia fotónu je závislá od jeho frekvencie, vo vákuu to vieme ekvivalentne formulovať aj pomocou jeho vlnovej dĺžky či jeho relativistickej hmotnosti  $m$ . Potom platí

$$E = hf = h \frac{c}{\lambda} = mc^2.$$

Pre daný fotón sú všetky tieto veličiny konštantné. My však chceme, aby konštantné neboli, ale aby záviseli od času. Konkrétne, aby sa hodnota menila s narastajúcim vekom vesmíru. Ten je voči nami skúmaným časovým horizontom veľmi veľký, a teda by sa tieto veličiny javili ako konštanty, avšak pri časových škálach, na ktorých pozorujeme "rozpínanie vesmíru", by sa už tento efekt prejavil. Naším cieľom bude teraz určiť, ako sa musia meniť tieto veličiny/konštanty aby to zodpovedalo nami pozorovanému červenému posunu. Najprv sa teda zamyslíme, ktoré veličiny sa môžu meniť. Ak má platiť zákon zachovania energie, tak pre jeden konkrétny fotón musí platiť

$$mc^2 = \text{konst.}$$

Z toho vidíme, že ak by sa mala simultánne s tým, ako starne vesmír, meniť rýchlosť svetla, musela by sa meniť aj hmotnosť fotónu. Zo špeciálnej relativity ale vieme, že vek fotónu je vždy rovný nule, keďže sa pohybuje rýchlosťou svetla, a to bezohľadu na to, ako veľkú časť vesmíru preletí. Z jeho pohľadu by sa mu musela zmeniť hmotnosť o nenulovú hodnotu za nulový čas. Niečo také si veľmi dobre nevieme predstaviť a preto ponecháme myšlienku o meniacej sa rýchlosti svetla a hmotnosti

fotónu bokom a budeme ďalej uvažovať, že tieto dve veličiny zostanú konštantné. Keďže zadaním požadujeme zmenu vlnovej dĺžky fotónu v dôsledku zmeny veku vesmíru, spolu s ňou sa bude meniť aj frekvencia svetla, nakoľko rýchlosť svetla považujeme v každom čase za konštantu. Z toho vidíme že sa musí meniť Planckova konštanta, aby platil zákon zachovania energie. Vyjadríme si teda vlnovú dĺžku fotónu s konštantnou energiou  $E$  v závislosti od veku vesmíru  $T$  ako

$$\lambda(T) = \frac{h(T)c}{E}.$$

Označíme si súčasný vek vesmíru ako  $t$  a vek vesmíru, v ktorom bol vyžiarený náš skúmaný fotón  $T$ . Potom červený posun  $z$  spôsobený rozdielnosťou Planckovej konštanty spočítame ako

$$z = \frac{\lambda_T - \lambda_t}{\lambda_t} = \frac{\frac{h(T)c}{E} - \frac{h(t)c}{E}}{\frac{h(t)c}{E}} = \frac{h(T) - h(t)}{h(t)}.$$

Ak máme nejaký objekt, u ktorého nameriame červený posun  $z$ , je jeho rýchlosť v smere od nás  $v$  rovná

$$v = \frac{z^2 + 2z}{z^2 + 2z + 2}c.$$

Ak poznáme empirický Hubblov vzťah, je potom táto rýchlosť úmerná Hubblovej konštante a vzdialenosti galaxie, z ktorej k nám fotón doputoval. Túto vzdialenosť vieme zapísať ako rozdiel časov  $t$  a  $T$  vynásobený rýchlosťou svetla. Platí

$$v = \frac{z^2 + 2z}{z^2 + 2z + 2}c = H(t - T)c.$$

Po dosadení do výrazu pre  $z$  do tohto vzťahu a niekoľkých úpravách dostávame

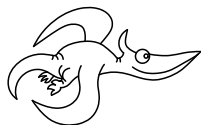
$$\frac{h^2(T) - h^2(t)}{h^2(T) + h^2(t)} = H(t - T)c.$$

Predpokladajúc, že Hubblova konštanta je prevrátenou hodnotou veku nášho vesmíru t.j.  $H = 1/t$  a prenásobením celej rovnice menovateľom dostávame

$$\begin{aligned} h^2(T) - h^2(t) &= (1 - TH)(h^2(T) + h^2(t)), \\ h^2(T)(1 - (1 - TH)) &= h^2(t)((1 - TH) + 1), \\ h(T) &= h(t)\sqrt{\frac{2 - TH}{TH}}. \end{aligned}$$

Pre konkrétnu hodnotu Planckovej konštanty ako ju poznáme dnes  $h(t)$  vieme teda dopočítať späť, aká bola jej hodnota v minulosti. Môžeme si všimnúť, že člen  $TH$  hrá vlastne úlohu "veku vesmíru v jednotkách súčasného veku vesmíru". Nakoľko tento vzorec bol odvodený z predpokladu, že k nám prichádza svetlo z minulosti, nie je možné z neho získať informáciu o hodnote Planckovej konštanty v budúcnosti. Vidíme teda, že existuje spôsob, ktorým sa dá vysvetliť pozorovaný

červený posun vzdialených objektov inak, ako rozpínaním vesmíru. Nakoľko je táto teória konzistentná s kvantovou fyzikou tu rozoberať nebudem, jedná sa skôr o prvý nástrel toho, ako inak by sa to dalo interpretovať, aby to nebolo úplne v rozpore so základmi fyziky. Toto riešenie ako autor považujem za úplné, to však neznamená, že nemôže existovať ďalšie, ktoré by túto problematiku rovnako dobre, alebo aj lepšie odôvodnilo.



## Zadání experimentálních úloh

### Úloha I.E ... pružnost špejle

12 bodů



Změřte průhyb špejle položené na jejích koncích v závislosti na síle působící na jejím středu (viz obrázek).

(řešení str. 106)

### Úloha II.E ... sypká

12 bodů

Změřte sypný úhel alespoň 2 látek běžně používaných v kuchyni (např. mouka, cukr, sůl apod.).

(řešení str. 111)

### Úloha III.E ... magneticky přitažlivá

12 bodů

Společně se zadáním této série jsme vám rozeslali poštou plošný magnet (magnetickou fólii). Tento magnet je trochu jiný než tyčové magnety – v ploše se střídavě střídají severní a jižní pól. Díky tomu se při přiblížení k feromagnetickému povrchu uzavře skrz kov „magnetický obvod“ a magnet drží (např. na ledničce) a unese na sobě třeba i obrázek. Vašimi úkoly jsou:

- Změřit plochu a tloušťku fólie, kterou využijete k experimentům.
- Změřit střední vzdálenost mezi dvěma nejbližšími stejnými magnetickými póly (dvojnásobek opačných).
- Změřit maximální užitečnou hmotnost (tedy hmotnost bez hmotnosti magnetu), kterou unese  $1\text{ cm}^2$  magnetu, je-li zatížení magnetu rovnoměrné, pokud magnet přichytíte zespoda k vodorovně umístěnému cca. 1 mm tlustému plechu z magneticky měkké oceli.

Nezapomeňte určit i chyby měření. Fólie, kterou jsme vám dodali, může být samolepící (je přes ni bílá fólie a pod ní lepidlo). V tom případě bílou fólii nahraďte něčím, na co budete upevňovat užitečnou hmotnost.

(řešení str. 116)

### Úloha IV.E ... tíha struny

12 bodů

Změřte délkovou hustotu struny, která vám měla přijít poštou společně se zadáním. Strunu ale nesmíte vážit.

*Nápověda* Zkuste strunu rozkmitat.

(řešení str. 119)

### Úloha V.E ... nezbedné fotony

12 bodů

Spolu se zadáním úlohy vám přišly polarizační brýle. Máte tedy 2 polarizační filtry. Když je dáte za sebe tak, aby směry jejich polarizace byly na sebe kolmé, nebude



skrz ně procházet téměř žádné světlo. Pokud ale mezi ně nyní vložíte třetí vhodně natočený filtr, můžete pozorovat, že bude procházet nemalé množství světla. Změřte závislost propustnosti na úhlu natočení prostředního filtru.

*Poznámka:* Jako první filtr a zároveň zdroj světla doporučujeme použít svítící displej. *(řešení str. 126)*

### **Úloha VI.E ... nehrajte si se sirkami**

12 bodů

Změřte rychlost hoření špejle v závislosti na úhlu naklonění vůči vodorovné rovině. *(řešení str. 130)*



## Řešení experimentálních úloh

### Úloha I.E ... pružnost špejle



Změřte průhyb špejle položené na jejích koncích v závislosti na síle působící na jejím středu (viz obrázek).

Pokud špejli o délce  $L$  podepřeme na okrajích a uprostřed zatížíme, špejle se prohne. Jedná se o deformaci tahem i tlakem, kdy se spodní vrstvy protahují a horní vrstvy se zkracují. Vrstva uprostřed zůstane stejně dlouhá. Na internetu<sup>1</sup> jsme našli vzorec, podle kterého pro průhyb špejle při vzdálenosti podpěr  $L$  zatížené uprostřed silou  $F$  platí

$$d = \frac{FL^3}{48EI_a},$$

kde  $E$  je Youngův modul pružnosti v tahu a  $I_a$  je kvadratický moment průřezu, v tomto případě kruhu. Spočítá se jako součet druhých mocnin vzdáleností všech bodů plochy průřezu od osy, na které zůstává délka materiálu stejná, což je v tomto případě vodorovná osa. Pokud si vzdálenost každého bodu od této osy označíme  $y$ , máme

$$I_a = \int_S y^2 dS.$$

Nyní přejdeme do polárních souřadnic. Pomocí vzdálenosti od středu  $r$  a úhlu od vodorovné roviny  $\varphi$  si vyjádříme  $y = r \sin \varphi$ . Element plochy  $dS$  můžeme spočítat jako  $dS = r d\varphi dr$ . Dále si celý integrál rozdělíme na integrál přes kružnici pro konstantní  $r$ , a na integrál těchto kružnic přes  $r$ . Poloměr celého kruhu si označíme  $R$  a využijeme skutečnosti, že integrál přes celý kruh se rovná dvojnásobku integrálu přes půlkruh

$$I_a = 2 \int_0^R \int_0^\pi (r \sin \varphi)^2 r d\varphi dr = 2 \int_0^R r^3 \int_0^\pi \sin^2 \varphi d\varphi.$$

Pomocí metody per-partes spočítáme

$$\int_0^\pi \sin^2 \varphi d\varphi = \sin \varphi \cos \varphi - \int_0^\pi \cos^2 \varphi = \sin \varphi \cos \varphi - \int d\varphi + \int \sin^2 \varphi d\varphi,$$

$$\int_0^\pi \sin^2 \varphi d\varphi = \frac{1}{2} [\varphi - \sin \varphi \cos \varphi]_0^\pi = \frac{\pi}{2}.$$

Dostáváme tak

$$I_a = \pi \int_0^R r^3 dr = \frac{\pi R^4}{4}.$$

<sup>1</sup>[http://physics.mff.cuni.cz/kfpp/skripta/kurz\\_fyziky\\_pro\\_DS/display.php/kontinuum/3\\_5](http://physics.mff.cuni.cz/kfpp/skripta/kurz_fyziky_pro_DS/display.php/kontinuum/3_5)

Sřed špejle jsme zatěžovali závažím o hmotnosti  $m$ , výsledný vzorec pro průhyb špejle je

$$d = \frac{mgL^3}{12E\pi R^4}.$$

### Postup

Měření jsme prováděli pro 5 různých špejlí, které jsme pro přehlednost barevně označili jako bílou, oranžovou, zelenou, žlutou a modrou. Podpěru špejlí jsme vytvořili z pojízdných šuplíků, mezi kterými jsme nastavovali mezeru o šířce postupně 25 cm, 20 cm, 15 cm a 10 cm. Přes ně jsme položili nezatíženou špejli, na kterou jsme nalepili milimetrový papír. Postupně jsme přes podpěry pokládali pokusné špejle, které jsme uprostřed zatěžovali lahví s vodou. Na milimetrovém papíru jsme pak odečetli průhyb špejle  $d$ . Pro každou špejli jsme naměřili prohnutí pro 11 až 30 hmotností, při kterých se špejle ještě nezlomila. Jelikož u kratších vzdáleností byly průhyby menší, mohli jsme použít těžší závaží. Nakonec jsme ještě pomocí posuvného měřítka změřili průměry špejlí.

### Výsledky

Měření závislosti průhybu  $d$  na hmotnosti závaží  $m$  jsme pro každou špejli a každou vzdálenost  $L$  vynesli do grafu, pro ilustraci zde uvádíme graf 21 s hodnotami pro bílou špejli pro  $L = 10$  cm. Vzdálenost podpěr jsme měřili pásmem, proto jeho chybu odhadujeme na 0,5 cm. Průhyb jsme měřili milimetrovým papírem, jehož nepřesnost odhadujeme na 0,5 mm. Všechny grafy jsme proložili přímkou s předpisem

$$f : d = Am,$$

kde  $A$  je konstanta úměrnosti, která představuje výraz

$$A = \frac{gL^3}{12E\pi R^4}.$$

Dále jsme do tabulky 3 vynesli výsledky měření průměru špejle, ze kterých jsme následně spočítali její poloměr včetně výběrové směrodatné odchylky průměru. Kombinací statistické chyby a chyby měřidla jsme určili celkovou nejistotu měření.

Do tabulky 4 jsme zapsali všechny vypočítané fitovací koeficienty včetně chyby fitu. Podle teorie by koeficient  $A$  měl záviset na třetí mocnině vzdálenosti podpěr  $L$ . Do grafů jsme tak vynesli závislost  $A$  na  $L^3$  (pro ilustraci zde uvádíme graf 22 s fitem pro bílou špejli) a hodnoty jsme proložili přímkou s předpisem

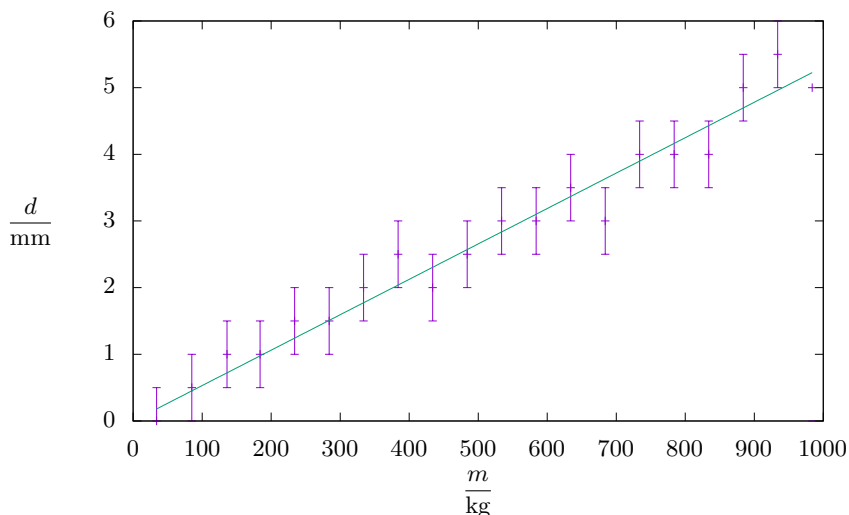
$$f : A = BL^3,$$

kde

$$B = \frac{g}{12E\pi R^4}.$$

Výsledné hodnoty koeficientu  $B$  i s chybami jsou v tabulce 5. Z nich jsme následně pro každou špejli spočítali Youngův modul pružnosti podle vzorce

$$E = \frac{g}{12\pi R^4 B}.$$



Obr. 21: Závislost průhybu bílé šejple na zatížení pro  $L = 10$  cm.

Chybu měření jsme určili klasicky z parciálních derivací

$$u_E = \sqrt{\left(\frac{\partial E}{\partial B} u_B\right)^2 + \left(\frac{\partial E}{\partial R} u_R\right)^2} B \sqrt{\left(\frac{u_B}{B}\right)^2 + \left(\frac{4u_R}{R}\right)^2}.$$

### Diskuse

Youngův modul pružnosti v ohybu nám pro jednotlivé šejple vyšel mezi 15 GPa a 23 GPa s přesností přibližně 4%, což je velmi přesné měření. Jako udávanou hodnotu modulu pružnosti v ohybu dřeva jsme na internetu <sup>2</sup> našli hodnoty mezi 9 GPa a 16 GPa, což je sice číselně méně než naše naměřené hodnoty, ale řádově se shodují. Vyšší hodnota námi naměřeného modulu pružnosti může být způsobena tím, že udávané hodnoty jsou pro velká prkna a trámy, kde se ve dřevě nacházejí suky a jiné nepravidelnosti, zatímco šejple je tenká a tudíž je celá z kvalitního dřeva. To, že nám pro každou šejpli vychází modul pružnosti jinak, je způsobeno tím, že dřevo je organický a navíc nepravidelný materiál. Proto i tabulkové hodnoty jsou pouze průměrné hodnoty pro jednotlivé druhy. Největší nepřesnosti našeho měření budou zřejmě způsobeny tím, že při malém průhybu se špatně odečítala jeho velikost na milimetrovém papíře, a dále také tím, že šejple se neprohýbala zcela rovnoměrně.

<sup>2</sup><https://www.drevostavitel.cz/clanek/mechanicke-vlastnosti-dreva-domacich-drevin>

Tab. 3: Měření průměru jednotlivých špejlí a výpočet jejich poloměru. Chyba jednoho měření je 0,025 mm.

	bílá	oranžová	modrá	žlutá	zelená
	$\frac{2R}{\text{mm}}$	$\frac{2R}{\text{mm}}$	$\frac{2R}{\text{mm}}$	$\frac{2R}{\text{mm}}$	$\frac{2R}{\text{mm}}$
1.	2,75	2,85	2,85	2,70	2,70
2.	2,70	2,80	2,80	2,70	2,70
3.	2,70	2,85	2,80	2,75	2,60
4.	2,75	2,80	2,75	2,80	2,60
5.	2,90	2,85	2,75	2,75	2,75
6.	2,75	2,80	2,85	2,85	2,70
7.	2,65	2,85	2,85	2,80	2,60
8.	2,70	2,80	2,80	2,85	2,70
9.	2,65	2,85	2,80	2,80	2,60
10.	2,80	2,80	2,85	2,75	2,65
$2R$	2,74	2,83	2,81	2,78	2,66
$\sigma_{2R}$	0,024	0,008	0,012	0,017	0,018
$R$	1,37	1,41	1,41	1,39	1,33
$u_R$	0,02	0,01	0,01	0,02	0,02

### Závěr

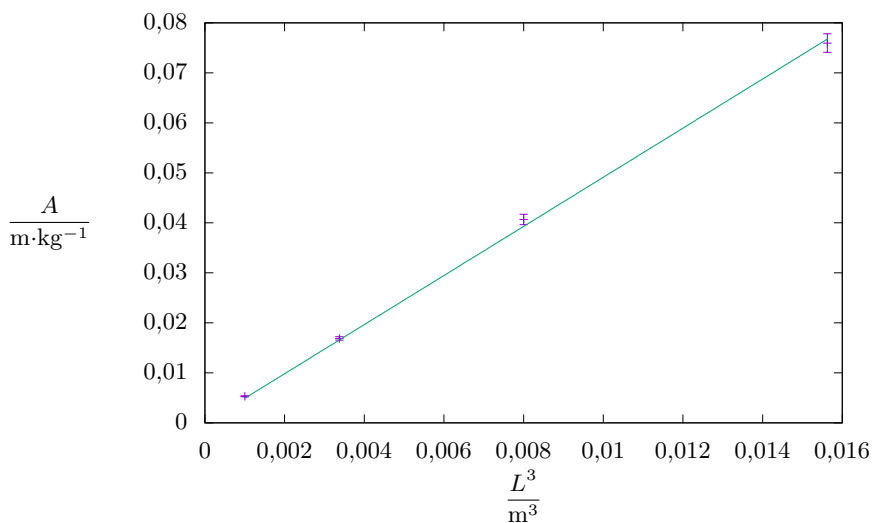
Podařilo se nám ověřit teoretický předpoklad, že závislost průhybu špejle na působící síle je lineární. Dále jsme spočítali Youngův modul pružnosti v ohybu jednotlivých špejlí, výsledky jsou v tabulce 5.

Tab. 4: Spočítané fitovací koeficienty včetně chyby fitu.

	bílá	oranžová	zelená	žlutá	modrá
$\frac{L}{m}$	$A$ $10^{-3} \cdot m \cdot kg^{-1}$	$A$ $10^{-3} \cdot m \cdot kg^{-1}$	$A$ $10^{-3} \cdot m \cdot kg^{-1}$	$A$ $10^{-3} \cdot m \cdot kg^{-1}$	$A$ $10^{-3} \cdot m \cdot kg^{-1}$
0,25	$76,0 \pm 1,9$	$58,0 \pm 0,9$	$65,0 \pm 1,3$	$59,0 \pm 0,9$	$56,0 \pm 0,6$
0,20	$41,0 \pm 1,0$	$31,0 \pm 1,2$	$38,0 \pm 1,0$	$29,0 \pm 1,2$	$29,0 \pm 0,7$
0,15	$17,0 \pm 0,4$	$14,0 \pm 0,4$	$14,0 \pm 0,3$	$14,0 \pm 0,3$	$15,0 \pm 0,4$
0,10	$5,0 \pm 0,1$	$5,0 \pm 0,2$	$4,0 \pm 0,3$	$5,0 \pm 0,2$	$6,0 \pm 0,1$

 Tab. 5: Spočítané hodnoty fitovacího koeficientu  $B$  a výpočet Youngova modulu pružnosti v ohybu.

	$B$ $kg^{-1} \cdot m^{-2}$	$u_B$ $kg^{-1} \cdot m^{-2}$	$E$ GPa	$u_E$ GPa
bílá	4,91	0,05	15,1	0,5
oranžová	3,77	0,08	17,4	0,4
zelená	4,25	0,13	15,7	0,6
žlutá	3,76	0,08	18,7	0,6
modrá	3,63	0,11	22,9	0,9


 Obr. 22: Závislost koeficientu  $A$  na třetí mocnině vzdálenosti podpěr u bílé špejle

## Úloha II.E ... sypká

Změřte sypný úhel alespoň 2 látek běžně používaných v kuchyni (např. mouka, cukr, sůl apod.).

### Teória

Ako je dobre známe, nasypaná látka vytvorí kužel s uhlom sklonu stien (voči vodorovnej rovine)  $\alpha$  – to je hľadaný sypný uhol. Na stenu kužela sa môžeme pozrieť ako na naklonenú rovinu s koeficientom trenia  $\mu$ ; z rovnováhy medzi tiažovou a trecou silou pre vrchnú vrstvu látky dostávame známy vzťah

$$\mu = \operatorname{tg} \alpha .$$

Nejde o úplne realistický model, lebo zrnká môžu držať aj v strmšom kuželi, ak sa na seba vhodne „naskladajú“, teda ak zapadnú do dier medzi inými zrnami tak, že nedržia iba kvôli treniu. Dalo by sa teda čakať, že látka s drsnejšími alebo väčšími zrnami bude mať sypný uhol väčší.

### Postup pri experimente

Sypný uhol budeme merať pre nasledujúce látky: hladká múka, hrubá múka, (kryštalový) cukor, soľ, mak.

Danú látku  $N$ krát ( $N = 10$ ) nasypeme na rovný tanier z čo najmenej výšky a jemne zatrasieme. Výsledný kužel zboku odfoťíme, na strany kužela na fotke nakreslíme priamky a nájdeme uhly, ktoré zvierajú s vodorovnou osou. Zo všetkých  $2N$  hodnôt pre ľavé a pravé strany kuželov vypočítame priemerný uhol  $\bar{\alpha}$  a jeho štandardnú odchýlku (štatistickú) podľa vzťahu

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^{2N} (\alpha_i - \bar{\alpha})^2}{2N(2N - 1)}} .$$

Odhadneme systematickú chybu jedného merania  $\Delta\alpha$ , z ktorej vypočítame štandardnú odchýlku (systematickú)

$$\sigma_B = \sqrt{\frac{1}{2N} \Delta\alpha} ;$$

celkovú štandardnú odchýlku potom vypočítame podľa vzťahu

$$\sigma = \sqrt{\sigma_A^2 + \sigma_B^2} .$$

### Výsledky merania

V tabuľke 6 je uvedený nameraný uhol  $\alpha$  pre jednotlivé látky spolu s priemerom. Na obr. 23 až 27 sú fotky kôpok použitých pri experimentoch.

Chybu určenia uhla  $\alpha$  odhadneme na  $0,5^\circ$ . Potom je systematická odchýlka priemeru rovná  $\sigma_B \doteq 0,1^\circ$ ; vidíme, že je oproti štatistickej zanedbateľná, celková odchýlka teda bude približne rovná štatistickej.

Tab. 6: Nameraný sklon stien nasypného kužela.

$i$	hl. múka		hr. múka		cukor		soľ		mak	
	$\alpha_1[^\circ]$	$\alpha_p[^\circ]$	$\alpha_1[^\circ]$	$\alpha_p[^\circ]$	$\alpha_1[^\circ]$	$\alpha_p[^\circ]$	$\alpha_1[^\circ]$	$\alpha_p[^\circ]$	$\alpha_1[^\circ]$	$\alpha_p[^\circ]$
1	34,5	45,0	40,1	37,5	41,2	41,7	42,2	45,3	40,9	38,2
2	47,8	43,5	38,7	35,2	43,3	37,3	44,1	41,5	40,8	40,1
3	48,6	41,7	37,8	39,6	41,8	39,6	42,7	41,0	38,4	38,1
4	41,1	48,4	39,9	38,3	46,7	39,2	40,2	40,0	38,6	39,2
5	49,1	48,8	38,8	38,0	35,4	44,0	41,2	36,5	40,3	38,4
6	46,0	43,5	43,9	39,0	42,9	41,3	36,6	40,8	37,2	38,8
7	56,4	54,1	37,6	38,4	41,1	39,7	40,4	37,7	38,9	36,5
8	57,5	52,2	37,5	38,0	41,9	40,3	39,9	41,4	37,6	36,1
9	49,8	47,5	38,0	37,0	38,9	41,8	47,3	35,0	42,6	37,5
10	50,0	49,5	38,4	40,8	40,6	40,3	39,7	39,6	37,7	38,4
$\bar{\alpha}$	48°		38,6°		41,0°		40,7°		38,7°	
$\sigma_A$	1°		0,4°		0,6°		0,7°		0,4°	

### Diskusia

Nameraný uhol  $\alpha$  môže byť väčší ako skutočný kvôli naskladaniu zrníek. Kužel nasypanej látky by mal byť ale stabilný aj pri ľubovoľnom menšom uhle, čo sa prejaví hlavne v prípade, že sypeme z väčšej výšky a padajúce zrnká majú dostatočnú kinetickú energiu na to, aby sa zosypali nižšie.

Ďalej experiment ovplyvňuje to, že nevznikne dokonalý kužel – nedá sa sypať presne z jedného bodu, pri podstave a vrchole bude kužel viac zaoblený kvôli padajúcim zrnkám a všeobecne bude nepravidelný tam, kde sa zrnká viac naskladajú na seba. Určiť presne povrch kužela je tiež problém kvôli rozmerom zrníek.

Prípadný sklon fotografie eliminujeme tým, že sčítavame uhly pre pravú a ľavú stenu kužela. Na meranie ale vplýva to, že pri fotení z konečnej diaľky alebo nedokonalé zboku neodfotíme presne prierez kužela.

Pri hladkej múke sa objavuje ten problém, že ľahko tvorí hrudky, ktoré sa vedľa nakopit do dosť veľkej výšky. Tento efekt, na menšej škále, efektívne vytvára väčšie a nepravidelné zrná, aj keď samotné častice múky sú veľmi malé a pravdepodobne guľové. To spôsobuje, že je sypný uhol vcelku vysoký a ťažko sa meria (steny kužela sú dosť nepravidelné). U ostatných meraných látok to nepozorujeme, napr. pri práškovom cukre áno. Pravdepodobne sú hrudkovité len veľmi jemné látky.

Vidíme tiež, že pre mak a hrubú múku je sypný uhol veľmi podobný – v rámci odchýlky merania rovnaký – a pre cukor a soľ je tiež dosť podobný. Podobnosť medzi týmito dvojicami je v tvare zŕn (mak a hrubá múka majú vcelku guľový tvar, cukor a soľ sú kryštalické); na druhej strane veľkosť zŕn nemá na použitých látkach pozorovateľný vplyv.

Ďalším nežiaducim efektom je vlhkosť, ktorá by mala zvýšiť sypný uhol, je ale ťažké odhadnúť o koľko. Použitá soľ bola dosť navlhnutá, ostatné látky boli suché.



Tab. 7: Sypný uhol jednotlivých látok.

látka	sypný uhol
hladká múka	$(48 \pm 1)^\circ$
hrubá múka	$(38,6 \pm 0,4)^\circ$
cukor	$(41,0 \pm 0,6)^\circ$
soľ	$(40,7 \pm 0,7)^\circ$
mak	$(38,7 \pm 0,4)^\circ$

### Záver

Nameraný sypný uhol pre jednotlivé látky je uvedený v tabuľke 7.



Obr. 23: Kôпка hladkej múky.

Použité látky s podobným tvarom zŕn sa správajú podobne, veľkosť zŕn nie je veľmi podstatná. Veľmi jemné látky, ktoré vytvárajú hrudky, budú mať zasa veľký sypný uhol.

Treba podotknúť, že žiadne závery o správaní sypného uhla podľa vlastnosti látky nevieme vyvodit s istotou – použitých látok je príliš málo.



Obr. 24: Kôpka hrubej múky.



Obr. 25: Kôpka cukru.



Obr. 26: Kôpka soli.



Obr. 27: Kôpka maku.

### Úloha III.E . . . magneticky přitažlivá

Společně se zadáním této série jsme vám rozeslali poštou plošný magnet (magnetickou fólii). Tento magnet je trochu jiný než tyčové magnety – v ploše se střídavě střídají severní a jižní pól. Díky tomu se při přiblížení k feromagnetickému povrchu uzavře skrz kov „magnetický obvod“ a magnet drží (např. na ledničce) a unese na sobě třeba i obrázek. Vašimi úkoly jsou:

- Změřit plochu a tloušťku fólie, kterou využijete k experimentům.
- Změřit střední vzdálenost mezi dvěma nejbližšími stejnými magnetickými póly (dvojnásobek opačných).
- Změřit maximální užitečnou hmotnost (tedy hmotnost bez hmotnosti magnetu), kterou unese  $1\text{ cm}^2$  magnetu, je-li zatížení magnetu rovnoměrné, pokud magnet přichytíte zespoda k vodorovně umístěnému cca. 1 mm tlustému plechu z magneticky měkké oceli.

Nezapomeňte určit i chyby měření. Fólie, kterou jsme vám dodali, může být samolepící (je přes ni bílá fólie a pod ní lepidlo). V tom případě bílou fólii nahradte něčím, na co budete upevňovat užitečnou hmotnost.

#### Teória

Na magnetickej fólii dochádza k striedaniu severného a južného pólu v pásoch. Po priložení fólie k feromagnetickému povrchu dôjde k dočasnej magnetizácii materiálu. K odtrhnutiu fólie dôjde vtedy, keď tiažová sila pôsobiaca na jednotku plochy fólie (pri rovnomernom zaťažení) prekoná magnetickú silu, pôsobiacu medzi fóliou a feromagnetickým materiálom.

#### Postup pri experimente

Na to, aby sme zistili plochu a hrúbku fólie, ktorú budeme používať pri meraní, orežeme nepravidelnosti na okrajoch fólie a predelíme ju na polovicu. Následne pomocou posuvného meradla zmeriame rozmery jednej polovice fólie. Plochu určíme ako  $S = a \cdot b$  a celkovú chybu bude tvoriť len systematická chyba merania

$$\sigma_1 = \sqrt{(a\Delta b)^2 + (b\Delta a)^2}.$$

Druhú polovicu fólie využijeme na zistenie strednej vzdialenosti medzi najbližšími súhlasnými magnetickými pólmi, kedy fólie priložíme k sebe a budeme ich po sebe posúvať. V jednom zo smerov pohybu bude dochádzať ku skokom pri posúvaní fólii po sebe. Z týchto skokov dokážeme určiť vzdialenosť medzi pólmi, pretože jeden skok znamená presun daného pólu z nesúhlasného pólu druhej fólie na ďalší nesúhlasný pól. Označme počet týchto skokov  $k$ . Počet súhlasných pólův  $n$  na povrchu fólie teda bude  $n = k + 1$ . Priemernú strednú vzdialenosť medzi pólmi vyjadríme ako  $d = l/k$ , kde  $l$  je veľkosť posunutia pri preskakovaní fólii po sebe. Celkovú odchýlku vyjadríme ako

$$\sigma_2 = \frac{1}{k} \Delta l.$$

Na to, aby sme určili maximálnu užitočnú hmotnosť, ktorú magnet unesie, prípevníme naň mikroténové vrečko, do ktorého budeme postupne pridávať závažie, napríklad kryštálový cukor. Vrečko upevníme tak, aby ťažová sila pôsobila rovnomerne po celej ploche fólie. Pred začiatkom merania zvažíme samotnú fóliu. Po odtrnutí fólie zvažíme hmotnosť vrečka spolu s fóliou. Meranie opakujeme  $N$  krát ( $N = 30$ ). Zo všetkých hodnôt vypočítame priemernú hmotnosť cukru  $m$  a jej štandardnú štatistickú odchýlku podľa vzťahu

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^N (m_i - m)^2}{(N - 1)N}}.$$

Systematickú chybu merania  $\sigma_B$  určíme ako nepresnosť použitej váhy. A teda celkovú chybu merania určíme pomocou vzťahu

$$\sigma_3 = \sqrt{\sigma_A^2 + \sigma_B^2}.$$

Maximálnu užitočnú hmotnosť na jednotku plochy už ľahko dostaneme zo vzťahu  $\rho = m/S$ , pričom odchýlku merania dostaneme pomocou vzťahu

$$\sigma_4 = \sqrt{\left(\frac{\partial \rho}{\partial S} \sigma_1\right)^2 + \left(\frac{\partial \rho}{\partial m} \sigma_3\right)^2} = \sqrt{\left(\frac{\rho}{S} \sigma_1\right)^2 + \left(\frac{\rho}{m} \sigma_3\right)^2}.$$

### Výsledky merania

Namerané rozmery fólie sú  $a = (69 \pm 1)$  mm,  $b = (31 \pm 1)$  mm,  $h = (1,58 \pm 0,03)$  mm. Teda plocha fólie je  $S = (21,4 \pm 0,8)$  cm<sup>2</sup>. Počet skokov  $k$  pri posúvaní fólie po sebe na vzdialenosti  $l = (29,9 \pm 0,3)$  mm bol  $k = 6$ , teda priemerná stredná vzdialenosť medzi pólmi je  $d = (4,98 \pm 0,05)$  mm. Hmotnosť samotnej fólie je  $m_0 = (10,79 \pm 0,01)$  g. Fólia bola pri realizácii experimentu prichytená k doske z magneticky mäkkej ocele o hrúbke  $(1,50 \pm 0,03)$  mm. V tabuľke 8 je uvedená hmotnosť cukru, pri ktorej sa fólia odtrhla. Nepresnosť merania je  $\pm 0,01$  g.

Výsledná priemerná hmotnosť cukru je  $m = (110,8 \pm 0,6)$  g. Z čoho dostávame pre maximálnu užitočnú hmotnosť na jednotku plochy hodnotu  $\rho = (5,2 \pm 0,2)$  g·cm<sup>-2</sup>.

### Diskusia

Hmotnosť, pri ktorej sa magnetická fólia odtrhla, pomerne výrazne menila svoju hodnotu pri konkrétnom meraní, čo je spôsobené tým, že je obtiažne rovnomerne rozložiť hmotnosť pridávaného závažia. Kvôli tomu môže nastať stav, v ktorom je určitý úsek fólie zaťažený oveľa viac ako ostatné, čo spôsobí predčasné odtrhnutie fólie.

Taktiež nepresnosti merania rozmerov fólie sú spôsobené hlavne tým, že fólia nie je ideálne zastrihnutá na okrajoch. A ešte tým, že pri meraní rozmeru  $a$  a rozmeru  $b$  trochu pružila, čím sa jej rozmery deformovali.

Meranie nám taktiež ovplyvňuje aj to, ako veľmi sa zmagnetizuje feromagnetický materiál, keď naň priložíme magnetickú fóliu. To môže spôsobovať prípadné rozdiely v hodnotách  $\rho$  v závislosti od hrúbky materiálu a od jeho magnetickej tvrdosti. Na magneticky tvrdších materiáloch bude užitočná hmotnosť udržaná magnetom vyššia.

Tab. 8: Nameraná maximálna hmotnosť

$N$	$\frac{m}{g}$	$N$	$\frac{m}{g}$	$N$	$\frac{m}{g}$
1	105,42	11	106,87	21	108,51
2	112,73	12	114,28	22	111,67
3	106,94	13	107,53	23	109,34
4	113,53	14	113,12	24	107,73
5	111,31	15	107,85	25	108,90
6	116,05	16	105,25	26	115,25
7	110,59	17	111,41	27	114,58
8	117,17	18	109,56	28	113,09
9	105,73	19	115,10	29	107,00
10	108,42	20	113,97	30	113,74

Tab. 9: Výsledky

$h$	$(1,58 \pm 0,03)$ mm
$d$	$(4,98 \pm 0,05)$ mm
$S$	$(21,4 \pm 0,8)$ cm <sup>2</sup>
$m$	$(110,8 \pm 0,6)$ g
$\rho$	$(5,2 \pm 0,2)$ g·cm <sup>-2</sup>

### Záver

Zmerali sme rozmery magnetickej fólie a hmotnosť, ktorú unesie. Z týchto veličín sme potom spočítali plochu a užitočnú hmotnosť na jednotku plochy. Ďalej sme určili strednú vzdialenosť medzi súhlasnými magnetickými pólmi. Výsledné hodnoty sú zhrnuté v tabuľke 9.

## Úloha IV.E . . . tíha struny

Změřte délkovou hustotu struny, která vám měla přijít poštou společně se zadáním. Strunu ale nesmíte vážit.

Nápověda Zkuste strunu rozkmitat.

### Teorie

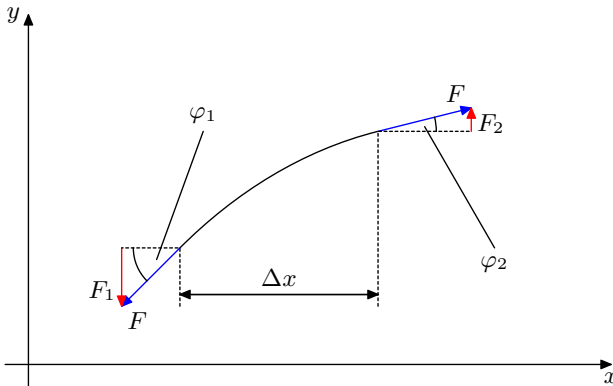
Nejprve odvodíme rovnice, popisující malé výchylky struny v rovině. Necht máme výchylku  $y$  v závislosti na souřadnici  $x$  a čase  $t$ . V bodě  $x$  si zvolme element struny o délce  $\Delta x$  (vzhledem k souřadnici  $x$ ), který je na obou stranách napínán silou  $F$  (viz obrázek 28). Označme  $\varphi_1$  resp.  $\varphi_2$  úhel mezi tečnou ke struně a osou  $x$  na levém, resp. pravém konci zkoumaného úseku. Protože předpokládáme malé výchylky a tím pádem i malé úhly  $\varphi$ , je  $\cos \varphi_{1,2} \approx 1$  a pro  $y$ -ovou složku výsledné síly, působící na daný úsek, platí

$$F_y = F_2 - F_1 = F (\sin \varphi_2 - \sin \varphi_1) \approx F \left( \frac{\partial y(x + \Delta x, t)}{\partial x} - \frac{\partial y(x, t)}{\partial x} \right),$$

kde jsme využili toho, že derivace funkce má význam směrnice tečny ke grafu funkce a  $\sin \varphi_i \approx \text{tg } \varphi_i \approx \varphi_i$ .

Podle 2. Newtonova pohybového zákona je tato síla rovna hmotnosti elementu krát jeho zrychlení ve směru  $y$ . Má-li struna délkovou hustotu  $\mu$ , hmotnost úseku je rovna  $\mu \Delta x$  a my dostáváme

$$\mu \Delta x \frac{\partial^2 y}{\partial t^2} = F_y. \quad (26)$$



Obr. 28: Nákres vlny pro výchylku  $u$

Derivaci výchylky v bodě  $x + \Delta x$  odhadneme prvními dvěma členy Taylorova polynomu. Dosazení do předchozích vztahů pro  $F_y$  vede na

$$F_y = F \left( \frac{\partial y(x, t)}{\partial x} + \frac{\partial^2 y(x, t)}{\partial x^2} \Delta x - \frac{\partial y(x, t)}{\partial x} \right) = F \frac{\partial^2 y(x, t)}{\partial x^2} \Delta x.$$

Nyní můžeme dosadit zpět do pohybové rovnice (26). Dostaneme

$$\frac{\partial^2 y}{\partial x^2} - \frac{\mu}{F} \frac{\partial^2 y}{\partial t^2} = 0.$$

Jak vidíme, dostali jsme vlnovou rovnici, která má pro obecnou amplitudu  $A$  a fázový posun  $\delta$  řešení

$$y(x, t) = A \cos \left( \frac{2\pi}{\lambda} x - 2\pi f t + \delta \right),$$

kde  $f$  je frekvence kmitání,  $\lambda$  vlnová délka a pro fázovou rychlost vlnění  $v_f$  platí

$$v_f = f \lambda = \sqrt{\frac{F}{\mu}}.$$

Máme-li strunu o délce  $l$ , která je upevněná na obou koncích, pro základní mód kmitání bude platit  $\lambda = 2l$  a všechny další vlnové délky budou ve tvaru  $\lambda = 2l/n$  ( $n$  je přirozené číslo). Nás zajímá především základní frekvence (vyšší jsou totiž často utlumené), tedy

$$\frac{1}{f} = 2l \sqrt{\frac{\mu}{F}} = kl. \quad (27)$$

Vidíme že toto je rovnice přímky vzhledem k proměnné  $l$ , kde jsme všechny konstanty schovali do

$$k = 2 \sqrt{\frac{\mu}{F}}. \quad (28)$$

### Postup

Měření jsme provedli tak, že jsme strunu ve vodorovné poloze na jednom konci upevnili, zatímco druhý jsme zatížili předem určeným závažím (lahví s vodou) a nechali jsme ho ležet přes hranu stolu. Poté jsme ke struně umístili mikrofon, propojený s notebookem a pomocí programu Audacity jsme nahrávali zvuk rozkmitané struny. Z nahrávek jsme poté pomocí frekvenční analýzy zjistili vlastní frekvenci struny.

Měřili jsme jak pro různé délky struny, od 0,2 m do 1,4 m, tak pro různé hmotnosti závaží, od 0,6 kg do 1,6 kg. Délku struny jsme měřili pásmem, hmotnosti kuchyňskými vahami.

Pro každou hodnotu délky a napětí struny jsme strunu rozkmitali několikrát a nahrávku jsme analyzovali jako celek.



Tab. 10: Tabulka naměřených frekvencí.

$l$ m	$f(0,6 \text{ kg})$ Hz	$f(0,8 \text{ kg})$ Hz	$f(1,0 \text{ kg})$ Hz	$f(1,2 \text{ kg})$ Hz	$f(1,4 \text{ kg})$ Hz
1,4	90	107	-	-	-
1,3	94	107	-	-	106
1,2	98	101	-	102	106
1,1	100	91	109	91	131
1,0	-	106	110	130	148
0,9	-	115	150	160	154
0,8	130	146	152	159	-
0,7	149	156	-	-	-
0,6	247	261	255	250	232
0,5	248	265	266	284	261
0,4	278	285	281	301	308
0,3	287	306	280	416	431
0,2	420	562	675	692	678

### Výsledky

Naměřené frekvence jsme zanesli do tabulky 10. Pro nějaké kombinace napětí a délek se nám nepodařilo naměřit reálnou frekvenci, proto jsou tato políčka tabulky prázdná.

Do grafů 29 až 33 jsme vynesli závislost převrácené hodnoty frekvence (neboli periody) na délce struny. Každý graf jsme proložili přímkou a určili jsme její směrnici. Výsledky jsou v tabulce 11.

Tab. 11: Spočítané směrnice přímek.

$m$ kg	$k$ $10^{-3} \text{ s} \cdot \text{m}^{-1}$
0,6	$8,51 \pm 0,25$
0,8	$8,22 \pm 0,36$
1,0	$8,22 \pm 0,32$
1,2	$8,12 \pm 0,35$
1,4	$7,29 \pm 0,13$

### Výpočet

Napínací síla  $F$  v rovnici (28) je tíha závaží  $mg$ , můžeme tedy snadno spočítat hledanou délkovou hustotu struny

$$\mu = \frac{k^2 F}{4} = \frac{k^2}{4} mg.$$

Tab. 12: Délková hustota.

$\frac{m}{\text{kg}}$	$\frac{\mu}{\text{g}\cdot\text{m}^{-1}}$
0,6	$0,107 \pm 0,006$
0,8	$0,133 \pm 0,012$
1,0	$0,166 \pm 0,013$
1,2	$0,194 \pm 0,017$
1,4	$0,182 \pm 0,007$

Nejistotu jednoho měření  $\mu$  spočítáme klasicky z parciálních derivací

$$u_\mu = \sqrt{\left(\frac{\partial\mu}{\partial k} u_k\right)^2 + \left(\frac{\partial\mu}{\partial m} u_m\right)^2} = \mu \sqrt{\left(\frac{2u_k}{k}\right)^2 + \left(\frac{u_m}{m}\right)^2}.$$

Chybu měření hmotnosti uvažujeme  $u_m = 0,005$  kg,  $u_k$  dostaneme jako parametr fitu. Výsledky jsou uvedeny v tabulce 12.

Délková hustota nám vyšla pro každé napětí struny jiná, což by mohlo poukazovat na jistou závislost mezi napětím a délkovou hustotou struny. Tato závislost by ovšem měla být klesající (čím větší napětí, tím menší délková hustota), z čehož soudíme, že různost výsledků je pouze statistická chyba. Spočítáme jejich aritmetický průměr a výběrovou směrodatnou odchylku průměru jakožto nejistotu typu A, největší z chyb jednotlivých výsledků budeme považovat za nejistotu typu B. Kombinovanou nejistotu měření pak spočítáme jejich kvadratickým součtem

$$\begin{aligned} \bar{\mu} &\doteq 0,16 \text{ g}\cdot\text{m}^{-1}, \\ u_{\mu,A} &\doteq \sigma_\mu = 0,02 \text{ g}\cdot\text{m}^{-1}, \\ u_{\mu,B} &\doteq 0,02 \text{ g}\cdot\text{m}^{-1}, \\ u_\mu &= \sqrt{u_{\mu,A}^2 + u_{\mu,B}^2} \doteq 0,03 \text{ g}\cdot\text{m}^{-1}. \end{aligned}$$

Naměřená hodnota délkové hustoty je  $\mu = (0,16 \pm 0,03) \text{ g}\cdot\text{m}^{-1}$ .

### Diskuse

Měření délkové hustoty je řádově správné, ale poměrně nepřesné. Největším zdrojem chyb je určení vlastní frekvence struny. To je způsobené špatnou kvalitou nahrávky v důsledku okolního šumu. Dále se v nahrávce nachází výrazných signálů více, proto nemusí být vždy jasné, která z frekvencí je vlastní. Dalším zdrojem nepřesností může kromě poměrně přesných měření délky a hmotnosti být také samotná změna délkové hustoty během měření, například odlupováním barvy z kovové části struny nebo její pružnou deformací. Nesmíme taky zapomenout na to, že frekvence tlumených kmitů se vůči teorii bez tlumení posouvá.

Výrazným zdrojem chyb může být to, že strunu napíná závaží – lahev s vodou, která se může různě hýbat (jako kyvadlo, obyčejný pohyb vody uvnitř lahve, kmity

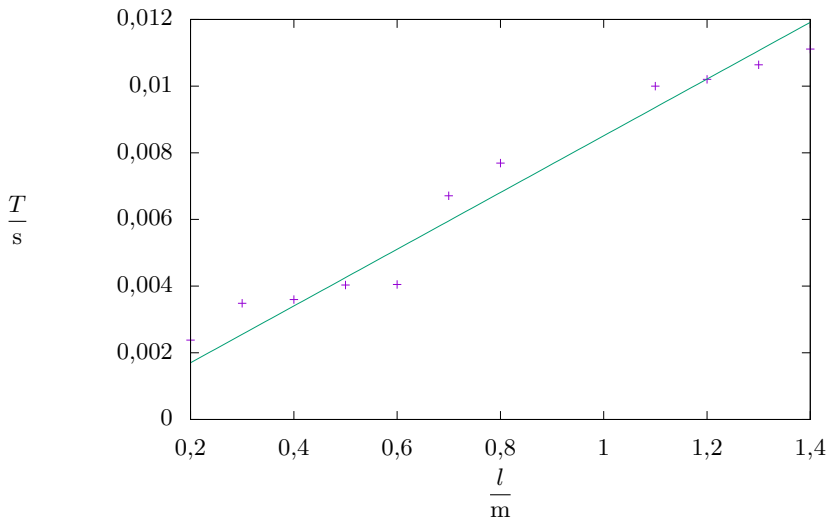
vody uvnitř lahve) v závislosti na tom, jak strunu rozkmitáme. To může ovlivňovat skutečné kmity struny i samotnou frekvenční analýzu.

Po měření šuplerou vychází poloměr struny zhruba 0,15 mm, což při započítání modulu pružnosti konstrukční oceli  $E = 210 \text{ GPa}$  dává odhad relativního prodloužení struny  $\varepsilon = 0,1 \%$ . Je to sice jen řádový odhad, ale i tak můžeme tuto chybu považovat za zanedbatelnou. Typické hmotnosti kovového drátu s daným poloměrem se pohybují v rozsahu od  $0,2 \text{ g}\cdot\text{m}^{-1}$  (hliník) do  $0,6 \text{ g}\cdot\text{m}^{-1}$ , naměřená hmotnost je tedy poněkud nízká, ale řádově správná (měření poloměru může být taky ovlivněno barevným nátěrem).

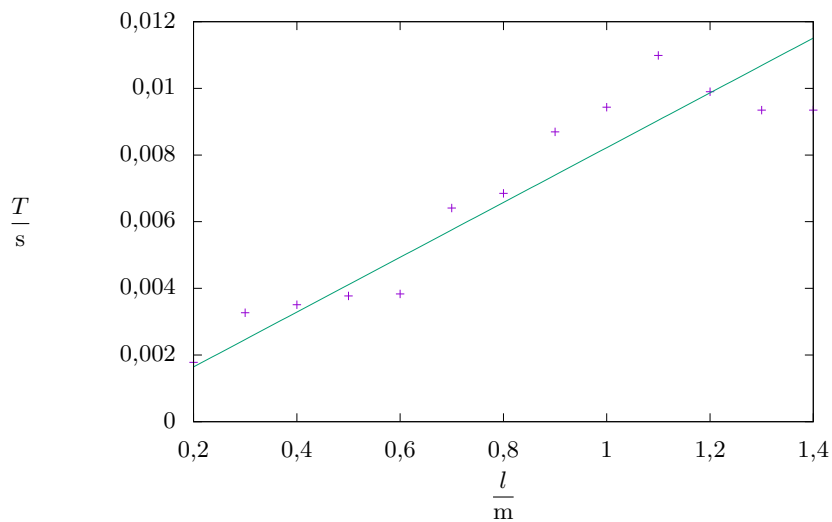
Navíc vůbec neuvažujeme například vliv teploty okolí a další jevy. Vzhledem k nepřesnostem při určování vlastní frekvence je však můžeme považovat za zanedbatelné.

### Závěr

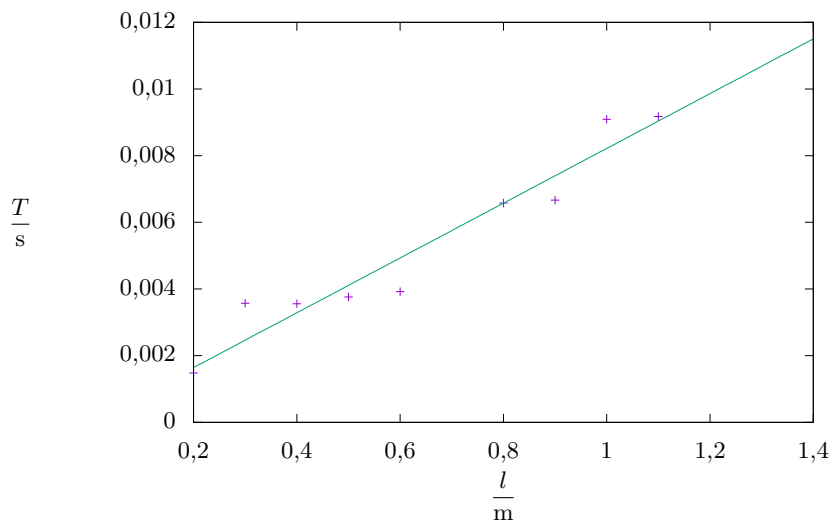
Naměřená délková hustota struny je  $\mu = (0,16 \pm 0,03) \text{ g}\cdot\text{m}^{-1}$ . Hmotnost celé struny tak vychází přibližně 0,2 g, což je řádově správný výsledek.



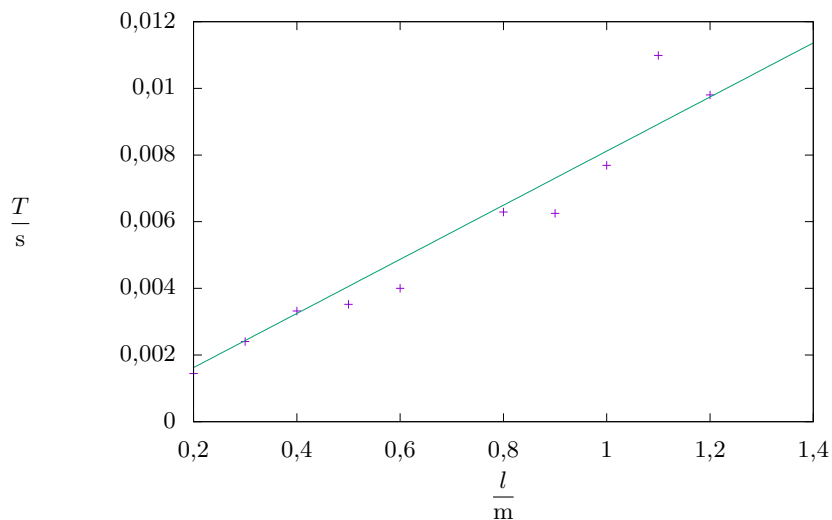
Obr. 29: Závislost periody na délce struny pro  $m = 0,6 \text{ kg}$ .



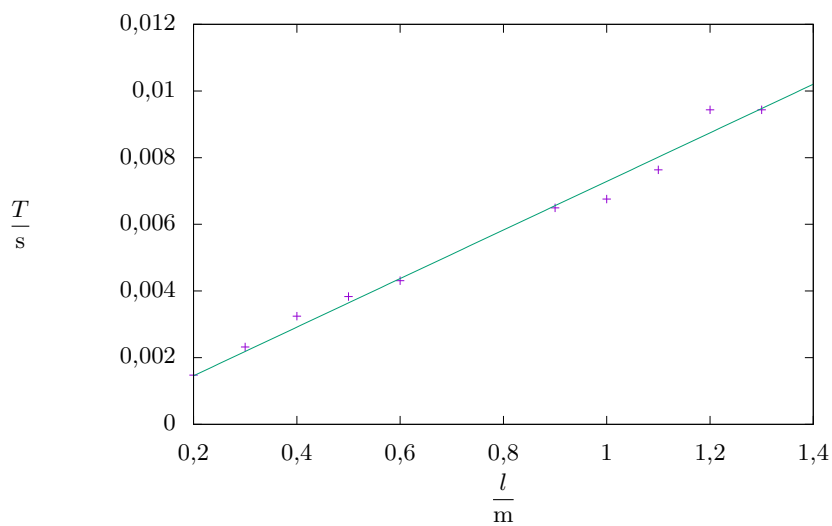
Obr. 30: Závislost periody na délce struny pro  $m = 0,8$  kg.



Obr. 31: Závislost periody na délce struny pro  $m = 1,0$  kg.



Obr. 32: Závislost periody na délce struny pro  $m = 1,2$  kg.



Obr. 33: Závislost periody na délce struny pro  $m = 1,4$  kg.

## Úloha V.E ... nezbedné fotony

*Spolu se zadáním úlohy vám přišly polarizační brýle. Máte tedy 2 polarizační filtry. Když je dáte za sebe tak, aby směry jejich polarizace byly na sebe kolmé, nebude skrz ně procházet téměř žádné světlo. Pokud ale mezi ně nyní vložíte třetí vhodně natočený filtr, můžete pozorovat, že bude procházet nemalé množství světla. Změřte závislost propustnosti na úhlu natočení prostředního filtru.*

*Poznámka: Jako první filtr a zároveň zdroj světla doporučujeme použít svítící displej.*

### Teorie

Každé příčné vlnění, tedy i paprsek světla, má svojí polarizaci. Polarizace je dána rovinou, ve které se vlna pohybuje. Světlo se jistě pohybuje po přímce, tedy rovina polarizace musí obsahovat tuto přímku a okolo ní může být natočena libovolným směrem. Většina světla, které vidíme, je nekoherentní (tzn. každý foton má jinou polarizaci, není zde žádná preferované rovina pohybu fotonů, světlo je nepolarizované), protože vzniká odrazem slunečního světla (které je také nekoherentní).

Téměř všechny displeje, monitory a televize však vyzářují polarizované světlo, které má pevně daný směr polarizace a všechny vycházející fotony tedy leží v rovnoběžných rovinách.<sup>3</sup>

Když polarizované světlo (foton) projde polarizačním filtrem, zachová si pouze složku amplitudy, která je rovnoběžná se směrem osy daného polarizátoru. Pokud je původní polarizace světla kolmá na filtr, pak přes něj neprojde. V opačném případě (filtr je rovnoběžný se směrem polarizace) projde ideálně všechno světlo. Tím se dostáváme k tomu, že použité filtry nejsou ideální a když filtr umístíte rovnoběžně, tak přeci jen nějakou část světla pohltní, a i při kolmém umístění projde nějaké (velmi malé) množství světla. Dále však uvažujme „ideální“ polarizační fólii.

Necheť máme zdroj koherentního světla (například monitor) o amplitudě  $A_0$ . Při průchodu polarizačním filtrem je zachována pouze složka amplitudy rovnoběžná s osou polarizačního filtru. Pokud je filtr vůči zdroji světla natočen o úhel  $\alpha$ , průchodem světla se amplituda zmenší na

$$A_1 = A_0 \cos \alpha .$$

Pro  $\alpha = 0^\circ$  (polarizace světla je před průchodem rovnoběžná s osou filtru) se amplituda nezmění, pro  $\alpha = 90^\circ$  (polarizace světla je kolmá na osu filtru) bude amplituda nulová (neprojde žádné světlo). Například ale pro  $\alpha = 45^\circ$  nebude amplituda poloviční, ale zmenší se na  $\frac{1}{\sqrt{2}}A_0$ .

Prošlé světlo si zachová pouze složku amplitudy<sup>4</sup> rovnoběžnou se směrem osy daného filtru. Nyní necháme světlo projít druhým polarizačním filtrem, který je

<sup>3</sup>Pozor, některé fólie na monitory či mobilní telefony mohou světlo „odpolarizovat“.

<sup>4</sup>Mluvíme tady o amplitudě, ale ve skutečnosti jde o pravděpodobnost. Každý foton, který prochází polarizačním filtrem má určitou pravděpodobnost, že celý projde skrz (příčemž změní polarizaci) nebo že bude celý pohlcen. Pokud fotonů prochází hodně (což v našem případě je rozhodně splněno), dostáváme stejné výsledky, jako když počítáme s tím, že se zachovává pouze daná složka amplitudy.

natočen o úhel  $\beta$  oproti původnímu zdroji světla. Amplituda se zmenší na  $A_2$ . Musíme dávat pozor, který úhel dosazujeme, protože nyní už je polarizace světla otočena o úhel  $\alpha$  oproti původnímu směru, takže platí

$$A_2 = A_1 \cos(\beta - \alpha) .$$

V našem případě uvažujme  $\beta = 90^\circ$ , což vede na

$$A_2 = A_1 \cos(90^\circ - \alpha) = A_1 \sin \alpha = A_0 \sin \alpha \cos \alpha .$$

Nyní je potřeba si uvědomit, že se zajímáme o intenzitu prošlého světla a ne o jeho amplitudu. Intenzita světla je přímo úměrná jeho energii (resp. výkonu dopadajícímu na jednotkovou plochu) a energie je úměrná druhé mocnině amplitudy (stejně jako u mechanických harmonických kmitů). Intenzitu značíme  $I$  a pro vztah počáteční a konečné intenzity platí

$$I_2 = I_0 \sin^2 \alpha \cos^2 \alpha = \frac{1}{4} I_0 \sin^2 2\alpha .$$

Bude se nám hodit, že závislost  $I_2$  na  $\alpha$ , tedy  $I_2(\alpha)$ , je  $\frac{\pi}{2}$ -periodická a navíc je každá perioda symetrická, takže stačí proměřit rozpětí od  $0^\circ$  do  $45^\circ$ .

Problém je, že nepracujeme s dokonalými filtry (nikdy neprojde 100 % ani 0 %), takže bude potřeba zavést nějakou korekci. Toho můžeme docílit například přeškálováním naší funkce

$$I_2(\alpha) = \frac{b}{4} I_0 \sin^2 2\alpha ,$$

kde  $b$  je zatím neznámá konstanta, která určuje maximální propustnost filtru.

Uvážíme-li ještě korekci na to, že nenajdeme dokonale kolmý směr (tj. vůči tomuto směru bude filtr posunut o  $\alpha_0$ ), máme

$$I_2(\alpha) = \frac{b}{4} I_0 \sin^2(2\alpha + 2\alpha_0) . \quad (29)$$

### Postup

V dostatečně zatemněné místnosti (abychom eliminovali vliv vnějšího světla, které odráží horní polarizační filtr) zapneme monitor, na kterém zobrazíme čistě bílé pozadí, s úhломěrem, který nám bude ukazovat úhel náklonu, a podle toho budeme otáčet prostřední filtr. Nejprve je potřeba umístit vrchní (druhý) polarizátor tak, aby jeho osa byla kolmá na směr polarizace světla svítícího z monitoru. Toho docílíme pomalým otáčením filtru a sledováním „od oka“, kdy nastane minimum propustnosti světla. V tomto minimu daný filtr upevníme lepicí páskou.

Poté vložíme prostřední filtr, kterým budeme postupně otáčet. Úhel natočení zvolíme pomocí úhломěru, podle kterého zarovnáme rovnou stranici poloviny brýlí.

Jako zařízení na měření intenzity světla použijeme mobilní fotoaparát, který je důležité mít během celého měření pevně ukotvený na stojanu a neměnit jeho nastavení (popř. si dát pozor, aby si neměnil nastavení sám podle potřeby).

Fotografie posléze analyzujeme v nějakém programu, který dokáže určit jas. Konkrétně my jsme použili GIMP, který dokonce umožňuje vybrat určitou oblast a spočítat její průměrný jas.

### Gamma korekce

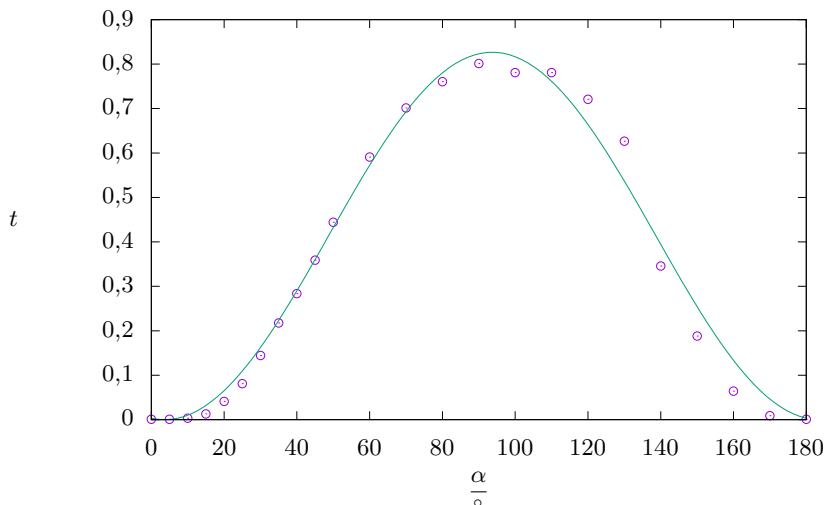
Jelikož citlivost lidského oka není lineární (snadno rozlišíme změnu malé intenzity světla, ale těžko bychom od sebe rozlišovali dvě vysoké intenzity), používá se při ukládání a zobrazování fotografií tzv. gamma korekce. Proto intenzita, kterou ukazuje program, nemusí být její skutečná hodnota. Naměříme-li intenzitu  $I_n$  (tj. hodnotu, kterou má uloženou počítač), pak její skutečnou hodnotu  $I_s$  můžeme vypočítat jako

$$I_s = I_n^\gamma, \quad (30)$$

kde  $\gamma$  je gamma faktor. Pro většinu dnes používaných zařízení se užívá hodnota  $\gamma = 2,2$ .

Fotoaparát má za určitého nastavení danou maximální světelnou intenzitu, kterou dokáže zaznamenat.  $I_n$  má nějakou maximální hodnotu (typicky buď 255 nebo 100 %). V případě, že se na fotografii takováto hodnota vyskytuje, je fotografie přeexponovaná a je potřeba provést jednu nebo více z následujících úprav. Snížit expoziční čas (dobu, po který je snímána jedna fotografie), přivřít clonu (omezit množství světla, které se dostane do snímáče) nebo snížit citlivost snímáče (to najdete pod zkratkou ISO). Je tedy důležité dbát na to, aby naše fotografie nebyly přeexponované. Vztahem 30 pak dokážeme naměřené intenzity linearizovat.

### Naměřená data



Obr. 34: Naměřená data, proložená závislostí 29.

Naměřená intenzita bílého pozadí byla 94 %. Po gamma korekci (30) to dává



$I_0 \doteq 87\%$ . Propustnost  $t$  je definována jako podíl propuštěné intenzity  $I_s$  a počáteční intenzity  $I_0$

$$t = \frac{I_s}{I_0}.$$

Tab. 13: Naměřené hodnoty intenzity a propustnosti  $t$  v závislosti na úhlu natočení.

$\frac{\alpha}{^\circ}$	$\frac{I_n}{\%}$	$\frac{I_s}{\%}$	$t$
0	4	0,1	0,001
5	4	0,1	0,001
10	7	0,3	0,003
15	13	1,1	0,013
20	22	3,6	0,041
25	30	7,1	0,081
30	39	12,6	0,144
35	47	19,0	0,218
40	53	24,7	0,283
45	59	31,3	0,359
50	65	38,8	0,444
60	74	51,6	0,591
70	80	61,2	0,701
80	83	66,4	0,760
90	85	69,9	0,801
100	84	68,1	0,781
110	84	68,1	0,781
120	81	62,9	0,721
130	76	54,7	0,626
140	58	30,2	0,346
150	44	16,4	0,188
160	27	5,6	0,064
170	11	0,8	0,009
180	4	0,1	0,001

V grafu 34 jsou tato data vynesena a proložena závislostí (29). Koefficienty  $\alpha_0$  a  $b$  byly určeny pomocí fitu

$$\alpha_0 = (-3,7 \pm 0,8)^\circ,$$

$$b = 3,80 \pm 0,06.$$

## Závěr

Změřili jsme závislost propustnosti  $t$  na úhlu natočení prostředního filtru. Naměřené hodnoty odpovídají teoretické předpovědi. Celkově nám vychází

$$I_2(\alpha) = \frac{3,80 \pm 0,06}{4} I_0 \sin^2(2\alpha - (7,4 \pm 1,6)^\circ).$$

## Úloha VI.E . . . nehrajte si se sirkami

Změřte rychlost hoření špejle v závislosti na úhlu naklonění vůči vodorovné rovině.

Dřevo (respektive jeho hlavní složky) při hoření za normálních podmínek<sup>5</sup> nereaguje přímo s kyslíkem. Nejdříve totiž musí působením tepla dojít k rozpadu molekul dřeva na menší molekuly (neboli k pyrolýze). Ta má několik fází, které se nejlépe popisují teplotou dřeva. V první fázi, která probíhá přibližně do 200 °C, se dřevo primárně vysušuje, tato fáze je tedy endotermická. Ve druhé fázi, která probíhá do cca 280 °C, sice již dochází k tepelnému rozkladu sloučenin ve dřevě, stále se ale jedná převážně o endotermické reakce, při kterých nevznikají příliš hořlavé plyny. Dřevo v této fázi tmavne. Nad teplotou 280 °C se pyrolýza už stává exotermickou reakcí a vznikají sloučeniny, které snadno hoří. V této fázi pozorujeme plamen. Při teplotě nad cca 500 °C se dřevo skládá téměř čistě z uhlíku, který reaguje s kyslíkem.<sup>6</sup>

K zapálení musí mít dřevo dostatečný přísun tepla, aby dokázalo překonat endotermické fáze pyrolýzy a přešlo do třetí fáze, ve které vzniká teplo nutné k udržení celého procesu hoření. Čím více nehořící část špejle zasahuje do plamene, tím rychleji zvládne získat dostatek tepla, aby mohla také začít hořet. To se projeví vyšší rychlostí hoření. Zároveň je možné, že pro některé úhly už nebude docházet k dostatečnému přenosu tepla, aby se reakce udržela, a špejle zhasne. Také je zřejmé, že rychlost hoření bude záviset na množství tepla potřebného k překonání první fáze. To závisí na složení dřeva a především na jeho vlhkosti.<sup>7,8</sup>

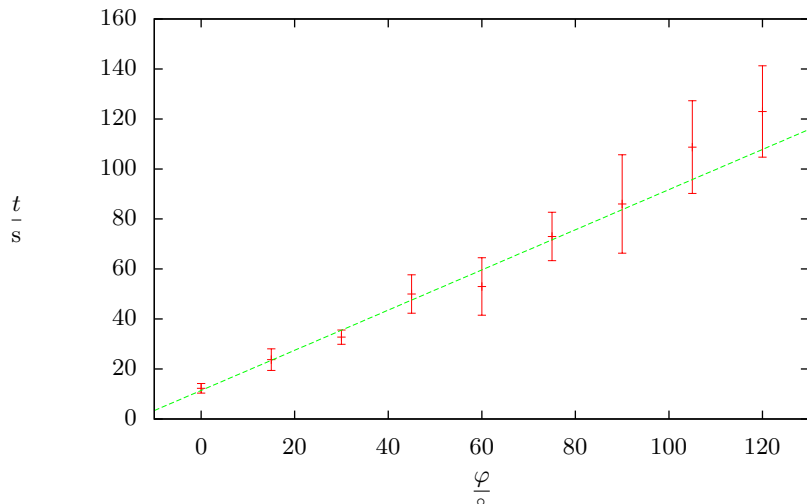
Měření probíhalo tak, že jsme přibližně uprostřed špejle vyznačili po 10 cm značky, špejli jsme uchytili do stojanu a změřili jsme čas mezi dohořením k první a ke druhé značce. Jako místo, kde je plamen, jsme brali rozhraní světlé a tmavé části špejle, které bylo nejvíce ostré. Měřili jsme pro úhly v rozsahu od 0° do 120° po 15°. Pro každý úhel jsme provedli 4 měření, ze kterých jsme vypočítali střední hodnotu a směrodatnou odchylku. Tyto hodnoty jsme vynesli do grafu 35 a pro přehlednost jsme je proložili afinní funkcí  $ax + b$ . Následně jsme z klasického vzorce

<sup>5</sup>Nenormálními podmínkami je myšlena například přítomnost silného oxidačního činidla, případně vhodného katalyzátoru.

<sup>6</sup><https://www.fpl.fs.fed.us/documnts/fplmisc/rpt2136.pdf>

<sup>7</sup>To jsme skutečně ověřili. Když jsme zkusili zapálit špejle, které jsme našli v zásobě doma, zjistili jsme, že nehoří ani ve vodorovné poloze a že se musí držet téměř svisle s plamenem dole, aby vůbec hořely. Proto jsme je za účelem vysušení vložili na 20 minut do trouby, která byla předem rozehrátá na 150 °C. Vysoušení pomohlo a špejle poté hořely i ve svislé poloze s plamenem nahore. Pro samotné měření jsme však použili jiné špejle.

<sup>8</sup>Vliv vody je nejvýraznější, protože má velmi vysoké měrné vypařovací teplo a také její obsah ve dřevě nejvíce kolísá.



Obr. 35: Závislost doby hoření na úhlu od kolmice. ( $0^\circ$  odpovídá svisle orientované špejli hořící zdola nahoru,  $90^\circ$  pak vodorovné pozici.)

pro rychlost vypočítali rychlost hoření, kterou jsme vynesli do grafu 36. Ten jsme proložili funkcí  $1/(ax + b)$ , která vznikla transformací původní afinní funkce.

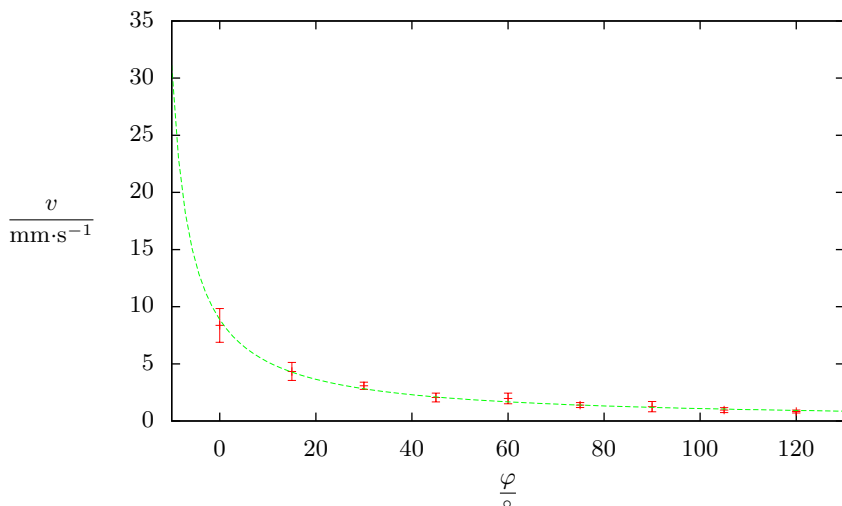
Z grafu 35 vidíme, že čas je afinní funkcí úhlu,<sup>9</sup> což sice vypadá jako zajímavý jev, ale pro nás je mnohem důležitější rychlost hoření. Ta by intuitivně měla být přibližně úměrná výkonu, který se přenáší do dřeva. Jak vidíme, je popisována hyperbolou, která na první pohled nenaznačuje žádné fyzikální vysvětlení. Proto jsme stejná data i proloženou funkci vynesli v polárních souřadnicích do grafu 37, který již vypadá mnohem zajímavěji. Z dat (a tedy především proložené funkce) vidíme, že tvoří tvar nápadně podobný plamenu svíčky. Tvar plamene jistým způsobem odpovídá rozložení tepla v oblasti hoření. Proto špejle bude dostávat celkový výkon úměrný průmětu dané oblasti grafu do přímky, která odpovídá špejli. Rychlost hoření pak bude přímo úměrná tomuto výkonu.<sup>10</sup>

Výsledky měření jsou ovlivněny mnoha různými okolnostmi. Největší vliv na rychlost hoření má vlhkost dřeva. Ta výrazně ovlivní dobu sušení, a tím i celkovou rychlost hoření, jak jsme již zmínili výše. Vzhledem k tomu, že proudění plynů v okolí plamene ovlivňuje teplo, které je přenášeno, bude rychlost hoření záviset na proudění vzduchu v místnosti.<sup>11</sup> Dále se samozřejmě mohou projevit nejistoty

<sup>9</sup>Pro velké úhly se projevuje určitá nelinearita. V této fázi ale špejle téměř nehoří a často zhasíná, čímž je přesnost měření i našeho modelu silně ovlivněna.

<sup>10</sup>Samozřejmě, že výkon je rozprostřen po špejli rovnoměrně, ale během hoření se za čas potřebný k přehoření nějakého úseku v daném bodě energie potřebná k překonání endotermických fází reakce nastřádá rychleji.

<sup>11</sup>Venku je naprosto nereálné cokoliv smysluplného naměřit.

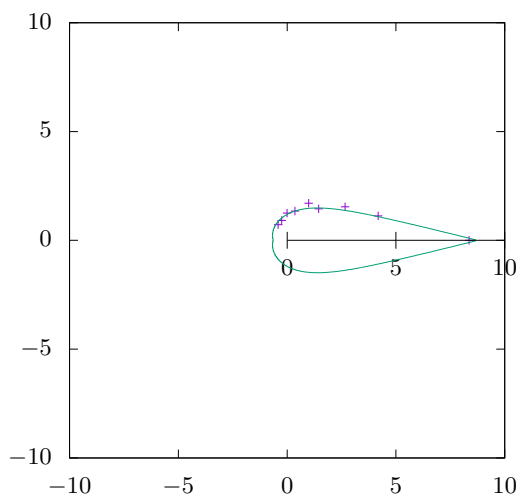


Obr. 36: Závislost rychlosti hoření na úhlu od kolmice. ( $0^\circ$  odpovídá svisle orientované špejli hořící zdola nahoru,  $90^\circ$  pak vodorovné pozici.)

měření času a úhlu. Také je možné, že špejle mají lehce odlišnou rychlost hoření z důvodu různých vlastností materiálu, ale to jsme částečně vyloučili použitím špejlí ze stejného balení. Můžeme předpokládat, že jsou z podobného materiálu, skladovaného ve stejných podmínkách. Navíc jsme měření opakovali, čímž bylo možné odhadnout rozptyl času pro daný úhel. Nemyslíme si, že napříč měřeními s různými podmínkami či špejlemi by mělo smysl porovnávat číselné hodnoty rychlosti, protože je téměř nemožné odfiltrovat vnější vlivy.

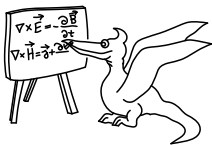
Všechny tyto okolnosti však nemají tak zásadní dopad, aby změnilly charakter závislosti, kterou máme rozumně podloženou teorií. Explicitní tvar závislosti rychlosti hoření na čase sice z teorie neznáme, ten ale není příliš podstatný. Může se jednat o jinou funkci, například mocninnou, případně exponenciální, neboť pro velké úhly se začínají časy odchylovat od lineárního průběhu. Z tohoto důvodu by bylo zajímavé provést měření pro vysušené špejle, které umožňují měřit ve velkém rozsahu úhlů, čímž bychom zjistili, zda je nelinearita dána jen tím, že plamen je na pokraji zhasnutí, nebo zda to má fundamentálnější příčinu. Druhý zajímavý problém, který by bylo vhodné dále zkoumat, je získání prostorového modelu teploty v plameni pomocí metod mechaniky kontinua, ať už numericky nebo analyticky.<sup>12</sup> Oba tyto návrhy ovšem přesahují rámec tohoto textu, tak je necháváme jako námět pro další výzkum.

<sup>12</sup>V druhém případě by samozřejmě bylo nutné model silně zjednodušit rozsáhlými aproximacemi a ani poté není vůbec jisté, zda-li by se nám podařilo nalézt řešení.



Obr. 37: Závislost rychlosti hoření na úhlu, vyneseno v polárních souřadnicích.  
Radiální osa je v  $\text{mm}\cdot\text{s}^{-1}$





## Numerické metody a počítačové simulace

### Kapitola 1: Rozjezdová

#### Úvod

S třicátým prvním ročníkem FYKOSu přichází i nový seriál, v němž se budeme zabývat tématem, které zasahuje téměř do všech ostatních fyzikálních oborů – jedná se o počítačovou fyziku. Fyzika byla jednou z prvních oblastí vědy, která začala počítače naplno využívat a dnes jim vděčíme za velkou část pokroku, kterého jsme na poli experimentu i teorie v posledním půlstoletí dosáhli. Proto jsme usoudili, že ačkoli bylo toto téma již ve FYKOSích seriálech zpracováno<sup>1</sup> (ročníky VIII. a XXI.), neuškodí se k němu opět vrátit.

V průběhu roku budeme sledovat paralelně dvě různé linie: v jedné se budeme zabývat numerickými metodami a v druhé počítačovými simulacemi. V numerice vás seznámíme se základními úlohami, které by měl každý fyzik umět řešit, jako je hledání kořenů algebraických rovnic, integrace diferenciálních rovnic a další. Simulace budou poněkud hravější se zaměřením především na stochastickou metodu Monte Carlo a na jednoduché modely fyzikálních i ne-tak-fyzikálních procesů, jako je perkolace, růst krystalů, tvorba kolon v silničním provozu, šíření lesních požárů a mnoha dalších.

Nemá smysl zastírat, že vzhledem ke zvolenému tématu nebude možné se zcela vyhnout programování. Ačkoli se budeme snažit, aby každý díl seriálu obsahoval dostatek analyticky řešitelných úloh (tj. řešitelných bez pomoci počítače), bude s průběhem roku přibývat těch, které vyžadují sepsání krátkého skriptu. Nemíníme zde suplovat KSP<sup>2</sup>, proto Vás po krátkém úvodu do programování prezentovaném níže odkazujeme na internetové tutoriály. Pokud to myslíte s fyzikou vážně, tak se programování v budoucnu nevyhnete, a je lepší začít dříve než později. Pro bojácne ještě uvedme, že na vyřešení některých numerických úloh bude stačit Microsoft Excel či jiný tabulkový editor (nicméně tento přístup nedoporučujeme).

To je vše, co jsme Vám chtěli úvodem sdělit. Doufáme, že pro Vás bude letošní seriál poučný i zábavný.

#### Základy počítačové fyziky

Počítače jsou dnes již běžnou součástí našich životů, nejinak je tomu ve vědě a s počítačovou fyzikou a numerickými metodami se setká nejspíše každý fyzik. Do této

<sup>1</sup><http://fykos.cz/ulohy/archiv>

<sup>2</sup>Korespondenční seminář z programování MFF UK, <https://ksp.mff.cuni.cz/>.

oblasti totiž nespádají pouze numerické simulace, ale i všednější problémy jako vyčíslení nějaké složitější funkce, či nalezení kořenů algebraické rovnice, kterou nelze vyřešit analyticky.<sup>3</sup> Může se zdát, že jde v tomto případě o triviální problémy řešitelné selským rozumem, a nějaké speciální studium numerických metod k jejich vyřešení snad ani není potřeba. Nicméně, jak sami brzy poznáte, numerické metody bývají mocné, ale zrádné, ve výjimečných případech může jít i o život.<sup>4</sup> Proto je vhodné se s nimi alespoň trochu seznámit a zjistit, jak fungují. **Ale neprosťo nutně je jim nikdy bezmezně nevěřit a vždy mít nějakou kontrolu či intuici plynoucí z daného fyzikálního problému, že výsledek získaný numerickou cestou není zcela špatně.**

Tato zrádnost numerických metod plyne z numerických chyb. Ty mohou vzniknout dvěma hlavními způsoby:

- chyby metody – způsobené aproximacemi oproti analytickému řešení
- zaokrouhlovací chyby

### Reprezentace čísel v počítači

Zaokrouhlovací chyby jsou způsobeny tím, že desetinná čísla v počítači nejsou uložena přesně. Nejmenší jednotkou informace v počítači, se kterou lze přímo operovat, je *byte*. Ten obsahuje 8 bitů, chlívčků, do kterých můžeme zapsat jedničku, nebo nulu. Přirozenou číselnou soustavou počítače tedy není desítková soustava, jako u člověka,<sup>5</sup> ale soustava dvojková.

Čísla v počítači jsou dvou běžných typů, celá (*integer*) a reálná (s plovoucí desetinnou čárkou, *float*). Celá čísla dokážeme uložit přesně v nějakém rozsahu, závisícím na velikosti použitého bloku paměti. Např. do jednoho bytu dokážeme uložit čísla od 0 do 255, nebo od  $-128$  do 127. Většinou se ale používají celá čísla o délce 4 byty (*integer*) nebo 8 bytů (*long*).

Reálná čísla se ukládají v tzv. semilogaritmickém tvaru

$$s \cdot m \cdot \beta^e,$$

kde  $s$  je znaménko,  $m$  je mantisa,  $e$  exponent a  $\beta$  základ číselné soustavy (v našem případě  $\beta = 2$ ). Mantisa i exponent pak mají pevně danou velikost paměti, závisící na použitém datovém typu. Ve skutečnosti jsou reálná čísla uložena o něco složitěji<sup>6</sup>, nám ale postačí tato základní představa. Tento způsob uložení zajišťuje udržení počtu platných cifer i pro čísla lišící se o mnoho řádů. Například typ *double*, který budeme výhradně používat, dokáže uložit kladná i záporná čísla s absolutní velikostí od přibližně  $10^{-324}$  do  $10^{308}$  s přesností na 15 až 16 platných cifer. Tento údaj se nazývá *strojová přesnost* a je hlavním údajem ovlivňujícím velikost zaokrouhlovacích chyb.

Jako ilustrace, že k zaokrouhlení dochází, poslouží výpočet  $0,1 + 0,2$ . V *double precision* obdržíme výsledek  $0,30000000000000004$ . Proč zde ale k zaokrouhlení

<sup>3</sup>Jako příklad poslouží rovnice  $x = \sin x$ .

<sup>4</sup><http://www-users.math.umn.edu/~arnold/disasters/>

<sup>5</sup>I když my také dokážeme prsty „vypnout“ a „zapnout“ jako bity. Sami si zkuste, že na prstech rukou dokážete napočítat do  $2^{10} - 1 = 1023$ .

<sup>6</sup><https://www.topcoder.com/community/data-science/data-science-tutorials/representation-of-integers-and-reals-section-2>

došlo, když 0,1 i 0,2 jsou přesné tvary? Potíž je v tom, že jde o přesné tvary v desítkové soustavě, počítač ale pracuje ve dvojkové soustavě a v ní mají obě čísla nekonečný periodický rozvoj. Tento příklad demonstruje, že nemá smysl zjišťovat, zda se dvě reálná čísla přesně rovnají.

### Typy zaokrouhlovacích chyb

Pokud bychom byli schopni udržet přesnost na 16 platných cifer po celou dobu výpočtu, měli bychom vyhráno. Problém je, že existují operace, které tuto přesnost za určitých podmínek nezachovávají. Nyní se podíváme na nejběžnější typy takových zaokrouhlovacích chyb, konkrétně na cancellation, ordering a smearing.

Pokud se omezíme na základní matematické operace sčítání, násobení, dělení a odčítání dvou kladných čísel, zjistíme, že první tři operace pro libovolně velké operandy přesnost více méně zachovávají.<sup>7</sup> Pokud se ale rozhodneme odečíst dvě blízká čísla, liší se například až na 10. platné cifře, zjistíme, že výsledek bude mít přesnost pouze 5 – 6 platných cifer, neboť pouze o těchto cifrách byla nějaká informace v původních operandech. Tento problém se nazývá *cancellation*. Náš příklad je poněkud přehnaný, nicméně v praxi je cancellation problémem i pro čísla, co jsou jen řádově stejná, neboť se vzniklá chyba bude pravděpodobně s přibývajícimi operacemi dále zvětšovat. Cancellation lze odstranit pouze tak, že problematickou operaci přepíšeme, aby neobsahovala odčítání blízkých čísel. Pro ilustraci cancellation si vezmeme kvadratickou rovnici  $x^2 + 10^6x + 2 = 0$ . Pokud pro nalezení kořenů použijeme klasický vzorec

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

pak, protože  $a$  i  $c$  jsou malá v porovnání s  $b$ , nastane pro jeden z kořenů cancellation při odečítání  $-b$  a diskriminantu. Protože pro kvadratickou rovnici platí  $x_1x_2 = c/a$ , můžeme problematický kořen vypočítat jako

$$x_2 = \frac{c}{ax_1}.$$

V původním výpočtu při použití double precision získáme následující kořeny:  $x_1 = -9,999999999799998 \cdot 10^5$  a  $x_2 = -2,0000152289867401 \cdot 10^{-6}$ , zatímco s použitím opravy  $x_2 = -2,000000000040000 \cdot 10^{-6}$ . Výsledky se tedy liší již na šesté platné cifře. Poznamenejme, že jsme se hrozby cancellation zcela nezbavili. Pokud totiž  $b^2 \approx 4ac$ , pak nastane těžko odstranitelná cancellation již při výpočtu diskriminantu.

Dalším častým problémem je *ordering*. Jeho důsledkem je, že u sčítání závisí na pořadí sčítanců. Nejlépe si jej demonstrujeme na příkladu, kde naším úkolem bude spočítat částečný součet prvních  $N$  členů řady

$$\sum_{k=1}^N \frac{1}{k}.$$

---

<sup>7</sup>Nějaká chyba vznikne při každé operaci, v ideálním případě ale bude v řádu strojové přesnosti.



Použijeme  $N = 10^7$  a single precision (datový typ float, strojová přesnost  $10^{-7}$ ).<sup>8</sup> Pokud budeme řadu sčítat klasicky „odpředu“, tedy od nejmenších  $k$ , dostaneme výsledek<sup>9</sup>  $1,5403682709 \cdot 10^1$ . Protože pro  $N \rightarrow \infty$  řada diverguje (její součet je nekonečno), očekávali bychom, že pro  $N = 10^8$  obdržíme větší částečný součet. Realita je ale taková, že od  $N = 10^7$  se součet nijak nezměnil. A změnil by se ani pro libovolně větší  $N$ , obdrželi jsme tedy vlastně „výsledek“, že  $\infty \doteq 15,4!$

Co se vlastně pokazilo? V každém kroku sčítáme dvě čísla, dosavadní součet a další člen posloupnosti. A zatímco součet postupně roste, členy posloupnosti klesají. Při součtu velkého a malého čísla pak z menšího čísla neuložíme celou jeho přesnost, ale jen to, co se vejde do přesnosti většího z čísel. Při jednom součtu by byla tato chyba zanedbatelná,<sup>10</sup> ale jak součet provádíme mnohokrát, chyba narůstá. Řešením je sčítat řadu „odzadu“. Tak postupně roste součet i členy posloupnosti a my v každém kroku sčítáme ne tak rozdílná čísla. Tímto postupem dostaneme pro  $N = 10^7$  součet  $1,4392651558 \cdot 10^1$  a pro  $N = 10^8$  součet  $1,8807918549 \cdot 10^1$ . Pokud použijeme double precision, dostaneme pro  $N = 10^8$  součet  $1,8997896414 \cdot 10^1$ . Je tedy vidět, že ani druhý způsob není zcela ideální, rozhodně je ale přesnější, než způsob první. Chyba v druhém postupu je nejspíš již neodstranitelnou zaokrouhlovací chybou. Pokud vezmeme v úvahu, že v každém kroku může vzniknout chyba  $10^{-8}$  a my provedli  $10^8$  kroků, pak můžeme očekávat chybu až řádu 1. Budiž toto varování proti používání single precision ve vědeckých výpočtech.

Posledním problémem, který zmíníme, je *smearing*. Nastává, pokud sčítáme mnoho kladných a záporných čísel, jejichž součet je malý. Opět si jej ukážeme na příkladu. Funkci  $\exp(x)$  lze zapsat pomocí tzv. Taylorova rozvoje

$$\exp(x) = 1 + x + \frac{x^2}{2} + \dots = \sum_{k=0}^{\infty} \frac{x^k}{k!}.$$

Pokud se pokusíme v double precision spočítat  $\exp(20)$  pomocí prvních 100 členů rozvoje, dostaneme výsledek  $4,85165195409790158 \cdot 10^8$ , což je blízko správné hodnotě  $4,85165195409790277 \cdot 10^8$ . Pokud se ale pokusíme stejnou metodou spočítat  $\exp(-20)$ , dostaneme  $6,13825973860946424 \cdot 10^{-9}$ , což se od správné hodnoty  $2,06115362243855786 \cdot 10^{-9}$  velmi liší. Problém spočívá v tom, že se v rozvoji střídají velké kladné a záporné členy, které se navzájem téměř vyruší, což vede ke ztrátě přesnosti. V tomto konkrétním případě lze využít identity  $\exp(-x) = 1/\exp(x)$ , což vede na  $\exp(-20) = 2,06115362243855828 \cdot 10^{-9}$ . Tento výsledek je již blízko správné hodnotě.

### Stabilita algoritmu a podmíněnost úlohy

Jak jsme viděli, různé algoritmy jsou různě citlivé na chybu vstupu. Této vlastnosti se říká *stabilita algoritmu*. Algoritmus je stručně řečeno<sup>11</sup> stabilní, pokud malá chyba vstupu vždy vyprodukuje pouze malou chybu výstupu. Naším cílem je

<sup>8</sup>Problém by samozřejmě nastal i při použití double precision, ale nebyl by tak výrazný.

<sup>9</sup>Pro názornost je uvedeno více platných cifer, než je strojová přesnost.

<sup>10</sup>Na úrovni přesnosti většího z čísel.

<sup>11</sup>Existují různé druhy stability, nicméně nám postačí tato základní intuitivní představa.

tedy pokud možno nalézt pro náš problém stabilní algoritmus, který jej řeší. Otázka zní, existuje pro každý problém stabilní algoritmus? Odpověď bohužel zní ne. Problémům, pro které existuje stabilní algoritmus, říkáme *dobře podmíněné*, ostatním *špatně podmíněné*. Příkladem dobře podmíněné úlohy je numerická integrace, příkladem špatně podmíněné úlohy řešení diferenciálních rovnic či numerická derivace. I špatně podmíněné úlohy ale potřebujeme občas řešit, jen je potřeba si v takovém případě počínat obzvlášť opatrně, jak uvidíme v dalších dílech seriálu.

### Složitost algoritmu

Dnešní vědecké výpočty mohou běžet i několik týdnů či dokonce měsíců a spotřebují gigabyty operační paměti. Proto je důležité algoritmy navrhovat s ohledem na jejich časovou a paměťovou složitost. My tu sice takto náročné výpočty dělat nebudeme, přesto je dobré si jisté základy osvojit. Zaměříme se přitom především na časovou složitost.

Začněme jednou radou. Pokud budete v praxi řešit jakýkoliv problém, pro který existuje standardní algoritmus, nesnažte se jej implementovat sami, ale vždy použijte nějakou knihovnu pro numerické výpočty. Ušetříte si tím spoustu času a nervů, máte jistotu, že algoritmus je implementovaný správně a navíc výpočet bude rychlejší, protože autoři knihoven použili mnoho různých triků a optimalizací. V tomto seriálu si ale z pedagogických důvodů napíšeme implementace vlastní.

Nyní již k časové složitosti. *Časová složitost* algoritmu je doba jeho běhu v závislosti na velikosti vstupu. Dobu jeho běhu můžeme odhadnout počtem základních operací, které se musejí vykonat. Některé operace<sup>12</sup> jsou pomalejší než jiné, v numerických výpočtech nás ale zajímají především aritmetické operace. Tam platí, že dělení je o chlup pomalejší než násobení a to je o chlup pomalejší než sčítání a odčítání. Vždy ale mějme na mysli, že prioritou je zajistit stabilitu algoritmu, až poté jeho rychlost. Rychle získaný výsledek je nám k ničemu, pokud je špatně.

Učebnicovým příkladem, kde se optimalizace počtu operací může vyplatit, je vyčíslování polynomů. Mějme polynom  $5x^3 + 2x^2 + 8x + 6$ . Pokud jej vyčíslíme pro nějaké  $x$  takto přímočaře, budeme potřebovat  $3 + 2 + 1 = 6$  násobení<sup>13</sup> a 3 sčítání. Také ale můžeme výraz zapsat jako  $6 + x(8 + x(2 + 5x))$ . Tímto způsobem potřebujeme pouze 3 násobení a 3 sčítání. Tomuto způsobu se říká *Hornerovo schéma*.

Většinou nás ale nezajímá přesný počet proběhlých operací, ale pouze trend, jak časová složitost roste pro velký vstup. Tomuto trendu se pak říká *asymptotická časová složitost*. Jako příklad si vezmeme výše zmíněné vyčíslení polynomu. Velikostí vstupu je zde počet členů polynomu. Při tradičním způsobu je počet sčítání  $N - 1$ , kde  $N$  je počet členů polynomu. Počet násobení je ale  $N(N - 1)/2$ . Celkově tedy potřebujeme  $0,5N^2 + 0,5N - 1$  operací, což se pro velká  $N$  chová zhruba jako  $N^2$ . (Asymptotická) časová složitost prvního algoritmu je tedy kvadratická. Pro Hornerovo schéma potřebujeme  $N - 1$  násobení a  $N - 1$  sčítání, dohromady

<sup>12</sup>Především ty, které interagují s něčím mimo náš program, například výpis do souboru, či alokace paměti.

<sup>13</sup>Mocnění celým číslem je v podstatě opakované násobení a také se jej tak většinou vyplatí implementovat. Cvičení pro zájemce: jak rychleji počítat hodně velké mocniny?

tedy  $2N - 2$  operací. Časová složitost je tedy lineární. Pro velké polynomy se nám tedy vždy vyplatí Hornerovo schéma.

Předvedli jsme si, že pro potřeby asymptotické časové složitosti nás zajímá pouze nejrychleji rostoucí člen, a to bez konstanty před ním. To můžeme formálně vyjádřit pomocí tzv. notace velkého  $O$ . Lineární, resp. kvadratickou časovou složitost bychom zapsali jako  $O(N)$ , resp.  $O(N^2)$ . S touto notací se ještě setkáme i mimo kontext časové složitosti. Vždy ale bude znamenat to, že nás zajímá pouze trend a uvádíme tedy pouze nejvíce signifikantní člen bez konstanty.<sup>14</sup>

### Simulace v počítačové fyzice

Fyzika popisuje přírodní jevy za pomoci matematického aparátu. Obvykle je exaktní popis situace příliš složitý a pracujeme pouze s modelem, který zachycuje všechny podstatné vlastnosti studovaného jevu. Pokud není možné tento model v ruce na papíře v rozumném čase vyřešit a jeho další redukce není přípustná, vstupují do hry počítačové simulace. Jako *počítačovou simulaci* budeme tedy nadále označovat program, který na základě matematického modelu počítá veličiny charakteristické pro studovaný systém. Pro konkrétní představu může jít o časovou závislost teploty chladnoucího hrnku čaje nebo o střední počet částic vyzářený radioaktivním izotopem za určitý časový interval.

Existuje nepřehledné množství úloh, které lze řešit za pomoci simulací, a ještě větší množství numerických a statistických metod, které k jejich řešení můžeme použít. My se zde budeme věnovat takovým úlohám, které nevyžadují oborové znalosti z fyziky a pouze základní znalosti programování. Vyhnete se proto například molekulové dynamice, elektromagnetickému vlnění a dalším zajímavým, ale náročným oblastem. Budeme ovšem dbát na to, aby naše modely měly fyzikální motivaci a aplikaci.

### Klasifikace simulací

Než se začneme věnovat konkrétním metodám a modelům, rozdělíme si simulace do několika kategorií.

Simulace můžeme dělit na diskrétní a spojitě. *Spojitá simulace* je obvykle založena na diferenciální rovnici, pomocí níž dokážeme předpovědět stav v systému v libovolném čase. Jednoduchým příkladem nám budiž Newtonova pohybová rovnice

$$\mathbf{F}(t) = m\ddot{\mathbf{x}}(t) = m \frac{\partial^2 \mathbf{x}}{\partial t^2}, \quad (31)$$

kde  $\ddot{\mathbf{x}}$  značí zrychlení hmotného bodu (druhá derivace polohového vektoru  $\mathbf{x}$ ). Při numerickém řešení této rovnice<sup>15</sup> volíme konečné časové kroky, v nichž hledáme polohu bodu, ale nejsme omezeni tím, jak velké časové kroky zvolíme. Omezuje nás pouze přesnost ukládání číselných hodnot v počítači a časový horizont. Oproti tomu *diskrétní simulace* hledají stav v systému pouze v předem pevně daných časových

<sup>14</sup>Detailní čtení: <https://www.topcoder.com/community/data-science/data-science-tutorials/computational-complexity-section-2/>

<sup>15</sup>Řešit Newtonovu rovnici se naučte později v tomto seriálu.

úsecích. Například model predikující počet obyvatel Evropy v následujících letech by mohl být založen na rovnici

$$P_{i+1} = AP_i \left(1 - \frac{P_i}{B}\right). \quad (32)$$

Proměnná  $P_i$  zde označuje počet lidí v roce  $i$ ,  $A$  a  $B$  jsou parametry. Tyto typy rovnic se nazývají diferencní<sup>16</sup> a jsou diskrétní obdobou diferenciálních rovnic. Jelikož v našem světě plyne čas spojitě, jsou diskrétní časové vývoje vždy aproximací, která s sebou přináší ztrátu přesnosti, ale také zrychlení výpočtu.

Pro zvědavé: Rovnice (32) představuje takzvaný logistický růst. Časový vývoj získáme nejsnáze převedením na diferenciální rovnici

$$\frac{dP}{dt} = AP(1 - P/B).$$

Separací proměnných a integrací parciálních zlomků získáme řešení

$$P = \frac{BP_0 e^{At}}{B + P_0(e^{At} - 1)}.$$

Tento model se používá se například v biologii k simulaci vývoje zvířecích populací. Parametr  $A$  pak představuje růstový koeficient a parametr  $B$  koeficient saturace (přemnožení vede k nedostatku potravy a životního prostoru). Zajímavější model získáme přidáním predace. Model kořist-predátor je dán soustavou rovnic

$$\begin{aligned} K_{i+1} &= AK_i - BK_i P_i, \\ P_{i+1} &= CK_i P_i - DP_i. \end{aligned} \quad (33)$$

Kořist je  $K$  (např. králík), predátor  $P$  (např. puma) a  $A, B, C, D$  jsou parametry. K tomuto modelu se vrátíme v pozdějších dílech seriálu.

Diskretizace se netýká pouze času, ale také prostoru a dalších parametrů. Dobrým příkladem je Isingův model, který má význam při simulování magnetických vlastností pevných látek ve vnějším magnetickém poli. V tomto modelu je prostor rozdělen  $n$ -rozměrnou mříží a pro každou z jejích buněk určujeme hodnotu elektronového spinu.<sup>17</sup> Isingův model patří do významné kategorie mřížkových modelů, na něž ještě přijde řeč.

Dále existuje dělení na deterministické a stochastické simulace. *Deterministické simulace* jsou založené na systému matematických pravidel, pomocí kterého dokážeme v každém časovém kroku určit, v jakém stavu se bude systém nacházet v kroku následujícím. Pro případ diskrétního modelu můžeme obecně psát

$$x_{t+1} = f(t, x_t, x_{t-1}, \dots, x_0)$$

kde  $f$  je funkce, která jednoznačně určuje hodnotu veličiny  $x$ . *Stochastická simulace* je založena na náhodě. Máme sice předpis, který udává vývoj systému (jinak bychom nemohli simulovat!), ale tento předpis mluví pouze o pravděpodobnosti,

<sup>16</sup>Obecně můžeme mluvit o rekurentních vztazích, kde každý člen sekvence je popsán pomocí funkce závislé na (ne nutně všech) předchozích členech.

<sup>17</sup>Kvantově-mechanická veličina, která se svými vlastnostmi podobá momentu hybnosti.

se kterou se bude systém v následujícím časovém kroku nacházet v určitém stavu. Příkladem stochastického procesu je sekvence hodů kostkou. Pokud jsme již například v předchozích hodech naházeli v součtu hodnotu 30, tak víme, že v příštím hodu dosáhne celkový součet hodnoty v rozmezí od 31 do 36, nedokážeme však říci, jaká z těchto hodnot to bude. Víme pouze to, že pravděpodobnost nabytí jakékoli jedné z těchto hodnot je  $1/6$  (není-li kostka „cinklá“).

K tomu, abychom správně pochopili stochastické modely, si potřebujeme osvojit základy pravděpodobnosti a statistiky. Než se do toho pustíme, tak dodejme, že simulace lze dělit do dalších skupin. Prudce rozvíjejícím se oborem jsou *kvantové simulace*. Kvantová statistika nebude předmětem tohoto seriálu, proto zmíníme-li někdy v dalších dílech seriálu simulace kvantových systémů, tak pouze jako zajímavost. Není-li simulace kvantová, nazývá se *klasická simulace*.

Ještě se na chvíli zdržme u pojmů kvantová simulace a deterministická simulace. S kvantovou fyzikou je neodmyslitelně spjata Schrödingerova rovnice. Možná vás nyní zarazí, že simulace založená na Schrödingerově rovnici je deterministická. Každý systém popsaný diferenciální rovnicí je deterministický. Platí však, že vlnová funkce, která je výsledkem řešení Schrödingerovy rovnice, nepopisuje jednoznačně stav fyzikálního systému, tj. vztah mezi vlnovou funkcí a pozorovatelnou veličinou je nedeterministický, naše simulace nikoli.

Podobné zmatení může nastat ohledně simulací chaotických systémů – tyto simulace jsou též deterministické. Chaotičnost spočívá ve vysoké citlivosti systému na počátečních hodnotách. Proto pokud pro stejné počáteční podmínky dostaneme jiný výsledek simulace, nejedná se o náhodné chování, ale o nepřesnou počítačovou reprezentaci čísel popisujících stav systému.

Pokud v sobě rovnice popisující časový vývoj systému zahrnuje náhodné chování, jedná se o stochastickou rovnici. Metodami řešení těchto rovnic se zabývá stochastický kalkulus.

### Základní pojmy z pravděpodobnosti a statistiky

Statistika a teorie pravděpodobnosti tvořily významnou část seriálu minulého ročníku. Zde shrneme pouze to, co budeme potřebovat pro pochopení stochastických procesů a pro zpracování výsledků simulací. Cílem je pouze intuitivní pochopení a praktická aplikace – pokud vám bude připadat následující text příliš složitý, zaměřte se pouze na význam výrazů (38) až (41), tj. na pojmy střední hodnota, rozptyl, zákon velkých čísel a statistická definice pravděpodobnosti. Pokud vás naopak pravděpodobnost zajímá více do hloubky, přečtěte si seriál 30. ročníku.<sup>18</sup>

Abychom se neutopili v abstrakci, budeme si pojmy z teorie pravděpodobnosti vysvětlovat na házení šestistěnnou hrací kostkou. Událost, kdy hodíme kostkou a padne konkrétní číslo, nazýváme *elementární jev*. Množina  $\Omega = \{1, 2, 3, 4, 5, 6\}$  se potom nazve *prostor elementárních jevů*. Označme  $\mathcal{A}$  množinu všech neprázdných podmnožin<sup>19</sup> vybraných z  $\Omega$ . Jako *náhodný jev*  $A$  označíme libovolnou množinu

<sup>18</sup>Seriál naleznete na adrese <http://fykos.cz/ulohy/archiv>. Pokud vám ani to nestačí, doporučujeme knihu Zvára, Štěpán: *Pravděpodobnost a matematická statistika*.

<sup>19</sup>Počet těchto podmnožin je  $2^n - 1$ , kde  $n$  je počet prvků množiny  $\Omega$ .

$A \subseteq \mathcal{A}$ . Například náhodný jev „Na kostce padlo sudé číslo“ je reprezentován množinou  $A = \{2, 4, 6\}$ . *Doplňkový jev* k jevu  $A$  je množina  $A^c = \Omega - A$  a v uvedeném příkladu by šlo o náhodný jev „Na kostce padlo liché číslo“,  $A^c = \{1, 3, 5\}$ .

Je zřejmé, že jev „Padlo číslo menší než čtyři“ bude nastávat častěji než jev „Padla šestka“. Abychom tuto skutečnost nějak kvantifikovali, zavádíme *pravděpodobnostní míru*  $P$ . Pravděpodobnostní míra má tři důležité vlastnosti:

- Je nezáporná,  $P(A) \geq 0$ .
- Je normovaná,  $P(\Omega) = 1$ .
- Je aditivní na disjunktích<sup>20</sup> množinách,

$$P(A_1 + A_2 + \dots) = P(A_1) + P(A_2) + \dots$$

Výraz  $P(A)$  označuje *pravděpodobnost*, že nastane jev  $A$ . Teorie nám neříká, jak volit  $\Omega$ ,  $\mathcal{A}$ ,  $P$ . Pro případ házení kostkou bude platit  $P(A) = |A|/|\Omega|$ , kde svislé čáry značí kardinalitu, tedy počet prvků v množině.

Dále zavádíme *podmíněnou pravděpodobnost*<sup>21</sup>

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (34)$$

Čteme: Pravděpodobnost, že za předpokladu nastání jevu  $B$  nastane jev  $A$ , je rovna pravděpodobnosti, že nastanou jevy  $A$  a  $B$  zároveň, dělené pravděpodobností jevu  $B$ . Necht jev  $A$  znamená, že padne sudé číslo, a  $B$  znamená, že padne číslo větší než 3. Potom

$$P(A|B) = P(\{2, 4, 6\}|\{4, 5, 6\}) = \frac{P(\{2, 4, 6\} \cap \{4, 5, 6\})}{P(\{4, 5, 6\})} = \frac{P(\{4, 6\})}{P(\{4, 5, 6\})} = \frac{\frac{2}{6}}{\frac{3}{6}} = \frac{2}{3}.$$

O jevech říkáme, že jsou *nezávislé*, pokud platí  $P(A|B) = P(A)$ .

Mějme množinu neslučitelných jevů  $B_1, B_2, \dots$ , z nichž alespoň jeden nastává (tedy  $B_i \cap B_j = \emptyset \forall i, j$  a  $B_1 \cup B_2 \cup \dots = \Omega$ ). Pak *Věta o úplné pravděpodobnosti* tvrdí

$$P(A) = \sum_i P(A|B_i)P(B_i). \quad (35)$$

Toto tvrzení vám může pomoci při řešení jedné z bonusových úloh.

Označme nyní prvky množiny  $\Omega$  malým písmenem  $\omega$ . Necht existuje funkce  $X$ , která každému prvku  $\omega$  přiřadí reálné číslo, formálně  $X : \Omega \rightarrow \mathbb{R}$ . Funkce  $X$  se nazývá *náhodná veličina* a číslo  $X(\omega)$  je *realizace náhodné veličiny*. Význam náhodné veličiny si opět ilustrujeme na hodů kostkami. Nyní budeme mít dvě šestistěnné kostky a množina  $\Omega$  bude tedy obsahovat celkem 36 prvků:

$$(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2), (2, 3), \dots, (6, 6).$$

<sup>20</sup>Disjunkttní množiny nemají žádný společný prvek, tj. jejich průnikem je prázdná množina.

<sup>21</sup>Každou pravděpodobnost můžeme chápat jako podmíněnou, protože  $P(A|\Omega) = P(A)$ .

Jednotlivé prvky množiny  $\Omega$  značme  $\omega_i$  a jejich složky  $\omega_i^j$ . Tedy například  $\omega_9^1 = 2$ ,  $\omega_9^2 = 3$ . Ve většině stolních her nás zajímá součet ok na hozených kostkách. Tento součet představuje náhodnou veličinu  $X$  definovanou vztahem

$$X(\omega_i) = \omega_i^1 + \omega_i^2 \quad \forall i = 1, \dots, 36.$$

Pro každé  $x \in \mathbb{R}$  definujeme *distribuční funkci*  $F(x) = P(\{\omega : X(\omega) \leq x\})$ . V našem příkladu se jedná o diskrétní úlohu, náhodná veličina  $X$  tedy nabývá hodnot  $x_1 = 2, x_2 = 3, \dots, x_{11} = 12$  a píšeme<sup>22</sup>

$$P(\{\omega : X(\omega) = x_i\}) = p_i \equiv P(x_i). \quad (36)$$

Funkce  $P(x)$  se nazývá *pravděpodobnostní funkce*. Pokud bychom pracovali se spojitou náhodnou veličinou, definovali bychom distribuční funkci pomocí integrálu

$$F(x) = \int_{-\infty}^x f(t)dt, \quad (37)$$

kde se  $f(t)$  nazývá *hustota pravděpodobnosti* a je spojitou obdobou pravděpodobnostní funkce.

Náhodnou veličinu lze charakterizovat pomocí mnoha různých ukazatelů, zde si uvedeme pouze dva. *Střední hodnota* se definuje pro diskrétní a spojitá rozdělení jako

$$EX = \sum_i x_i p_i \equiv \mu, \quad EX = \int_{-\infty}^{\infty} x f(x)dx \quad (38)$$

a *rozptyl* jako

$$\text{Var}X = \sum_i (x_i - \mu)^2 p_i = E(X - \mu)^2 \equiv \sigma^2, \quad \text{Var}X = \int_{-\infty}^{\infty} (x - \mu)^2 f(x)dx \quad (39)$$

přičemž odmocnina rozptylu  $\sigma$  se nazývá *směrodatná odchylka*. Laicky řečeno rozptyl udává, jak moc široce jsou  $x_i$  rozprostřeny okolo střední hodnoty.

Význam střední hodnoty nejlépe pochopíme pomocí tzv. *zákona velkých čísel*. Ten nám říká, že aritmetický průměr výsledků získaných měření náhodné veličiny  $X$  se s rostoucím počtem pokusů  $N$  blíží střední hodnotě  $EX$ . Formálně zapsáno

$$E \lim_{N \rightarrow \infty} \left( \frac{1}{N} \sum_i^N X_i \right) = \mu. \quad (40)$$

Tento zákon nám zaručuje, že pro dostatečně velký počet pokusů získáme dobrý odhad střední hodnoty, což je nezbytný předpoklad pro simulace i fyzikální experimenty.

V této kapitole jsme prozatím vycházeli z tzv. Kolmogorovy axiomatické definice pravděpodobnosti, která je, jak jsme si ukázali, velmi obecná a neříká nám, jak pro účely simulace definovat  $\Omega$ ,  $\mathcal{A}$  a  $P$ . Proto v praxi používáme *statistickou*

<sup>22</sup>Často se také používá zkrácený zápis  $P(X = x_i)$ .

definici pravděpodobnosti, podle níž je pravděpodobnost jevu dána jeho *relativní četností* při dostatečném počtu pokusů. Tedy pokud v  $N$  pokusech nastane  $M$ -krát jev  $A$ , pak platí

$$\lim_{N \rightarrow \infty} \frac{M}{N} = P(A). \quad (41)$$

Nakonec si uveďme dvě důležitá diskrétní rozdělení:

- *Rovnoměrné rozdělení* je takové rozdělení, v němž náhodná veličina může nabývat  $n$  hodnot, a to s pravděpodobností  $1/n$  pro každou z nich. Pro pohodlnost popisu řekněme, že množinu realizací náhodné veličiny tvoří všechna celá čísla z intervalu  $\langle a, b \rangle$ , kde  $a, b$  jsou také celá, tedy  $n = b - a + 1$ . Pravděpodobnostní funkci potom můžeme zapsat jako

$$P(k) = \begin{cases} \frac{1}{n} & \forall k \in \{\mathbb{Z} \cap \langle a, b \rangle\} \\ 0 & \text{jinak} \end{cases} \quad (42)$$

a distribuční funkci jako

$$F(k) = \frac{\lfloor k \rfloor - a + 1}{n} \quad \forall k \in \langle a, b \rangle. \quad (43)$$

Příkladem náhodné veličiny s rovnoměrným rozdělením je hod jednou šestistěnnou kostkou. Střední hodnota rovnoměrného rozdělení je  $\mu = (a + b)/2$ , rozptyl  $\sigma^2 = (n^2 - 1)/12$ .

- *Binomické rozdělení* s parametry  $n, p$  je dáno pravděpodobnostní funkcí

$$P(k) = \binom{n}{k} p^k (1 - p)^{n-k}. \quad (44)$$

Parametr  $p \in \langle 0, 1 \rangle$  zde představuje pravděpodobnost, že v pokusu s binárním výsledkem 0 nebo 1 dostaneme číslo 1 (a číslo 0 pak s pravděpodobností  $1 - p$ ). Parametr  $n$  udává, kolikrát jsme pokus provedli. Náhodná veličina  $X$  s binomickým rozdělením  $P(k)$  potom udává počet úspěchů (hodnota 1) v těchto  $n$  pokusech. Typickým příkladem jsou opakované hody mincí, kde např. 1 je panna a 0 orel. Střední hodnota binomického rozdělení je  $\mu = np$ , rozptyl  $\sigma^2 = np(1 - p)$ .

### Generátory pseudonáhodných čísel

Představme si, že žijeme v sedmáctém století a přijde za námi Jacob Bernoulli a zeptá se nás, s jakou pravděpodobností nám při hodu šesti šestistěnnými kostkami padne číslo dvacet. Nyní máme na výběr: buďto jsme dobří počtáři a nalezneme všechny kombinace, které vedou na tuto hodnotu, nebo jsme hodně trpěliví a budeme kostkami házet tak dlouho, dokud nedostaneme výsledek s uspokojivou přesností. V dnešní době máme naštěstí počítače a ty mohou kostkou házet za nás. Je tu ovšem jeden problém: jak řekneme počítači, aby při každém hodu kostkou *náhodně* vybral, jaká hodnota padne? Při hodu skutečnou kostkou spočívá náhodnost ve vysoké citlivosti výsledku na rychlosti rotace kostky a úhlu dopadu (kostka se chová chaoticky). Co použijeme jako zdroj náhody v počítači?



Vytvářet náhodná čísla za pomoci počítače můžeme dvěma způsoby. První z nich využívá fyzikální procesy, například tepelný šum v hardwarových součástkách nebo kvantové jevy. Takto získaná náhodná čísla se v angličtině označují jako *true random*, opravdu náhodná (jejich hodnotu nedokážeme předpovědět o nic lépe než výsledek hodu kostkou). Nevýhodou těchto generátorů je, že nedokáží poskytnout dostatečně velké množství náhodných čísel v krátkém čase nebo vyžadují specializovaný hardware.

Obvyklejší je proto používat *generátory pseudonáhodných čísel*. Ty jsou založeny na jistém deterministickém algoritmu, který na základě zadané počáteční hodnoty (tzv. *seed*, může být získán pomocí hardwarového generátoru) generuje sekvenci čísel, která má statistické vlastnosti blízké sekvenci opravdových náhodných čísel. Existuje mnoho testů náhodnosti a zde se jimi nebudeme příliš dopodrobna zabývat. Pro úplnost uvedme, že se může jednat o testování četnosti výskytu jednotlivých čísel a jejich řetězců či rozdělení délky mezer mezi stejnými hodnotami. Jednoduchým vizuálním testem je vykreslení náhodných čísel pomocí černobílého kódování do plochy. Výsledek by měl připomínat šum analogové televize.<sup>23</sup>

Nejjednodušším generátorem pseudonáhodných čísel (a jediným, který si podrobně popíšeme) je *lineární kongruenční generátor*.<sup>24</sup> Tento generátor vytváří sekvenci čísel  $\{x_n\}$  na základě rekurentního vztahu

$$x_{n+1} = (ax_n + c) \bmod m, \quad (45)$$

kde parametry  $a$ ,  $c$  jsou celá čísla,  $m$  je přirozené číslo a  $\bmod$  je operátor udávající zbytek po dělení výrazu číslem  $m$ .  $x_0$  je seed, který musí zadat uživatel. Parametry volíme tak, abychom získali co nejdělsí neopakující se sekvenci čísel pro libovolný seed. Je zřejmé, že perioda nemůže být delší než  $m$ . Pokud chceme, aby generátor měl tuto délku periody pro všechny seedy, musí parametry splňovat následující podmínky:

- Čísla  $m$  a  $c$  jsou nesoudělná.
- $a - 1$  je dělitelné všemi prvočísly z rozkladu čísla  $m$ .
- Pokud je  $m$  dělitelné 4, je také  $a - 1$  dělitelné čtyřmi.

Fungování generátoru si ukážeme na dvou jednoduchých příkladech. Nejprve volme  $a = 6$ ,  $c = 1$ ,  $m = 13$ . Tato volba zřejmě nesplňuje druhé kritérium. Pro  $x_0 = 0$  dostaneme sekvenci

$$1, 7, 4, 12, 8, 10, 9, 3, 6, 11, 2, 0, \dots$$

Vidíme, že tato sekvence má periodu  $m - 1 = 12$ , přičemž v ní chybí číslo 5. Podobný výsledek dostaneme pro každý seed kromě  $x_0 = 5$ , kdy sekvence obsahuje pouze číslo 5. Volba parametrů  $a = 13$ ,  $c = 1$ ,  $m = 6$  splňuje všechna kritéria a sami si jistě snadno ověříte, že pro libovolný seed je perioda maximální. Má ovšem délku pouze  $m = 6$ . Pokud bychom zvolili  $c = 0$ , hovořili bychom o *multiplikačním*

<sup>23</sup>Náhodnost generátoru ve vašem internetovém prohlížeči si můžete pomoci vizuálního testu ověřit na adrese <http://www.psychicscience.org/vt2.aspx>.

<sup>24</sup>Název pochází z matematického pojmu kongruence zbytkových tříd. Říkáme, že  $a$  je kongruentní s  $b$  modulo  $n$ , pokud rozdíl  $a - b$  je beze zbytku dělitelný číslem  $n$ .

kongruenčním generátoru. Výpočet čísel v sekvenci je poté jednodušší, ale nesmíme volit  $x_0 = 0$ .

Nejvyšší efektivity kongruenční generátory dosahují, pokud je  $m$  mocnina dvou, např.  $2^{32}$ . Poté totiž v binární soustavě získáme zbytek po dělení prostě tak, že od 31. pozice všechny hodnoty nalevo<sup>25</sup> vynulujeme (v praxi je prostě vůbec nepočítáme). Většina implementací však nevrací čísla v rozsahu od 0 do  $m - 1$ , ale dělí je hodnotou  $m$ , čímž je přeskakuje do intervalu  $(0, 1)$ . To je velmi výhodné zvláště tehdy, když generátor používáme k simulaci stochastického chování, protože hodnoty pravděpodobnosti jsou také v rozsahu od 0 do 1.

Pečlivě musíme volit i čísla  $a$  a  $c$ . Výše uvedené podmínky, které na ně klade, zaručují pouze dlouhou periodu, ne její náhodnost. Pro volbu  $a = 1$ ,  $c = 1$ ,  $m = 2^{31}$ ,  $x_0 = 0$ , která podmínky splňuje, dostaneme sekvenci 1, 2, 3, 4, ... I bez diskuse je jasné, že nejde o náhodná čísla. Kongruenční generátory, které se v praxi používají, mají pečlivě zvolené hodnoty parametrů a prošly mnoha testy náhodnosti.

Před použitím generátoru pseudonáhodných čísel ve vašem oblíbeném programovacím jazyku si nejprve zkontrolujte, jakou metodu vlastně používáte. Například funkce `rand()` z knihovny `glibc` pro jazyk C je obyčejný lineární kongruenční generátor, který může být pro některé aplikace nedostačující. Python používá ve své funkci `random()` generátor založený na algoritmu Mersenne Twister, který využívá vlastnosti Mersennových prvočísel. Seed se získává ze systémového času.

Na závěr ještě zmiňme, že existují také *generátory kvazináhodných čísel*, jejichž cílem není přiblížit se co nejvíce opravdovým náhodným číslům, ale spíše co nejrovnoměrněji pokrýt zvolenou oblast  $n$ -dimenzionálního prostoru. Příkladem jsou například Sobolovy sekvence. Kvazináhodná čísla se s výhodou používají například v metodě Monte Carlo – o té si ale povíme více až v příštím díle.

## Drsný úvod do programování

Program je sekvence příkazů, které má počítač vykonat. Aby je mohl vykonat, musí jim rozumět. Počítač ovšem rozumí pouze sekvenci jedniček a nul, tzv. strojovému kódu (posloupnost instrukcí pro procesor). Dříve se v něm skutečně programy psaly, nicméně bylo to pro programátora nepohodlné, navíc každá generace počítačů rozuměla jinému strojovému kódu. Proto byly vynalezeny programovací jazyky, které jsou bližší lidskému uvažování a řeči. Dnes tedy programátor napíše pomocí nějakého programovacího jazyka tzv. zdrojový kód. Ten se poté, v závislosti na zvoleném jazyce, buď přímo za běhu interpretuje počítači, či je nejprve přeložen pomocí překladače (kompilátoru) do strojového kódu.

## Volba jazyka

Jednotlivé úlohy seriálu můžete řešit v libovolném jazyce, vždy je ale dobré k takovému řešení přiložit zdrojový kód. My budeme v ukázkách používat buď programovací jazyk Python, nebo tzv. pseudokód. Pseudokód je v podstatě zdrojový kód, kde jsou ale některé pasáže z důvodu stručnosti a přehlednosti popsány pouze

<sup>25</sup>Nebo napravo. Záleží na tom, jak ukládáme data do paměti. Více viz heslo *endianita*.

slovně, nikoliv v daném jazyce. Takový kód samozřejmě není přímo spustitelný na počítači.

Jazyk Python byl zvolen, protože jde o poměrně jednoduchý a přehledný jazyk, navíc dnes dominuje v oblasti data science. Pravděpodobně se s ním v praxi tedy setkáte. Chtěli bychom dopředu upozornit, že styl, jakým budeme zezačátku v Pythonu psát, není pro vědecké výpočty v Pythonu běžný. V praxi je hojně využívána knihovna NumPy<sup>26</sup> s knihovnou pro vědecké výpočty SciPy<sup>27</sup> a pokročilejší funkcionální konstrukce jazyka. Správné použití těchto prostředků dokáže zrychlit výpočty, nicméně podmínkou je znalost toho, jak je v Pythonu „pod kapotou“ nakládáno s operační pamětí. My si v prvních ukázkách vystačíme bez těchto knihoven, náš zdrojový kód tedy nebude zcela optimalizovaný, zato bude intuitivní a přehledný i pro ty, kteří nemají s Pythonem zkušenost. Později ovšem z důvodu stručnosti začneme knihovnu NumPy v ukázkách používat tam, kde to bude vhodné.

Python je interpretovaný jazyk, to znamená, že zdrojový kód nepřekládáme, ale rovnou spouštíme pomocí tzv. interpreteru. Interpreter Pythonu a kompletní dokumentaci jazyka i s tutoriály pro začátečníky najdete na <https://www.python.org>. Python je dnes používán ve dvou vzájemně nekompatibilních verzích 2 a 3. V našich ukázkách budeme používat novější verzi 3. Protože zdrojový kód je čistý text, potřebujeme kromě interpreteru již pouze libovolný textový editor, třeba Poznámkový blok<sup>28</sup> a můžeme začít programovat. Naše programy budou poměrně krátké, nezatažujte se tedy výběrem nějakých pokročilejších vývojových prostředí (IDE). Nevyužijeme jejich plný potenciál a pouze by nás mátlly.

### Základy syntaxe jazyka Python

Jak již bylo řečeno, nebudeme zde suplovat tutoriály, kterých lze na internetu nalézt spoustu. Místo toho si rovnou předvedeme ukázkou zdrojového kódu a rozebereme si, co tento kód dělá.

```
import math

def e_z_limity(n):
    if n < 1:
        print("My brain hurts")
        exit()
    vnitrek = 1.0+1.0/n
    vysledek = 1.0
    for i in range(n):
        vysledek *= vnitrek
    return vysledek
```

<sup>26</sup><http://www.numpy.org>

<sup>27</sup><https://www.scipy.org/scipylib/index.html>

<sup>28</sup>V nejhodnějším případě. Lepší je nainstalovat si kvalitní editor, např. Sublime Text, nebo použít jeden z editorů nainstalovaných v Linuxu.

```

def e_z_taylora(n):
    suma = 0.0
    faktorial = 1.0
    for i in range(1,n+2):
        suma += 1.0/faktorial
        faktorial *= i
    return suma

for i in range(1,10000):
    print("{} {} {}".format(i,math.e-e_z_limity(i),math.e-e_z_taylora(i)))

```

Program je vykonáván shora dolů po jednotlivých příkazech. Příkaz `import math` zajistí import matematické knihovny. V našem programu ji potřebujeme kvůli konstantě `math.e`, což je Eulerovo číslo. Dále máme prázdný řádek. Ten je uveden pouze pro přehlednost a je počítačem ignorován. Stejně tak jsou ignorovány řádky začínající znakem `#`. Jde o tzv. komentáře a slouží pouze pro snadnější orientaci v kódu. Nicméně doporučujeme naučit se je používat.

Další řádek začíná klíčovým slovem `def`. Jde o definici funkce, kterou jsme nazvali `e_z_limity()`. Tato funkce má jeden parametr `n`. Funkce je označení kusu kódu, tzv. těla funkce, který se provede, pokud tuto funkci zavoláme. Volání funkce se dělá uvedením názvu funkce s kulatými závorkami, které mohou obsahovat hodnoty parametrů. Pokud je parametrů více, jsou odděleny čárkami, jak vidíme například při volání funkcí `range()` nebo `format()`. Tělo funkce<sup>29</sup> je označeno od sazáním, tělo funkce `e_z_limity()` tedy končí řádkem `return vysledek`. Příkaz `return` znamená ukončení vykonávání funkce a případné vrácení nějaké hodnoty. Funkce `e_z_limity()` například vrací hodnotu uloženou v proměnné `vysledek`.

Na dalším řádku vidíme konstrukci `if`. Pokud je splněna podmínka, v našem případě `n < 1`, provede se blok příkazů za ní, v našem případě příkazy `print("My brain hurts")` a `exit()`. Pokud podmínka splněna není, blok se přeskočí. Konstrukce `if` může obsahovat i blok `else`, který se provede, pokud podmínka splněna není. Podmínkou může být cokoliv, co vrací hodnoty `True` (pravda) a `False` (nepravda).

Příkaz `print("My brain hurts")` vypíše na obrazovku řádek s textem „My brain hurts“ (bez uvozovek). Uvozovky zde označují, že jde o tzv. řetězec znaků (string). Příkaz `exit()` pak ukončí program. Další dva příkazy vytvoří proměnné pojmenované `vysledek` a `vnitrek` a přiřadí jim hodnotu 1, resp.  $1 + 1/n$ , kde  $n$  je aktuální hodnota parametru `n`. Proměnné jsou zjednodušeně řečeno „škatulky“, kam si můžete ukládat nějakou hodnotu, ať už jde o číslo, řetězec, ... K proměnným můžeme přistupovat vždy pouze v rámci bloku, ve kterém byly definovány, v tomto případě jde o tělo funkce `e_z_limity()`. Povšimněte si, že přiřazujeme hodnotu „1.0“ a ne „1“. Rozdíl je v tom, že „1“ je celé číslo, zatímco „1.0“ je reálné číslo (v double precision). Jak později uvidíme, z kontextu výpočtu chceme, aby `vysledek` a `vnitrek` byly reálné.<sup>30</sup>

<sup>29</sup> Stejně jako každý blok kódu v Pythonu.

<sup>30</sup> Ve skutečnosti by stačilo psát „1“. Důvodem jsou hlubší nuance jazyka Python, který v konkrétních situacích konvertuje celá čísla na reálná nebo naopak. Python 2 se v tom chová

Další řádek začíná klíčovým slovem `for`. Jde o tzv. cyklus, tedy blok příkazů, který se dokola opakuje. Ve většině jazyků je opakování prováděno, dokud je splněna podmínka, podobně jako u konstrukce `if`. Python je v tomto odlišný, cyklus je prováděn pro všechny prvky seznamu, či obecně libovolné kolekce hodnot, kterou lze postupně procházet. V našem případě funkce `range(n)` vrátí seznam celých čísel od 0 do  $n - 1$  včetně. Cyklus je pak v tomto pořadí postupně prochází, dosadí danou hodnotu do proměnné `i` a vykoná blok příkazů, v našem případě příkaz `vysledek *= vnitrek`. Ten vynásobí hodnoty proměnných `vysledek` a `vnitrek` a výsledek uloží znovu do proměnné `vysledek`. Jde pouze o zkrácení zápisu `vysledek = vysledek * vnitrek`. Poznamenejme, že v tomto cyklu proměnnou `i` vůbec nevyužíváme, celý cyklus slouží pouze k tomu, aby příkaz `vysledek *= vnitrek` proběhl  $n$ -krát za sebou. Cyklus tedy představuje pouhý výpočet  $n$ -té mocniny hodnoty proměnné `vnitrek`.

Poslední příkaz těla funkce je příkaz `return vysledek`. Kdykoliv se při vykonávání funkce narazí na příkaz `return`, nebo se dojde na konec těla funkce, vykonávání funkce se ukončí a program pokračuje v místě, odkud byla funkce zavolána. Pokud je za příkazem `return` uvedena nějaká konstanta nebo proměnná, je její hodnota dosazena na místo volání funkce. Říkáme, že funkce vrátila hodnotu. Příkladem mohou být různé matematické funkce, například `math.cos()`, která vrátí kosinus parametru, či funkce `range()`, která vrací seznam čísel.

Pokud budeme program sledovat dále, vidíme, že následuje definice funkce `e_z_taylora()`. Jejímú tělu byste již měli rozumět, za povšimnutí stojí pouze volání `range()` se dvěma parametry. V tomto případě funkce vrátí seznam celých čísel od 1 do  $n + 1$  včetně (hodnota  $n + 2$  už v seznamu není).

Následuje poslední cyklus `for i in range(1,10000)`. Protože se doposud vyskytovaly pouze definice funkcí a příkaz `import`, je toto místem, kde se skutečně začne vykonávat program. S funkcí `print()` jsme se již setkali, cílem tohoto cyklu je tedy 9999krát něco vypsat na obrazovku. Povšimněme si konstrukce `"{} {}".format(. . .)`. Ta vytvoří řetězec, dle vzoru v uvozovkách, ale místo každé dvojice znaků `{}` dosadí v pořadí argumenty funkce `format()`. Ta může mít libovolný počet argumentů. Cílem je tedy tisknout hodnoty `i`, `math.e-e_z_limity(i)` a `math.e-e_z_taylora(i)` oddělené mezerou, a to pro `i` nabývající postupně hodnot 1 až 9999.

Seznámili jsme se s konstrukcemi `if` a `for`, definicí funkce, proměnnými a výpisem na obrazovku. Rozhodně nejde o plný výčet konstrukcí jazyka Python, nicméně se jedná o ty nejpoužívanější a nejspíše si s nimi vystačíme po velkou část seriálu. Jsme si vědomi, že šlo opravdu o rychlokurz, proto silně doporučujeme projít si nějaký tutoriál a pokusit se napsat si pár cvičných programů. Strávíte tím sice několik večerů, nicméně základní znalost programování vám brzy vloženou námahu mnohokrát vrátí.

---

jinak než Python 3. Jestli neumíte dobře programovat, raději se na to nespolehejte.

*And now for something completely different*

Nyní, když zvládneme zdrojový kód přečíst, se podívejme na to, co má tento kód za úkol. Jak nám napoví názvy funkcí, patrně se snažíme vypočítat Eulerovo číslo různými způsoby, vypisujeme pak rozdíl těchto výpočtů od skutečné hodnoty.

Funkce `e_z_limity()` využívá vztahu

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n,$$

kdy vyčíslujeme výraz

$$\left(1 + \frac{1}{n}\right)^n$$

pro různě velký parametr  $n$ . Pro dostatečně velké  $n$  bychom měli přibližně dostat hodnotu  $e$ . Podmínka zde testuje, jestli hodnota parametru dává smysl. Je zde uvedena zcela účelově pouze pro ukázkou konstrukce `if`. Všimněme si, že není zcela neprůstředná, mohli bychom totiž v parametru předat například řetězec, či reálné číslo. Při psaní programů je obecně dobrým návykem kontrolovat, jestli je vstup validní, nicméně v našich prográmcích se tímto nebudeme většinou zabývat.

Funkce `e_z_taylora()` využívá Taylorova rozvoje exponenciály, který byl zmíněn v kapitole o smearingu. Pokud tento rozvoj vyčíslíme v bodě 1, dostaneme

$$\exp(1) = e = \sum_{k=0}^{\infty} \frac{1}{k!}.$$

Funkce pak skutečně přímočaře vyčísluje částečné součty rozvoje

$$\sum_{k=0}^n \frac{1}{k!}$$

v závislosti na hodnotě parametru  $n$  vyjadřujícím počet členů rozvoje. Mohli byste v tuto chvíli namítnout, že použitý postup bude náchylný k orderingu. Nicméně pokud program skutečně spustíme, zjistíme, že se pro dostatečně velké  $n$  liší výsledek výpočtu od skutečné hodnoty Eulerova čísla v řádu  $10^{-16}$ , tedy v řádu strojové přesnosti. Ordering je tedy v tomto případě zanedbatelný (nejspíš díky tomu, že  $1/k!$  klesá děsně prudce) a nebudeme se jím zabývat.

*Spuštění programu*

Uložme zdrojový kód do textového souboru, například `euler.py`. Poté spustíme příkazový řádek v adresáři, kam jsme uložili náš program. Ten nyní spustíme příkazem `python euler.py`. Uvidíme očekávaný výpis čísel. Program chvíli poběží, neboť výpis na obrazovku je poměrně pomalá operace, zvláště když ji provádíme téměř desettisíckrát<sup>31</sup>. Gratulujeme, právě jste spustili svůj první program. Pokud

<sup>31</sup>Problém není velikost výstupu, ale to, že program musí volat systémovou funkci pro výpis z programu. Výpis tedy můžeme zrychlit tím, že si výstup nejdřív vytvoříme jako řetězec a ten pak vypíšeme jednou. V Pythonu existují alternativní způsoby čtení a výpisu dat, které to opravdu dělají a jsou tak o hodně rychlejší

se vám program nedaří spustit, neváhejte se na nás obrátit, případně postupujte podle některého z tutoriálů pro instalaci a první spuštění pythonu.

Mít výpis čísel pouze na obrazovce je nepraktické. Proto znovu spustíme program, tentokrát příkazem `python euler.py > euler-data.txt`. Tím jsme přesměrovali výstup z programu do souboru `euler-data.txt`, který se vytvořil ve stejném adresáři, jako `euler.py`. Takto uložený výstup je již jednoduché dále používat, například jej můžeme zobrazit v grafu pomocí programu Gnuplot nebo importovat do Excelu či jiného tabulkového editoru.

Pokud si výstup prohlédneme, zjistíme, že nám stačí pouze 17 členů Taylorova rozvoje pro dosažení strojové přesnosti. Oproti tomu i pro  $n = 9999$  je výsledek funkce `e_z_limity()` přesný pouze na několik málo platných cifer. Nicméně tyto dvě metody nemůžeme zcela srovnávat, neboť jde v obou případech o zcela jiný postup a parametr `n` má v obou případech různý význam.

## Úloha I.S ... rozjezdová

10 bodů

- Upravte výraz  $\sqrt{x+1} - \sqrt{x}$  tak, aby nebyl náchylný k problémům cancellation, ordering a smearing. Ke kterým z těchto problémů byl původně náchylný a proč? Jaký je rozdíl ve výsledku původního a opraveného výrazu, pokud jej vyčíslíme v double precision pro  $x = 1,0 \cdot 10^{10}$ ?
- Popište funkci následujícího kódu. Jaký je rozdíl mezi funkcemi `a()` a `b()`? Pro jaké hodnoty `x` je lze použít? Nebojte se kód spustit a hrát si s hodnotou proměnné `x`. Určete také asymptotickou časovou složitost programu v závislosti na proměnné `x`.

```
def a(n):
    if n == 0:
        return 1
    else:
        return n*a(n-1)
def b(n):
    if n == 0:
        return 1.0
    else:
        return n*b(n-1)
```

```
x=10
print("{} {} {}".format(x, a(x), b(x)))
```

- Označme  $o_k$  a  $O_k$  obvod vepsaného a opsaného pravidelného  $k$ -úhelníku ke kružnici. Pak pro ně platí rekurentní vztahy

$$O_{2k} = \frac{2o_k O_k}{o_k + O_k}, \quad o_{2k} = \sqrt{o_k O_{2k}}.$$

Napište program, který pomocí těchto vztahů vypočítá hodnotu  $\pi$ , začněte přitom s opsaným a vepsaným čtvercem. S jakou přesností dokážete  $\pi$  takto aproximovat? Obdobu tohoto postupu původně navrhl a použil Archimedes.

d) Lukáš a Mirek hrají hru. Házejí férovou mincí a když padne orel, dá Mirek Lukášovi jedno FYKOSí tričko, když padne panna, dá jedno tričko Lukáš Mirkovi. Oba dohromady mají  $t$  triček, z toho  $l$  patří Lukášovi a  $m$  Mirkovi. Pokud jednomu z hráčů dojdou trička, hra končí.

1. Necht  $m = 3$  a Lukášova zásoba triček je nekonečná. Určete *nejpravděpodobnější* dobu trvání hry, tedy počet hodů mincí, po nichž hra skončí (protože Mirkovi dojdou trička).
2. Necht  $m = 10$ ,  $l = 20$ . Proveďte simulaci pomocí generátoru pseudonáhodných čísel a nalezněte pravděpodobnost, že Mirek vyhraje všechna Lukášova trička. Celou hru nechejte proběhnout alespoň 100krát (čím více opakování, tím lépe).
3. Jak se změní výsledek předchozí úlohy, jestliže Mirek minci „vylepší“ a panna nyní padá s pravděpodobností  $5/9$ ?

*Bonus* Vypočtete pravděpodobnosti analyticky a porovnejte výsledek se simulací.

- e) Mějme lineární kongruenční generátor s parametry  $a = 65\,539$ ,  $m = 2^{31}$ ,  $c = 0$ .
1. Vygenerujte alespoň 1 000 čísel a spočtete jejich střední hodnotu a rozptyl. Porovnejte se střední hodnotou a rozptylem rovnoměrného rozdělení na stejném intervalu.
  2. Nalezněte vztah, který vyjádří číslo v generované sekvenci jako lineární kombinaci čísel na dvou předchozích pozicích, tj. nalezněte koeficienty  $A$ ,  $B$  v rekurentním vztahu  $x_{k+2} = Ax_{k+1} + Bx_k$ . Pokud budeme považovat každá tři po sobě následující čísla za souřadnice bodu ve trojrozměrném prostoru, jak rekurentní vztah ovlivní prostorové rozložení těchto bodů?

*Bonus* Vygenerujte sekvenci alespoň 10 000 čísel a vykreslete 3D bodový graf, který ilustruje význam uvedeného rekurentního vztahu.

(řešení str. 230)

## Kapitola 2: Derivace a Monte Carlo integrace

V minulém díle jste se seznámili s reprezentací čísel v počítači a s chybami, které mohou plynout z nepřesnosti datového typu s plovoucí desetinnou čárkou, a naučili jste se, jak tyto chyby minimalizovat či jim zcela předcházet. Také již umíte určovat časovou složitost počítačových algoritmů. Tyto znalosti využijete při numerickém výpočtu derivací, což je téma tohoto dílu seriálu.

V simulační části seriálu po předchozím náročném dílu trochu zvolníme a podíváme se na již slíbenou stochastickou metodu Monte Carlo, přičemž si zopakujeme a prohloubíme znalosti týkající se statistického zpracování výsledků simulací. Ke slovu zde opět přijdou generátory pseudonáhodných čísel.

### Numerická derivace

Protože nepředpokládáme čtenářovu znalost pojmu derivace, pokusme se jej nejprve stručně nastínit. Derivace je operace, která zapůsobí na funkci  $f$  a vrátí jinou funkci – derivaci původní funkce, značíme  $f'$ . Pokud derivaci provedeme pro kon-



krétní hodnotu  $x$ , dostaneme tzv. *derivaci v bodě*  $f'(x)$ , tedy číslo. Při numerických výpočtech máme k dispozici pouze čísla, nikoliv spojité funkce, numerickou derivací tedy budeme chápat právě jako derivaci v bodě.

Derivace funkce  $f$  v bodě  $x$  je definována vztahem

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

Odtud je hned patrný význam derivace, tedy že derivace je podíl změny funkční hodnoty  $f$  a příslušné změny  $x$  za předpokladu, že je tato změna velmi malá (ve skutečnosti nekonečně malá). To má řadu praktických analogií, například průměrná rychlost je podíl změny polohy a příslušného přírůstku času. Pokud bude přírůstek času velmi malý, přejde průměrná rychlost v rychlost okamžitou. Můžeme tedy říci, že okamžitá rychlost je derivace polohy podle času.

Další, matematictější význam derivace je, že derivace v bodě je směrnice tečny<sup>32</sup> ke grafu funkce  $f$  v bodě dotyku  $x$ .

Můžeme také provést derivaci derivace funkce, což bude opět funkce. Tu nazýváme *druhou derivací* funkce  $f$  a značíme  $f''$ . Tak můžeme (pro hezké funkce) pokračovat dále. Obecně  $n$ tou derivací značíme  $f^{(n)}$ .

Čtenáři zběhlí v počítání derivací nejspíš ví, že výpočet derivace je pro hezké funkce přímočarý a derivaci v bodě pak už získáme snadno dosazením konkrétní hodnoty  $x$ , numerický výpočet derivace je v tomto případě zbytečný. Přesto existují situace, kdy se numerická derivace může hodit. Prvním příkladem jsou složité funkce, u kterých je náročné počítat derivaci analyticky. Druhým, častějším příkladem je situace, kdy neznáme původní funkci, ale pouze její funkční hodnoty v diskrétních bodech. Příkladem může být situace, kdy máme nafilmovaný pohybující se objekt, z každého snímku odečteme jeho polohu a chceme zjistit časový průběh rychlosti, s jakou se objekt pohyboval. Pak se nějaké formě numerické derivace nevyhneme.

Podívejme se nyní opět na definiční vztah derivace. Může nás napadnout, že numerickou derivaci implementujeme prostě tak, že  $\Delta x$  nebude nekonečně malé, ale bude to nějaký (konečně) malý krok  $h > 0$ . To lze chápat také tak, že místo tečny v bodě  $x$  konstruujeme sečnu protínající graf funkce v bodech  $x$  a  $x + h$ . Pak dostáváme vztah

$$f'(x) = \frac{f(x + h) - f(x)}{h}.$$

Tomuto druhu numerické derivace se říká *dopředná diference*,<sup>33</sup> ovšem, jak si za chvíli ukážeme, jde zpravidla o jeden z nejhorších a nejméně přesných způsobů výpočtu numerické derivace.

Obdobou výše zmíněného vzorce je *zpětná diference*

$$f'(x) = \frac{f(x) - f(x - h)}{h},$$

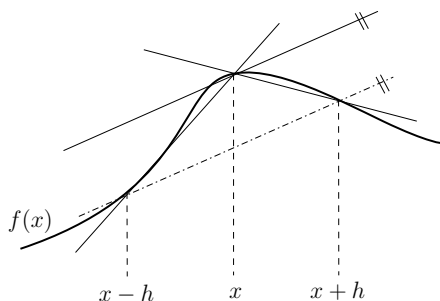
<sup>32</sup>Pokud tečnu zapíšeme rovnicí přímkou  $y = ax + b$ , pak směrnice je koeficient  $a$ . Směrnice tedy udává sklon dané přímky.

<sup>33</sup>Název pochází z toho, že členem  $f(x + h)$  „nahlédneme o krok dopředu“.

kdy derivaci aproximujeme „nahlédnutím o krok zpět“, namísto dopředu, jako u dopředné diference. Pro  $h \rightarrow 0$  zpětná a dopředná diference splývají, pro konečné  $h$  to ale neplatí, jak je vidět z obrázku 38. Navíc by dle intuice pro co nejlepší aproximaci mělo platit, že neděláme „krok zpět“, ani „krok dopředu“, ale symetricky na obě strany. Takový typ numerické derivace se nazývá *centrovaná diference* a platí pro ni vztah

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}.$$

Z obrázku 38 lze nahlédnout, že sečna příslušející centrované diferenci aproximuje tečnu v bodě  $x$  daleko lépe než sečny příslušející dopředné a zpětné diferenci. Tuto skutečnost si později dokážeme a podrobně rozebereme.



Obr. 38: Sečny dopředné, zpětné a centrované diference v bodě  $x$ .

### Taylorův rozvoj

Odbočme nyní a povězte si o mocném matematickém nástroji, který využijeme nejen v tomto díle seriálu. Existuje matematická věta, která říká, že každou hezkou<sup>34</sup> funkci můžeme nahradit nekonečným rozvojem

$$\begin{aligned} f(x) &= f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots = \\ &= \sum_{n=0}^{+\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n, \end{aligned}$$

kde  $a$  je nějaká hodnota, kolem které chceme rozvoj provést. Ta je volena tak, aby funkční hodnoty  $f$  a jejich derivací šly dobře vyčíslit, často se tedy volí  $a = 0$ . Pak rozvoj nazýváme rozvojem Maclaurinovým.

Taylorův rozvoj má hezkou vlastnost, že pro přibližnou aproximaci nemusíme použít rozvoj celý, ale můžeme jej někde useknout. Pak platí, že čím více členů po-

<sup>34</sup>Konkrétně funkci, která má všechny derivace spojité. U konečného rozvoje do řádu  $N$  postačuje spojitost derivací do řádu  $N + 1$ .

užijeme, tím je rozvoj přesnější. Také platí, že čím blíže je  $x$  ke zvolené konstantě  $a$ , tím je obvykle rozvoj přesnější. Konkrétně pro chybu konečného rozvoje

$$f(x) \approx T_N(x) = \sum_{n=0}^N \frac{f^{(n)}(a)}{n!} (x-a)^n$$

platí, že existuje číslo  $\xi$  mezi  $a, x$  takové, že

$$f(x) - T_N(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} (x-a)^{N+1}.$$

Pokud si nyní označíme krok  $h \stackrel{\text{def}}{=} x-a$  a uvědomíme si, že chyba závisí pouze na mocnině  $h$  (faktoriál je konečná konstanta,  $\xi$  závisí na  $h$ , ale díky spojitosti můžeme  $|f^{(N+1)}(\xi)|$  pro dostatečně malé  $h$  shora odhadnout konstantou), můžeme použít notaci velkého  $O$ , se kterou jsme se seznámili v minulém díle. Konečný rozvoj pak můžeme přepsat do tvaru

$$f(x+h) = \sum_{n=0}^N \frac{f^{(n)}(x)}{n!} h^n + O(h^{N+1}),$$

kde nyní provádíme rozvoj kolem pevného  $x$ .

Proveďme nyní rozvoj funkce  $f$  do druhého řádu a upravme

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + O(h^3), \\ f'(x) &= \frac{f(x+h) - f(x)}{h} - \frac{1}{2}f''(x)h + O(h^2), \\ f'(x) &= \frac{f(x+h) - f(x)}{h} + O(h), \end{aligned} \quad (46)$$

čímž jsme odvodili vztah pro dopřednou diferenci. Výhodou tohoto postupu je, že vidíme i chybu aproximace. Ta je řádu  $O(h)$ . Obdobně odvodíme i vztah pro zpětnou diferenci

$$\begin{aligned} f(x-h) &= f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 + O(h^3), \\ f'(x) &= \frac{f(x) - f(x-h)}{h} + O(h). \end{aligned} \quad (47)$$

Vidíme, že chyba metody je opět řádu  $O(h)$ .

Pokud nyní odečteme rovnici (47) od rovnice (46) dostaneme

$$\begin{aligned} f(x+h) - f(x-h) &= 2f'(x)h + O(h^3), \\ f'(x) &= \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \end{aligned}$$

což je vzorec pro centrovanou diferenci. Vidíme ale, že chyba je v tomto případě až řádu  $O(h^2)$ , neboť se členy s nižším řádem odečtyly.

### Richardsonova extrapolace

Metody vyšších řádů je možno odvodit sečtením a odečtením Taylorových rozvojų  $f(x \pm nh)$  pro  $n = 1, 2, 3, \dots$  tak, aby se všechny členy nízkých řádů (kromě členu s  $f'(x)$ ) odečetly. Ukážeme si zde ale postup známý jako *Richardsonova extrapolace*. Ze vztahu pro centrovanou diferenci máme

$$A(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + Ch^2 + O(h^4),$$

kde  $C = f'''(x)/3!$  a uvědomili jsme si, že člen s  $h^3$  je roven nule. Zapišme ten samý vztah pro krok  $2h$  (obecně bychom mohli  $nh$ )

$$A(2h) = \frac{f(x+2h) - f(x-2h)}{4h} = f'(x) + C(2h)^2 + O(h^4).$$

Nyní chceme odečíst tyto dvě rovnice tak, aby vypadl člen s  $h^2$ . To se nám podaří kombinací

$$\begin{aligned} A(2h) - 4A(h) &= -3f'(x) + O(h^4), \\ f'(x) &= \frac{-f(x+2h) + f(x-2h) + 8f(x+h) - 8f(x-h)}{12h} + O(h^4). \end{aligned}$$

Povedlo se nám tedy z původní centrované difference odstranit chybový člen nejnižšího řádu, musíme za to ale zaplatit znalostí funkční hodnoty ve čtyřech bodech namísto dvou.

Pro obecnou Richardsonovu extrapolaci platí, že pokud máme metodu

$$A(h) = F + Ch^k + O(h^{k+l}),$$

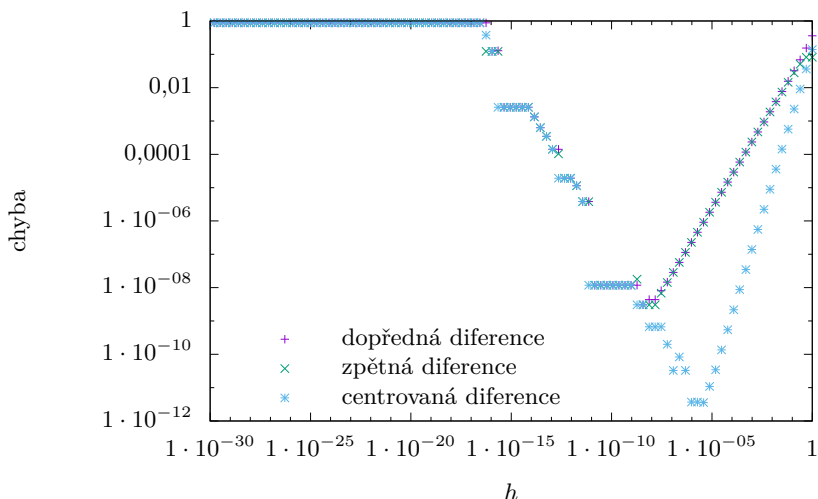
kde  $F$  je to, co chceme aproximovat (v našem případě  $F = f'(x)$ ), pak platí vztah

$$F = \frac{n^k A(h) - A(nh)}{n^k - 1} + O(h^{k+l}). \quad (48)$$

Všimněme si, že jsme vzorec (48) nenapsali speciálně pro numerickou derivaci, ale obecněji. Richardsonova extrapolace je totiž obecný nástroj použitelný v různých partiích numerických metod. Například při aplikaci v numerické integraci vede na tzv. Rombergovu integraci nebo při řešení obyčejných diferenciálních rovnic je na ní založena Bulirsch-Stoerova metoda. Nejspíš se s Richardsonovou extrapolací v průběhu seriálu ještě setkáme.

### Optimální volba kroku

V některých situacích máme krok pevně daný, například pokud máme už naměřená data z experimentu. Ale v situacích kdy máme možnost volby je vhodné vědět, jaký krok volit, abychom měli co nejmenší (nebo alespoň rozumně velkou) chybu. Z definice derivace, kdy používáme nekonečně malý krok, bychom očekávali, že bude vhodné vždy volit krok co nejmenší. Zde se ale projeví zrádnost numerických výpočtů, především zaokrouhlovací chyby známé z minulého dílu, takže vše bude nakonec úplně jinak.



Obr. 39: Závislost chyby určení derivace na kroku  $h$ . Jako testovací funkce byla zvolena  $f(x) = \sin x$  v bodě  $x = 0,5$ .

Vykresleme si nyní závislost chyby určení derivace na velikosti kroku  $h$ . Jako testovací funkci si zvolme třeba  $f(x) = \sin x$ , derivace (přesný výsledek) pak je  $f'(x) = \cos x$ . Závislost pro dopřednou, zpětnou a centrovanou diferenci je v log-log<sup>35</sup> grafu 39. Vidíme, že chyba s klesajícím  $h$  nejprve skutečně klesá, pro dopřednou a zpětnou diferenci lineárně, pro centrovanou diferenci rychleji – kvadraticky. Poté ale pro určitou hodnotu nastane zlom a chyba opět začne růst, případně se občas zastaví a vytvoří „schod“. Proč se to děje? Zatímco samotná chyba metody s klesajícím  $h$  stále klesá, zaokrouhlovací chyba, konkrétně typu cancellation, s klesajícím  $h$  roste. Pro určité optimální  $h = h_{\text{opt}}$  je chyba metody rovna zaokrouhlovací chybě a pro  $h < h_{\text{opt}}$  dominuje zaokrouhlovací chyba.

Dále vidíme, že pro centrovanou diferenci je optimální krok větší, než pro zpětnou a dopřednou diferenci. Je to náznak obecnějšího pravidla: „Čím větší řád metody, tím je optimální krok větší.“ Speciálně platí, že kombinace metody vysokého řádu a velmi malého kroku je cesta do pekel.

Odvodme si nyní teoretickou předpověď pro  $h_{\text{opt}}$  pro dopřednou diferenci. Hledáme tedy  $h$ , pro které bude chyba metody rovna zaokrouhlovací chybě. Předpokládejme, že zaokrouhlovací chyba při vyčíslení funkce je  $\Delta f(x) \approx f(x)\varepsilon$ , kde  $\varepsilon$  je

<sup>35</sup>Pokud nevíte, proč použít log-log graf pro mocninné závislosti, přečtěte si vzorové řešení úloh z 1. dílu.

strojová přesnost. Pak pro derivaci a její chybu platí

$$\begin{aligned} f'(x) + \Delta f'(x) &= \frac{f(x+h)(1+\varepsilon_1) - f(x)(1+\varepsilon_2)}{h} \\ &= \frac{f(x+h) - f(x)}{h} + \frac{f(x+h)\varepsilon_1 - f(x)\varepsilon_2}{h}, \end{aligned}$$

kde  $\varepsilon_1$  a  $\varepsilon_2$  jsou skutečné relativní chyby vyčíslení funkce, přičemž dle výše uvedeného platí, že jsou řádově rovny strojové přesnosti. Také poznamenejme, že nám jde o horní odhad chyby, proto v následujícím kroku sečteme všechny dílčí chyby v absolutní hodnotě. Pak pro zaokrouhlovací chybu dopředné difference platí

$$\Delta f'(x) \approx \frac{2\varepsilon|f(x)|}{h}.$$

Všimněme si, že  $\Delta f'(x) \propto h^{-1}$ . Právě tato vlastnost je důvodem, že pro malá  $h$  převáží.

Chyba metody dopředné difference  $\delta$  je z Taylorova rozvoje rovna

$$\delta = \frac{|f''(\xi)|}{2!} h \approx \frac{|f''(x)|}{2!} h,$$

kde jsme provedli odhad chyby dosazením  $x$  místo  $\xi$ , protože hodnotu  $\xi$  neznáme. Pro  $h = h_{\text{opt}}$  pak platí  $\delta \approx \Delta f'(x)$ , tedy (za předpokladu  $f''(x) \neq 0$ )

$$\begin{aligned} \frac{|f''(x)|}{2} h_{\text{opt}} &\approx \frac{2\varepsilon|f(x)|}{h_{\text{opt}}}, \\ h_{\text{opt}} &\approx 2\sqrt{\frac{|f(x)|}{|f''(x)|}} \sqrt{\varepsilon}. \end{aligned}$$

Protože nás zajímá pouze řádový odhad, stačí nám, že pro dopřednou diferencii platí  $h_{\text{opt}} \sim \sqrt{\varepsilon} \sim 10^{-8}$ , což odpovídá optimální hodnotě v grafu 39. Pro zpětnou diferencii bychom obdrželi stejný výsledek, pro centrovanou diferencii bychom pak obdobným postupem dostali  $h_{\text{opt}} \sim \varepsilon^{1/3} \sim 10^{-5}$ .

### Monte Carlo

*Monte Carlo* (zkráceně MC) je označení pro jakoukoli metodu, která k vyřešení úlohy používá náhodné vzorkování pravděpodobnostního rozdělení sledované veličiny pomocí generátoru pseudonáhodných čísel. Dle aplikace a použitých algoritmů se Monte Carlo dělí na kinetické, termodynamické (statistické), kvantové a mnoho dalších. V tomto seriálu se budeme postupně věnovat třem oblastem: MC integraci, geometrickému MC a zlehka se dotkneme i termodynamického MC. A aby nebyl výklad problematiky zbytečně komplikovaný, omezíme se zde na rovnoměrné rozdělení a složitější distribuce odsuneme do dalšího dílu.

### Principy metody

Princip metody Monte Carlo si ukážeme na konkrétní úloze, která se nazývá *Buffonova jehla*. Původní znění pochází z 18. století od hraběte de Buffon: *Nechť máme vodorovnou plochu rozdělenou rovnoběžnými čarami na pásy šířky  $a$ . Na tuto plochu hodíme jehlu délky  $b$ . Jaká je pravděpodobnost, že jehla protne čáru?* Vzhledem k době, kdy byla úloha zadaná, bylo možné vyřešit ji buď analyticky nebo experimentem, počítačová simulace nebyla dostupná. Naše zadání bude znít ovšem trochu jinak: Jak na základě opakovaného provedení experimentu uvedeného v Buffonově úloze určíme hodnotu čísla  $\pi$ ?

Možná jste nyní zmatení z toho, odkud se vzalo číslo  $\pi$ . To se ihned ozřejmí, jakmile vyřešíme původní Buffonovu úlohu. Její podstatou hledání tzv. *geometrické pravděpodobnosti* ve dvou dimenzích. Tento typ úloh po nás obecně požaduje porovnat obsah plochy, kterou představují pozitivní výsledky, s obsahem celkové plochy, na které se pohybujeme. Jednoduchým příkladem je nalezení pravděpodobnosti, že trefíme střed terče (bull's eye) za předpokladu, že se vždy trefíme do terče a rozdělení bodů dopadu šípky je rovnoměrné. Hledaná pravděpodobnost je pak dána poměrem obsahu červeného středu a obsahu celého terče.

V Buffonově úloze máme tyto dvě souřadnice: souřadnice  $x$  středu jehly ve směru kolmém na rovnoběžné čáry a úhel  $\vartheta$  natočení jehly v rovině. (Nenechte se zmást, souřadnice  $y$  v rovnoběžném směru zde nehraje žádnou roli.) Nyní přidáme dodatečnou podmínku<sup>36</sup>  $b < a$ , tedy délka jehly musí být menší než šířka pásů. Potom pravděpodobnost, že poloha středu  $x$  bude ve vzdálenosti menší než  $b/2$  od čáry (a tedy čára bude v dosahu jehly), udává vztah

$$P_1 = \frac{1}{a} \left( \frac{b}{2} + \frac{b}{2} \right) = \frac{b}{a}. \quad (49)$$

Nyní nalezneme pravděpodobnost  $P_2$ , že jehla protne rovnoběžku, ke které je nejbližší, pokud je tato rovnoběžka v dosahu jehly – všimněte si slova „pokud“, jde o podmíněnou pravděpodobnost. Pravděpodobnost, že jehla protne rovnoběžku, bude díky nezávislosti souřadnic dána součinem  $P_1 \cdot P_2$ . Aby jehla ve vzdálenosti  $x \leq b/2$  od rovnoběžky tuto rovnoběžku proťala, musí platit  $x \leq \frac{b}{2} \cos |\vartheta|$ . Toto si rozmyslíme z pravoúhlého trojúhelníku s přeponou  $b/2$  a přilehlou odvěsnou  $x$ . Inverzně, úhel  $\vartheta$  (v intervalu od  $-\pi/2$  do  $\pi/2$ , měřený od kolmice na rovnoběžku) leží v intervalu  $\langle \vartheta_{\min}, \vartheta_{\max} \rangle = \langle -\arccos(2x/b), \arccos(2x/b) \rangle$ . Pro dané  $x$  bychom nyní snadno určili hledanou pravděpodobnost jako poměr velikosti uvedeného intervalu s úhlem  $\pi$  (v radiánech). Jelikož jsou však  $\vartheta_{\min}$  a  $\vartheta_{\max}$  závislé na  $x$ , musíme provést integraci přes  $x$  od 0 do  $b/2$ :

$$P_2 = \frac{2}{b} \int_0^{b/2} \frac{2 \arccos(2x/b)}{\pi} dx = \frac{2}{\pi}. \quad (50)$$

Při výpočtu jsme využili tabulkový integrál

$$\int \arccos x dx = x \arccos x - \sqrt{1-x^2} + C. \quad (51)$$

<sup>36</sup>Řešení úlohy bez této podmínky je také možné, ale obsahuje trochu komplikovanější integrál a metoda výpočtu  $\pi$  pak již není tak zřejmá.

Řešení Buffonovy úlohy tedy zní: Jehla protne rovnoběžku s pravděpodobností  $P = P_1 P_2 = \frac{2b}{a\pi}$ .

Přímější, ale méně názorný postup řešení by vypadal následovně: pravděpodobnostní funkce souřadnic  $x$  (vzdálenost od nejbližší čáry) a  $\vartheta$  jsou

$$f_x(x) = \begin{cases} \frac{2}{a} & \text{pro } x \in \langle 0, a/2 \rangle \\ 0 & \text{jinak} \end{cases}$$

a

$$f_\vartheta(\vartheta) = \begin{cases} \frac{1}{\pi} & \text{pro } \vartheta \in \langle -\pi/2, \pi/2 \rangle \\ 0 & \text{jinak.} \end{cases}$$

Souřadnice  $x$  a  $\vartheta$  jsou nezávislé náhodné proměnné. V takovém případě můžeme sdruženou pravděpodobnostní funkci  $P(x, \vartheta)$  vyjádřit jako součin jednorozměrných pravděpodobnostních funkcí

$$f_{x,\vartheta}(x, \vartheta) = f_x(x)f_\vartheta(\vartheta) = \begin{cases} \frac{2}{a\pi} & \text{pro } x \in \langle 0, a/2 \rangle \wedge \vartheta \in \langle -\pi/2, \pi/2 \rangle \\ 0 & \text{jinak.} \end{cases} \quad (52)$$

Výše jsme si rozmysleli, že k protnutí rovnoběžky dojde pro  $x \leq \frac{b}{2} \cos|\vartheta|$ . Hledanou pravděpodobnost tedy získáme jako dvojný integrál

$$P = \int_{-\pi/2}^{\pi/2} \int_0^{\frac{b}{2} \cos|\vartheta|} \frac{2}{a\pi} dx d\vartheta = \int_{-\pi/2}^{\pi/2} \frac{b \cos|\vartheta|}{a\pi} d\vartheta = \frac{2b}{a\pi}. \quad (53)$$

Nabízí se tedy následující metoda, jak určit hodnotu čísla  $\pi$ : Obstaráme si velké množství jehel (pro uskutečnění experimentu jsou vhodnější špejle) a postupně<sup>37</sup> je budeme házet na pruhovanou plochu a počítat, kolikrát došlo k protnutí některé z rovnoběžek. Při házení musíme dbát na to, aby byl pohyb jehly vzduchem dostatečně chaotický a výslednou konfiguraci jehly bylo možné považovat za náhodnou. Nakonec vydělíme počet protnutí počtem jehel a dostaneme určité číslo  $c$ . Podle zákona velkých čísel pro  $n$  hodů platí

$$\lim_{n \rightarrow \infty} c = P, \quad (54)$$

tj. v limitě velkého počtu hodů vyjádříme číslo  $\pi$  vztahem

$$\pi = \frac{2b}{ac}. \quad (55)$$

Tato metoda k nalezení čísla  $\pi$  spadá do rodiny Monte Carlo. Před rozmachem MC simulací se opakované pokusy používaly pouze k odhadu nejistoty<sup>38</sup> výsledku získaného deterministickou simulací, například numerickou integrací diferenciálních rovnic. Metoda Monte Carlo přímo řeší danou úlohu na základě pravděpodobnostního přístupu.

Shrňme si nyní v obecnosti *základní kroky metody Monte Carlo*:

<sup>37</sup>Nebo všechny naráz, musíme však zaručit, že srážky jehel nenaruší náhodnost.

<sup>38</sup>Připomeňme, že nejistotou myslíme veličinu charakterizující distribuci výsledku okolo očekávané hodnoty. Více viz následující kapitola.



- Stanovíme množinu všech možných vstupních hodnot. V případě Buffonovy jehly jde o všechny dvojice hodnot, kterých můžou nabývat proměnné  $x$  a  $\vartheta$ .
- Zjistíme, jakými pravděpodobnostními funkcemi se řídí náhodné veličiny vstupující do simulace. V probrané úloze šlo o dvě rovnoměrná rozdělení pro veličiny  $x$  a  $\vartheta$ .
- Pomocí generátoru (pseudo)náhodných čísel vybíráme realizace zmíněných náhodných veličin. V experimentální formě Buffonovy jehly je náhodnost zaručena chaotickým letem jehly vzduchem.
- Na vybrané realizace aplikujeme deterministický vzorec a získáme výstupní hodnoty simulace.

Poslední bod zní velmi obecně, neboť se podstatně liší problém od problému – při zkoumání pohybu částic v plynu (termodynamické MC) uplatníme zcela jiné vztahy než při výpočtu integrálů. V Buffonově úloze je výstupní hodnotou odhad čísla  $\pi$ . Tato hodnota však nikdy nebude přesná, protože počet hodů jehlou/špejlí je konečný. Rádi bychom proto kvantifikovali chybu, které se při výpočtu dopouštíme. Bohužel, ne každá aplikace MC umožňuje snadno definovat, jak určíme nejistotu výsledku. Jak je tomu v případě Buffonovy úlohy?

### Monte Carlo integrace

Než zodpovíme otázku položenou na konci minulé kapitoly, podívejme se na Buffonovu úlohu obecněji. Při aplikaci Monte Carlo metody vlastně řešíme za pomoci simulace integrál (50), respektive (53). Řešení Buffonovy úlohy tedy spadá do specifické skupiny metod zvaných *Monte Carlo integrace*. Obecně tedy hledáme hodnotu integrálu

$$I = \int_{\Omega} f(x) dx, \quad (56)$$

kde  $f(x)$  je známá funkce a  $\Omega$  je oblast, přes kterou integrujeme. Jelikož nepředpokládáme, že každý čtenář je dobře obeznámen s integrálním počtem, krátce si zde vysvětlíme, jak pojem integrálu chápat.

Smyslem integrálního počtu je zobecnit pojmy jako je délka, plocha či objem. Pokud ve výrazu (56) položíme  $f(x) = 1$ , pak  $I$  bude udávat objem oblasti  $\Omega$ . Slovo „objem“ může být poněkud zavádějící, protože pokud je oblast například dvourozměrná plocha vytyčená body  $(0, 0)$ ,  $(2, 0)$ ,  $(2, 3)$  a  $(0, 3)$ , bude integrál představovat obsah dané plochy (zde obdélník o obsahu 6) – často však používáme termín objem pro oblasti s libovolnou dimenzí (obsah je tedy „2D objem“). Pokud je tvar oblasti komplikovaný, může být vyčíslení integrálu také komplikované. Vždy je však možné aplikovat numerický přístup: rozřežeme oblast na velký počet krychliček a sečteme jejich objem. Čím jemnější dělení použijeme, tím přesnějšího výsledku dosáhneme.

Z numerického pohledu se problém integrace příliš nezmění ani když budeme pracovat s libovolnou funkcí  $f(x)$ . Pouze budeme muset při sčítání krychliček

každý elementární objem vynásobit vahou danou hodnotou funkce  $f(x)$  ve středu krychličky.<sup>39</sup>

Teď když máme hrubou představu, jak může takový numerický výpočet integrálu vypadat, budeme schopni lépe ocenit přínos metody Monte Carlo. Prvním krokem je uzavření obecně komplikované množiny  $\Omega$  do boxu. Tento box představuje definiční obor pravděpodobnostní funkce  $p(x)$  od náhodné veličiny  $X$ . Stále pro jednoduchost uvažujeme, že  $X$  má rovnoměrné rozdělení,  $p(x)$  je tedy konstantní funkce, která všude nabývá hodnoty  $1/V$ , kde  $V$  je objem boxu (vzpomeňte, že pravděpodobnostní funkce je normovaná, takže integrál funkce  $p(x)$  přes celý box musí být roven 1). Navíc předpokládejme, že mimo oblast  $\Omega$  je funkce  $f(x)$  nulová. Pomocí generátoru náhodných čísel vybereme jednu realizaci  $x$  veličiny  $X$  a zkonstruujeme odhad integrálu

$$I_1(X = x) = \frac{f(x)}{p(x)}. \quad (57)$$

Očekáváme, že jednobodový odhad bude velmi špatný. Uvažujme na chvíli případ, kdy počítáme objem oblasti  $\Omega$  (označme ho  $O$ ), tj.  $f(x) = 1$  všude na  $\Omega$ . Pokud  $x \in \Omega$ , pak  $I_1 = V$ . Pokud  $x \notin \Omega$ , pak  $I_1 = 0$ . Hledaná hodnota zřejmě leží někde mezi 0 a  $V$ , ale nevíme kde. Dobrou vlastností tohoto odhadu je tzv. nestrannost. To znamená, že jeho střední hodnota je rovna přesnému výsledku. To si snadno potvrdíme aplikací definice střední hodnoty spojité náhodné veličiny

$$E[I_1] = \int_{\Omega} I_1(X = x)p(x)dx = \int_{\Omega} \frac{f(x)}{p(x)}p(x)dx = \int_{\Omega} f(x)dx = O; \quad (58)$$

tento výsledek platí pro libovolné  $f(x)$ . Pro výpočet rozptylu použijeme vztah<sup>40</sup>

$$\text{Var}[I_1] = E[I_1^2] - (E[I_1])^2, \quad (59)$$

tedy pro  $f(x) = 1$  a  $p(x) = 1/V$  dostaneme

$$\text{Var}[I_1] = \int_{\Omega} \left(\frac{f(x)}{p(x)}\right)^2 p(x)dx - O^2 = \int_{\Omega} Vdx - O^2 = VO - O^2 = O(V - O). \quad (60)$$

Vidíme tedy, že rozptyl našeho odhadu je přímo úměrný rozdílu mezi objemem obklopujícího boxu a oblastí zájmu  $\Omega$ . Kdybychom volili box shodný s  $\Omega$ , takže by platilo  $O = V$ , dostali bychom exaktní výsledek, ale pak celá úloha postrádá smysl, protože hledaný objem již od začátku známe. Jak tedy docílíme nižšího rozptylu (a v důsledku nižší směrodatné odchylky), aniž bychom museli omezit naši benevolenci při volbě boxu?

Odhad  $I_1$  je špatný, nicméně z geometrického pohledu víme, že pravděpodobnost zvolení  $x \in \Omega$  lze vyjádřit jako  $O/V$ . Tedy způsobem, jak odhad zlepšit, je

<sup>39</sup>Nemusí jít o střed a nemusí jít o krychličky, ale zatím s touto představou vystačíme. Fundovanější vysvětlení numerické integrace těmito metodami naleznete v dalších dílech seriálu.

<sup>40</sup>V předchozím dílu seriálu nebyl zmíněn, ale lze ho odvodit z definice střední hodnoty a rozptylu.

vzít velký počet nezávislých vzorků, označme ho  $N$ , a provést průměr přes hodnoty  $I_1(X = x_i)$ ,  $i = 1, \dots, N$ . Novým odhadem je tedy veličina

$$I_N = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}. \quad (61)$$

Rozptyl této veličiny je<sup>41</sup>

$$\begin{aligned} \text{Var}[I_N] &= \text{Var} \left[ \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \right] = \frac{1}{N^2} \left( \text{Var} \left[ \frac{f(x_1)}{p(x_1)} \right] + \text{Var} \left[ \frac{f(x_2)}{p(x_2)} \right] + \dots \right) \\ &= \frac{1}{N} \text{Var}[I_1] = \frac{1}{N} O(V - O), \end{aligned} \quad (62)$$

přičemž poslední úprava je opět pouze pro  $f(x) = 1$ . Pro směrodatnou odchylku tedy platí

$$\sigma[I_N] = \frac{1}{\sqrt{N}} \sqrt{O(V - O)}. \quad (63)$$

Výraz  $\sqrt{O(V - O)}$  je nějaký konstantní faktor daný volbou boxu a není příliš podstatný. Zásadní význam má ovšem faktor  $1/\sqrt{N}$ . Říká nám, že s rostoucím počtem vzorků klesá směrodatná odchylka s odmocninou počtu vzorků. Tedy zvýšíme-li počet vzorků stokrát, sníží se chyba desetkrát. To možná nezní jako příliš rychlá konvergence, ale při výpočtu mnohorozměrných integrálů se ukazuje být velice dobrou. Lze ukázat, že v dimenzi  $d$  konverguje výpočet integrálu pomocí dělení objemu na krychlové elementární buňky rychlostí  $N^{-2/d}$ , zatímco integrace Monte Carlo není na dimenzi závislá. Pro dimenze  $d > 4$  je tedy MC integrace preferovanou volbou.

Krátká poznámka k vyšším dimenzím – výpočet objemů geometrických těles v mnohorozměrných Eukleidových prostorech nemá mimo oblast čisté matematiky velký význam. Pokud však budeme každý volný parametr fyzikálního systému považovat za jednu dimenzi, nabízí se široká škála aplikací. Jako příklad uveďme kanonický soubor částic ideálního plynu ve statistické mechanice, kde každá z  $N$  částic plynu je popsána třemi složkami hybnosti a třemi složkami polohy. Integrací přes těchto  $6N$  proměnných získáme tzv. partiční sumu. Ta v sobě obsahuje termodynamické vlastnosti systému jako jsou vnitřní energie, entropie či tlak.

Odvozený vztah pro výpočet nejistoty odhadu integrálu za použití metody Monte Carlo můžeme aplikovat na Buffonovu jehlu. Pomocí generátoru náhodných čísel vybíráme  $N$  dvojic vzorků z rovnoměrných rozdělení  $x \in \langle 0, a/2 \rangle$ ,  $\vartheta \in \langle -\pi/2, \pi/2 \rangle$  a k výsledku přičteme  $1/N$  pokaždé, když je splněna podmínka  $x \leq \frac{b}{2} \cos |\vartheta|$ . Buffonova úloha nemá přímočarou geometrickou interpretaci jako

<sup>41</sup>To, že rozptyl součtu můžeme rozdělit na součet rozptylů, plyne z nezávislosti veličin  $I_1(X = x_i)$ . Můžete si to taky odvodit.

podíl obsahů dvou ploch, při výpočtu odchylky si proto vystačíme se znalostí pravděpodobnosti protnutí  $O/V = \frac{2b}{a\pi}$ . Pro relativní směrodatnou odchylku platí

$$\frac{\sigma}{O} = \frac{1}{\sqrt{N}} \sqrt{V/O - 1} = \frac{1}{\sqrt{N}} \sqrt{\frac{a\pi}{2b} - 1}. \quad (64)$$

Zvolíme-li konkrétní hodnoty parametrů, např.  $b = a = 1$ , dostaneme relativní směrodatnou odchylku

$$\frac{\sigma}{O} = 0,76 \frac{1}{\sqrt{N}}. \quad (65)$$

Poznamenejme ještě, že využit počítačovou simulaci Buffonovy úlohy k nalezení hodnoty čísla  $\pi$  není nejpraktičtější, protože při generování náhodných čísel potřebujeme číslo  $\pi$  již předem znát.

Existují metody, jak urychlit snižování rozptylu, ale jejich rozbor ponecháme do příštího dílu. Jednu speciální metodu přeci jen ve stručnosti uvedeme – nazývá se kvazi-Monte Carlo (qMC) a od běžné MC metody se liší tím, že nepoužívá pseudonáhodná čísla, ale tzv. kvazináhodná čísla (odborněji se mluví o posloupnostech s nízkou diskrepancí). Výhodou qMC je, že zatímco při generování náhodných čísel dochází k vytváření shluků, kvazináhodná čísla pokryjí zvolenou oblast rovnoměrněji. Nemůže se tedy stát, že bychom několikrát započítali stejný bod (nebo velmi blízké body). Zároveň však kvazináhodná čísla nejsou rozmístěna v pravidelné mříži, při změně oblasti tedy není potřeba polohy bodů přepočítat. Řád konvergence směrodatné odchylky se díky rovnoměrnému pokrytí může blížit až  $1/N$ . Za tento bonus však musíme zaplatit – kvazináhodná čísla nejsou nezávislá, jejich využití je proto omezeno (není možné je použít například v kryptografii) a statistické zpracování se kvůli korelacím komplikuje.

#### *Ukázková úloha: Integrace kruhu*

Zadání: Vypočtete obsah čtvrtkruhu o poloměru  $r = 1$  pomocí metody Monte Carlo.

Řešení: Čtvrtkruh vložíme do čtverce o straně délky  $r$  (tj. poloměry čtvrtkruhu jsou shodné s dvěma přilehlými stranami čtverce) a budeme generovat dvojice rovnoměrně rozdělených náhodných čísel na intervalu  $(0, 1)$ . Dvojice čísel představují souřadnice  $x, y$  bodů uvnitř čtverce, celkem jich vygenerujeme  $N$ . Poté spočteme všechny body, které splňují podmínku  $x^2 + y^2 \leq 1$ , tj. leží uvnitř čtvrtkruhu; tento počet označme  $M$ . Obsah je poté dán vztahem

$$S = \frac{M}{N} d^2 = \frac{M}{N}.$$

Jelikož známe vzoreček pro obsah kruhu, očekáváme výsledek

$$M/N = \pi/4 \pm 0,41 \frac{1}{\sqrt{N}},$$

kde jsme faktor 0,41 získali ze vztahu (63) pro  $V = r^2 = 1$  a  $O = \pi r^2/4 = \pi/4$ .

K řešení byl použit následující kód, který můžete s mírnými úpravami využít v seriálové úloze. Uvnitř kódu je pomocí komentářů vysvětlena potřebná syntax jazyka Python.<sup>42</sup>

```
# nacteme matematickou a numerickou knihovnu
import math
import numpy as np

# kolik bodu budeme generovat
pocet = 1000000
# pole (matice) nahodnych cisel od 0 do 1,
# ktera ma "pocet" radku a 2 sloupce, pricemz kazdy
# radek predstavuje souradnice x a y jednoho bodu
nahodna = np.random.rand(pocet,2)
# inicializace promenne, do niz budeme ukladat pocet zasahu dovnitr kruhu
pocet_uvnitr = 0

# cyklus, ktery prochazi cisla od 0 do pocet-1
for i in range(0,pocet):
    # v hranate zavorce piseme radkovy a sloupcovy index oddeleny carkou,
    # pomoci Pythagorovy vety kontrolujeme, ze bod lezi v kruhu
    if nahodna[i,0]**2 + nahodna[i,1]**2 < 1:
        # pokud je bod v kruhu, zvysime pocet zasahu o jedna
        pocet_uvnitr += 1

# vypocet prumeru; ve jmenovateli je prictena nula, aby byl vysledek float
prumer = pocet_uvnitr/(pocet+0.)
# odchylka od ocekavaneho vysledku a smerodatna odchylka dle textu serialu
# math.pi vola hodnota cisla pi
odchylka = np.abs(prumer - math.pi/4.)
smodch = np.sqrt(math.pi/4.*(1. - math.pi/4.)/pocet)

# nechame vypsati vysledek na obrazovku
print("Prumer: {}, smodch: {}, odchylka: {}".format(prumer, smodch, odchylka)
)
```

Výsledek prvního běhu programu vypadal následovně:

Prumer: 0.785928, smodch: 0.000410545841934,  
odchylka: 0.000529836602552.

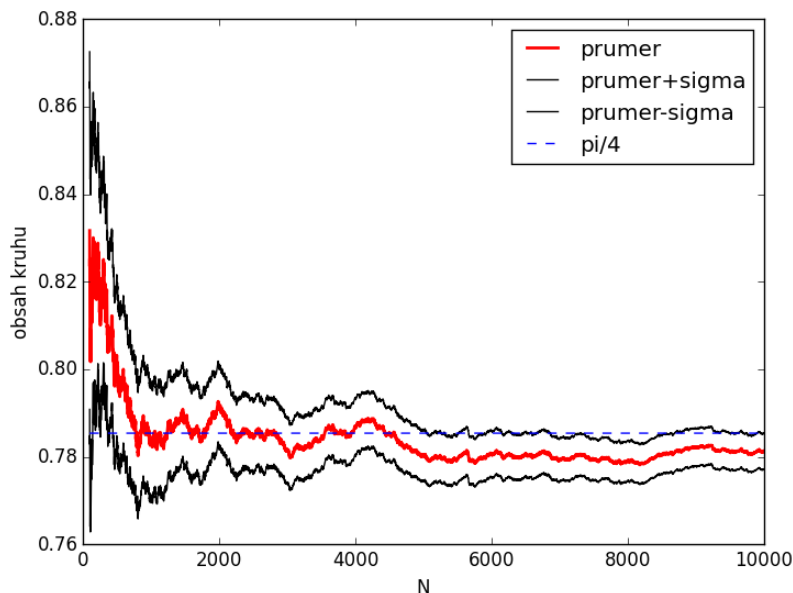
Jelikož jsme použili 1 000 000 bodů, pohybuje se relativní směrodatná odchylka okolo 1/1000 (odmocnina). Na obrázku 40 vidíte, jak se vyvíjí odhad průměru s rostoucím  $N$ .

Úlohu lze jednoduše vyřešit i v programu Microsoft Excel. Pomocí funkce RAND() naplníme dva sloupce do délky  $N$ . Volme třeba  $N = 1000$  a naplněny nechtě jsou sloupce A a B. Do buňky C1 vepíšeme vzorec =A12+B12 a vytáhneme do délky  $N$ . Poté do prázdné buňky vepíšeme

=COUNTIF(C1:C1000;"<1")/COUNT(C1:C1000)

První funkce spočte všechny buňky obsahující číslo větší než 1, druhá funkce započte všechny neprázdné buňky. Tak získáme odhad obsahu kruhu s průměrem  $d = 1$ .

<sup>42</sup>Kód je psán tak, aby byl srozumitelný, nikoli efektivní.



Obr. 40: Vývoj obsahu kruhu odhadnutého metodou Monte Carlo v závislosti na počtu bodů  $N$ . Prvních sto kroků je v grafu vynecháno kvůli velkým fluktuacím.

## Úloha II.S ... derivace a Monte Carlo integrace

10 bodů

a) Vykreslete závislost chyby na velikosti kroku pro metodu

$$f'(x) \approx \frac{-f(x+2h) + f(x-2h) + 8f(x+h) - 8f(x-h)}{12h}$$

odvozenou pomocí Richardsonovy extrapolace v textu seriálu. Jaký je optimální krok a minimální chyba? Porovnejte s centrovanou a dopřednou diferencí. Jako derivovanou funkci použijte  $\exp(\sin(x))$  v bodě  $x = 1$ .

*Bonus* Vypočítejte pro tuto metodu teoretickou velikost optimálního kroku pomocí odhadu chyb.

b) Na webu se nachází soubor s experimentálně zjištěnými  $t$ ,  $x$  a  $y$  souřadnicemi poloh hmotného bodu. Pomocí numerické derivace nalezněte časovou závislost složek rychlosti a zrychlení a vynesete obě závislosti do grafu. Jaký fyzikální děj bod nejspíše konal? Numerickou metodu si zvolte sami, svoji volbu ale odůvodněte.

*Bonus* Existuje v tomto případě přesnější varianta získání rychlosti a zrychlení, než přímočará aplikace numerické derivace?

- c) Máme zadán integrál  $\int_0^\pi \sin^2 x dx$ .
1. Nalezněte hodnotu integrálu z geometrické úvahy za pomoci Pythagorovy věty.
  2. Nalezněte hodnotu integrálu pomocí Monte Carlo simulace. Určete směrodatnou odchylku výsledku.
- Bonus* Vyřešte Buffonovu úlohu ze seriálu (odhad hodnoty čísla  $\pi$ ) pomocí MC simulace.
- d) Nalezněte vztah pro výpočet objemu šestidimenzionální koule pomocí metody Monte Carlo.
- Nápověda* Pythagorovu větu lze využít k měření vzdáleností i ve vyšších dimenzích.

(řešení str. 243)

### Kapitola 3: Na procházce s integrály

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

#### Náhodné procházky

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

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

### Markovovy procesy

V prvním dílu seriálu jsme se seznámili s pojmem náhodná proměnná. Jednalo se o proměnnou, jejíž hodnota není předem známa (tj. není to konkrétní číslo), pouze známe pravděpodobnosti nabývání hodnot z určité množiny. Tyto pravděpodobnosti byly dány pravděpodobnostním rozdělením, např. náhodná proměnná „hod kostkou“ mohla nabývat hodnot 1 až 6 z diskrétního rovnoměrného rozdělení. Pokud uspořádáme náhodné veličiny do řetězce, tj. přiřadíme náhodným veličinám indexy  $1, 2, \dots, n$ , nazýváme tento řetězec *náhodný proces*. Příkladem náhodného procesu je třeba série hodů mincí.<sup>43</sup> *Realizací* tohoto náhodného procesu je uspořádaná množina, jejíž každý prvek nabývá hodnoty „panna“ nebo „orel“ – například jsme vytvořili pomocí pěti hodů korunovou mincí realizaci náhodného procesu

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

V tomto procesu jsou prvky řetězce představovány navzájem nezávislými a shodně rozdělenými<sup>44</sup> náhodnými veličinami. Obecně však mohou být veličiny závislé a různě rozdělené. Uvažte třeba takovýto proces (A): Házím mincí, dokud nepadne panna, a pak házím kostkou, dokud nepadne šestka; pak házím mincí, dokud nepadne panna, a pak házím kostkou, dokud ... Jiný příklad (B): Házím šestistěnnou kostkou a sčítám oka tak dlouho, dokud nezískám součet větší nebo roven 20; poté vyměním šestistěnnou kostku za desetistěnnou a pokračuji v přičítání, dokud nedosáhnou součtu 40; poté se vrátím k šestistěnné kostce atd.

Nyní uvažujme speciální případ náhodného procesu, kdy jsme schopni určit stav v následujícím kroku bez znalosti historie procesu. Jinými slovy, pravděpodobnost realizace hodnoty  $x$  náhodné veličiny  $X_{i+1}$  dokážu jednoznačně určit na základě realizace  $y$  náhodné veličiny  $X_i$ . Těto vlastnosti říkáme přilehavě *bezpaměťovost* a příslušný náhodný proces se nazývá *Markovův*.<sup>45</sup> Formálně tuto vlastnost (pro diskrétní náhodné veličiny) zapíšeme

$$P(X_{i+1} = x_{i+1} | X_1 = x_1, X_2 = x_2, \dots, X_i = x_i) = P(X_{i+1} = x_{i+1} | X_i = x_i) \quad (66)$$

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

Série hodů mincí je triviálním případem bezpaměťového řetězce, kdy není potřeba znát ani stav v kroku  $i$  (toto platí pro každý řetězec navzájem nezávislých náhodných veličin). Proces A je také Markovův proces, další krok se řídí vždy současným stavem. Bepaměťový je i proces B – pro určení stavu v kroku B nám

<sup>43</sup>Konkrétně se jedná o tzv. Bernoulliho proces s pravděpodobnostmi  $p = 1/2$  a  $1 - p = 1/2$ .

<sup>44</sup>V anglické literatuře se můžete setkat se zkratkou *iid* – independent and identically distributed.

<sup>45</sup>Občas se v učebnicích můžete setkat i s označením markovský či markovovský proces, v obou případech jde o podobnostní přídavné jméno odvozené od ruského matematika Andreje Andrejeviče Markova.



postačí znalost aktuálního součtu a typu kostky, kterou jsme naposled házeli. Vymyslet proces, který není Markovův, je snadné, například: Házím kostkou a sčítám hodnoty, pokud padnou dvě šestky za sebou, přičtu k součtu v následujícím kroku bonus 10. Pro určení stavu v následujícím kroku je zjevně potřeba znát alespoň dva předchozí stavy, předpoklad (66) je porušen.

### Náhodná procházka v 1D

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Nalézt střední polohu opilce je velice jednoduché.<sup>46</sup> Platí

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

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

Poslední otázka je poněkud komplikovanější. Hledáme střední hodnotu  $E(|W_n|)$ , což je určitě nenulová hodnota. Odhadnout<sup>47</sup> chování tohoto výrazu lze pomocí odmocniny ze střední hodnoty kvadrátu (střední kvadratická hodnota, *root mean square*), kterou opět díky vlastnostem střední hodnoty zapíšeme jako

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

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

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

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

### Náhodná procházka ve více dimenzích

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

Náhodná veličina  $K_i$ , která představuje jeden krok, může ve 2D nabývat čtyř hodnot, které si označíme třeba  $l$ ,  $r$ ,  $u$ ,  $d$  (*left*, *right*, *up*, *down*). Také se na tuto veličinu můžeme dívat jako vektor se dvěma složkami, který může nabývat

<sup>46</sup>Při velkém obsahu alkoholu v krvi bude tato střední poloha pravděpodobně horizontální.

<sup>47</sup>Je ovšem možné spočítat  $E(|W_n|)$  i přesně. Můžete si to zkusit.

hodnot  $(1, 0)$ ,  $(0, 1)$ ,  $(-1, 0)$  a  $(0, -1)$ , každé s pravděpodobností jedna čtvrtina. Náhodnou procházku pak definujeme opět přes součet

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

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

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

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

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

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

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

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

Jako motivaci pro studium náhodných procházek jsme na začátku uvedli Brownův pohyb. Avšak správný model Brownova pohybu by měl být v čase spojitý. Provedeme-li pro náhodnou procházku s časovým krokem  $\Delta t$  limitu  $\Delta t \rightarrow 0$ , dostaneme tzv. Wienerův proces. Na počítači však není možné provést simulaci, která je v čase spojitá. Jelikož však naše definice náhodné procházky pevně svazuje časový krok  $\Delta t$  a délku kroku v prostoru  $\Delta l$ , můžeme se Wienerově procesu (a tedy i Brownově pohybu) přiblížit přeškálováním rozměru procházky. Jinými slovy, limitu  $\Delta t \rightarrow 0$  nahradíme limitou  $\Delta l \rightarrow 0$ . Abychom toto chování ozřejmili, ukážeme si nyní, jak vypadají výsledky simulace náhodné procházky.

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

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

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

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

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

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

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

Pro uvedené hodnoty jsme zjistili, že průměrná vzdálenost po 10 000 krocích je 89,2. To docela dobře odpovídá asymptotické hodnotě<sup>48</sup> 88,6. Spíše by nás však zajímalo, jak vypadá závislost střední vzdálenosti na počtu kroků a jak vůbec vypadá realizace náhodné procházky (trajektorie). Na konec kódu výše proto přidáme následující:

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

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

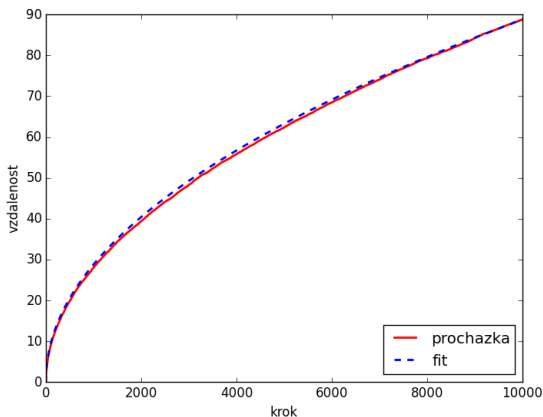
Ze závislosti vzdálenosti na počtu kroků vezmeme posledních 1 000 bodů, provedeme logaritmickou transformaci a fitujeme daty polynom prvního řádu (přímku). Proměnná `fit` v sobě obsahuje koeficienty  $a$ ,  $b$  lineárního fitu  $f(x) = ax + b$  a proměnná `cov` je matice,<sup>49</sup> na jejíž diagonále leží rozptyly odpovídající koeficientům  $a$ ,  $b$ . Po dostatečném počtu kroků  $n$  by se měla vzdálenost od počátku chovat jako  $\sqrt{n}$ , směrnice přímky by tedy měla být rovna exponentu 1/2. Výstupem jednoho běhu programu byl koeficient  $a = 0,4890 \pm 0,0004$ , což je poměrně blízko očekávané hodnotě 0.5<sup>50</sup>. Poslední řádku kódu vykreslují graf závislosti vzdálenosti na počtu kroků, abychom mohli asymptotické chování ověřit vizuálně (obr. 41), a také vykreslují trajektorii poslední vygenerované procházky. Příklady procházek o délkách 10 000 a 200 000 kroků jsou na obrázcích 42 a 43. V druhém případě již nedokážeme rozeznat jednotlivé kroky – trajektorie se blíží realizaci spojitého Brownova pohybu.

Kromě jednoduché procházky, které jsme se doteď věnovali, lze definovat ve 2D různé modifikace, které neměly v jedné dimenzi dobrý smysl. *Procházka bez návratu* v sobě zahrnuje dodatečný požadavek, že pokud nabývá  $K_i$  určité hodnoty, např.  $(1, 0)$ , nemůže  $K_{i+1}$  nabývat hodnoty opačné,  $(-1, 0)$ . *Procházka bez křížení* zakazuje, abychom do jakéhokoliv bodu vstoupili výše než jednou. Na obrázku 42

<sup>48</sup> Využili jsme znalosti multiplikativního faktoru pro dimenzi dvě, který má hodnotu  $\sqrt{\pi}/2$ . V řešení seriálových úloh nepožadujeme dohledávat tyto faktory.

<sup>49</sup> Jedná se o kovarianční matici. Pokud se o kovarianci chcete dozvědět více, doporučujeme 3. díl seriálu z 30. ročníku FYKOSu.

<sup>50</sup> Pokud bychom chtěli zkusit provést výpočet pro delší procházky, trval by jeden běh programu již poměrně dlouho (minuty a více). Nezájímáme-li se o trajektorie procházek, můžeme si místo polí `xs` a `ys` ukládat pouze aktuální vzdálenost, čímž se běh programu výrazně zrychlí.



Obr. 41: Vzdálenost 2D procházky od počátku v závislosti na počtu kroků. Graf vznikl průměrováním 10 000 procházek o délce 10 000 kroků. Mocninový fit byl proveden na posledních 1 000 bodech, v levé části grafu proto vidíme, že fit příliš nesesedí – nevykazuje asymptotické chování.

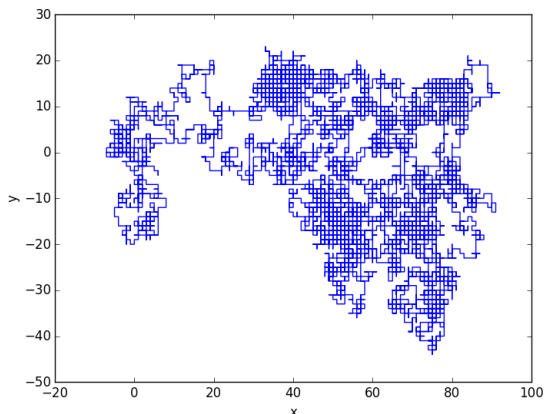
bychom tedy viděli jednoznačný začátek a konec. Platí zajímavé<sup>51</sup> tvrzení, že pravděpodobnost, že se náhodná procházka vrátí do počátečního bodu, roste s počtem kroků k jedné. Z toho nutně plyne, že každá procházka bez návratu jednou skončí. Nemá proto smysl studovat asymptotické chování střední vzdálenosti od počátku, ale můžeme například hledat střední délku procházky.

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

Náhodné procházky ve třech dimenzích<sup>52</sup> a vyšších nejsou příliš dobře prozkoumané, alespoň z analytického hlediska. Jedním z analytických výsledků je, že s rostoucí dimenzí postupně klesá pravděpodobnost návratu k nule, zatímco v jedné i ve dvou dimenzích je rovna jedné, jak jsme zmiňovali výše. Pomocí počítačových simulací lze získat představu o asymptotickém chování délky procházky, pravděpodobnosti dosažení absorpční bariéry apod. To si vyzkoušíte v seriálové

<sup>51</sup>Má využití například při maticovém zápisu integrálních rozvoji nebo při řešení Laplaceovy rovnice ve 2D.

<sup>52</sup>Ti, kdo zažili éru Windows 95/98, si pravděpodobně vzpomenou na spořič obrazovky vykreslující spleť potrubí (Pipes screensaver). Jednalo se o 3D náhodnou procházku bez křížení s reflexivními okrajovými podmínkami vykreslovanou pomocí knihovny OpenGL. Problémem tohoto spořiče bylo, že mnohdy vytěžoval obrazovku i procesor více než běžná kancelářská práce.



Obr. 42: Trajektorie 2D procházky o délce 10 000 kroků. V grafu jsou dobře vidět jednotlivé pravoúhlé kroky, je tedy zřejmé, že se nejedná o spojitý proces.

úloze. Ve velmi vysokých dimenzích nedokážeme simulacemi získat dobrou statistiku a nastupují zpět analytické metody.

### Numerická integrace

Naším cílem je numericky zjistit hodnotu určitého integrálu

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

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

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

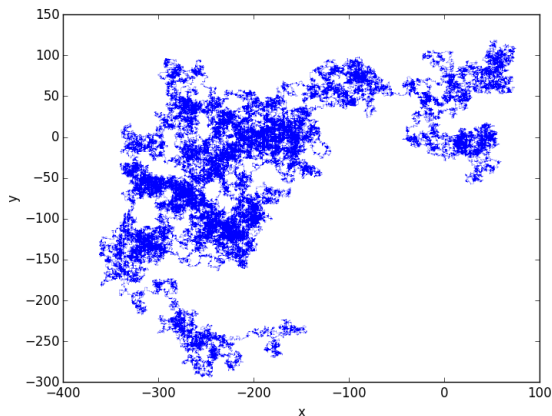
a spočtením určitého integrálu vztahem

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

Vzpomeňme si nyní na poučku,<sup>53</sup> že hodnota integrálu je rovna obsahu plochy pod grafem funkce  $f(x)$  (s opačným znaménkem, pokud je funkce záporná). Celý problém se tedy redukuje na výpočet obsahu plochy. První, co nás napadne, je aproximovat plochu obdélníkem o stranách  $b - a$  a  $f(\frac{a+b}{2})$ . Jde o jednu z variant

<sup>53</sup>Ve skutečnosti jde o definici Riemannova integrálu.





Obr. 43: Trajektorie 2D procházky o délce 200 000 kroků. Diskrétní mřížka již není patrná, procházka se blíží spojitému procesu.

*obdélníkového pravidla*, nicméně jak asi uhadneme, tato aproximace není pro většinu funkcí příliš přesná. Lepší rozhodně bude použít místo obdélníka lichoběžník. Aproximace integrálu je pak rovna

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

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

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

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

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

pak má tvar

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

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

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

Představili jsme si základní metody výpočtu určitého integrálu v jedné dimenzi. V praxi se pak používají i jiné, pokročilejší metody,<sup>54</sup> pro jejich použití je ale nutné ovládat pokročilý matematický aparát, proto se jimi zde nebudeme zabývat. N-C vzorce lze zobecnit i pro vyšší dimenze, vzpomeňme si ale na minulý díl, kde jsme zjistili, že pro  $d > 4$  je výhodnější použít Monte Carlo integraci. Za určitých okolností, například pokud oblast, přes kterou integrujeme není pravoúhlá, vyplatí se kvůli jednoduchosti použít MC integraci i v nižších dimenzích.

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

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

a platí (viz obecné odvození v minulém díle seriálu)

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

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

### Generování náhodných čísel s obecnou hustotou pravděpodobnosti

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

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

---

<sup>54</sup>Například Gaussovy kvadratury.

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

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

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

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

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

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

### Inverze distribuční funkce

Prvním způsobem generování náhodných čísel s libovolnou distribucí je následující tvrzení. Mějme náhodnou proměnnou  $X$  s distribuční funkcí  $F_X(x)$ . Pak proměnná  $Y$  definovaná výrazem  $Y = F_X(X)$  má rovnoměrné rozložení na intervalu  $\langle 0, 1 \rangle$ . Důkaz je jednoduchý<sup>55</sup>, platí

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

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

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

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

### Boxova-Mullerova transformace

Jednou z aplikací inverze distribuční funkce je efektivní metoda generování náhodných čísel s normálním rozdělením.<sup>56</sup> To má hustotu pravděpodobnosti

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

Je známo, že integrál hustoty pravděpodobnosti, resp. distribuční funkce nelze vyjádřit pomocí elementárních funkcí, ale jenom jako tzv. chybová funkce, která je tímto integrálem definována. S nalezením jednoduché inverze se pak můžeme

<sup>55</sup>Převzato z [https://en.wikipedia.org/wiki/Probability\\_integral\\_transform](https://en.wikipedia.org/wiki/Probability_integral_transform).

<sup>56</sup>Neboli Gaussovým rozdělením se středem  $\mu = 0$  a rozptylem  $\sigma^2 = 1$ .

rozloučit. Můžeme ale použít trik. Vezměme normální rozložení ve dvou dimenzích s navzájem nekorelovanými složkami

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

neboli v polárních souřadnicích

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

a pokusme se spočítat distribuční funkci vzhledem k radiální souřadnici

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

Nyní již můžeme distribuční funkci invertovat

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

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

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

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

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

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

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

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

má Gaussovo rozdělení se středem  $\mu$  a rozptylem  $\sigma^2$ .

*von Neumannova metoda*

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

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

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

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

*Metropolisův-Hastingsův algoritmus*

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

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

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

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

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

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

### Úloha III.S ... na procházce s integrály

10 bodů

- Vymyslete tři odlišné příklady markovovského procesu, z toho alespoň jeden fyzikální. Je procházka bez návratu markovovská? A co procházka bez křížení?
- Mějme 2D náhodnou procházku bez návratu na čtvercové síti s počátkem v bodě  $(x, y) = (0, 0)$ , která je omezena absorpčními bariérami  $b_1: y = -5$ ,  $b_2: y = 10$ . Nalezněte pravděpodobnost, že v bariéře  $b_1$  skončíme dříve než v  $b_2$ .
- Proveďte simulaci pohybu brownovské částice ve 2D a vykreslete graf závislosti střední vzdálenosti od počátku na čase. Uvažujeme diskrétní čas a konstantní délku kroku (jeden krok simulace trvá  $\Delta t = \text{konst}$ , délka kroku je  $\Delta l = \text{konst}$ ) a umožňujeme pohyb do libovolného směru, tj. každý krok je specifikován délkou a úhlem  $\vartheta \in \langle 0, 2\pi \rangle$ , přičemž všechny směry jsou stejně pravděpodobné. Zajímá nás především asymptotické chování, tedy vývoj střední vzdálenosti pro  $t \gg \Delta t$ .
- Chybová funkce je definována vztahem

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Tabelujte tuto funkci, tedy vypočítejte integrál pro mnoho různých  $x$ . Do řešení nevkládejte tabulku hodnot, ale graf funkce. Zkuste tuto funkci opět numericky zderivovat. Co dostanete?

- e) Najděte si definici hustoty pravděpodobnosti Maxwellova-Boltzmannova rozdělení  $f(v)$ , tedy rozdělení rychlostí molekul ideálního plynu. Spočítejte pak pomocí MC integrace střední hodnotu rychlosti definovanou

$$\langle v \rangle = \int_0^{\infty} v f(v) dv,$$

přičemž pro vzorkování použijte náhodná čísla dle Maxwellova-Boltzmannova rozdělení získaná Metropolisovým-Hastingsovým algoritmem. Hodnotu pro konkrétní zvolené parametry srovnajte s hodnotou z literatury.

(řešení str. 250)

## Kapitola 4: Kořeni a automati

31. ročník FYKOSu a s ním i seriál o numerických metodách a simulacích vstupují do druhé poloviny. Je proto na čase opustit metodu Monte Carlo a věnovat se zcela novému tématu – buněčným automatům. Seznámíme se s jejich principy a ukážeme si několik jednoduchých příkladů jejich použití. V numerické části se budeme věnovat klasickému, ale velmi užitečnému tématu: hledání kořenů funkcí.

### Hledání kořenů

Hledáním kořenů funkce  $f(x)$  máme na mysli hledání všech nebo některých řešení rovnice  $f(x) = 0$ . Předpokládáme přitom, že  $f$  je reálná funkce reálného argumentu, tedy že  $x$  i hodnoty  $f(x)$  jsou reálná čísla (a ne komplexní čísla, vektory, ...). Dále předpokládáme, že funkce je spojitá (její graf lze nakreslit jedním tahem), pro některé metody pak musíme předpokládat i spojitost prvních derivací (graf funkce nemá ostré „zuby“, takové funkce se proto označují jako hladké). Je poměrně zřejmé, že jde o rozumné předpoklady, které naprostá většina funkcí, se kterými jste se na střední škole setkali, splňuje.

Ve škole jste se většinou setkali pouze s rovnicemi, které dokážete analyticky vyřešit. Ve skutečnosti je ale takových rovnic poměrně málo.<sup>57</sup> Jako příklad analyticky neřešitelných rovnic poslouží třeba  $x = \cos x$  nebo řešení úlohy Jáchymovská z letošního FoLu. Hledání kořenů má ale i jiné aplikace, například může posloužit k numerickému nalezení inverze funkce. Pokud totiž pro zvolené  $y$  chceme najít  $x$  takové, že  $y = f(x)$ , pak nejde o nic jiného, než právě hledání kořenu funkce  $g(x) = f(x) - y$ . Věříme, že jsme vás již dostatečně namotivovali, pojďme si proto nějaké základní metody představit.

### Metoda půlení intervalu

Na začátku si potřebujeme zvolit dvě hodnoty  $a_0$  a  $b_0$  tak, že v intervalu  $(a_0, b_0)$  leží právě jeden kořen (ten, který chceme najít). Pak vezmeme střed tohoto intervalu, tedy hodnotu  $c_0 = (a_0 + b_0)/2$  a podíváme se, jestli kořen leží v intervalu  $(a_0, c_0)$ , nebo  $(c_0, b_0)$  (nebo jestli není  $f(c_0)$  zrovna nula, i když díky omezené přesnosti reálných čísel tento případ typicky nenastává). Interval, ve kterém leží kořen, pak

<sup>57</sup>Ve smyslu, že pokud náhodně vymyslíme rovnici, tak ji vyřešit nedokážeme.

použijeme jako interval  $(a_1, b_1)$  a iterujeme pro stále větší  $n$ , dokud nebude interval  $(a_n, b_n)$  dostatečně malý. Pak je kořen přibližně roven hodnotě  $(a_n + b_n)/2$  s maximální chybou  $(b_n - a_n)/2$ .

Algoritmus máme téměř hotov, chybí zjistit, jak rozhodnout, ve kterém z intervalů kořen leží. Využijeme zde Bolzanovy věty, která říká: „Nechť  $f$  je spojitá na omezeném a uzavřeném intervalu  $(a, b)$  a  $f(a)f(b) < 0$ . Pak existuje (alespoň jedno)  $c \in (a, b)$  takové, že  $f(c) = 0$ .“ Rozmyslete si, že věta říká poměrně intuitivní věc: „Pokud  $f(a)$  a  $f(b)$  mají rozdílné znaménko a spojíme je jedním tahem (grafem funkce), pak tento tah někde protíná osu  $x$ .“ Věta nám tedy napovídá, že správným testem je kontrolovat jestli  $f(a_i)f(c_i) < 0$ , nebo  $f(c_i)f(b_i) < 0$  (pokud  $f(c_i) \neq 0$ , bude alespoň jedna nerovnost platit). Všimněte si ale, že věta nám neříká, že kořen bude v intervalu pouze jeden. To musíme zajistit na začátku, proto jsme si původní interval zvolili tak, aby v něm ležel právě jeden kořen. Pokud toto nesplníme, pak může metoda samozřejmě selhat – pokud pro počáteční interval platí  $f(a_0)f(b_0) < 0$ , nějaký kořen nalezneme vždy, ale jiné se můžou ztratit.

Metoda půlení intervalu (též bisekce) je velmi jednoduchá metoda, její základní myšlenka se tedy ujala i v problémech, které s hledáním kořenů nesouvisí. Jako příklad poslouží třeba hledání hodnoty v setříděném poli, kdy se nejprve koukneme doprostřed, pokud je hodnota menší, koukneme se na prvek v jedné čtvrtině délky pole, pokud je větší, na prvek ve třetí čtvrtině, ... Je pozoruhodné, že se tato metoda osvědčila i jako „dřevorubecký způsob“ hledání některých typů chyb v programu. Pokud nám např. nějaká funkce dává chybný výsledek, stačí si postupně vypisovat hodnoty z poloviny metody, čtvrtiny metody, ... a ověřovat, jestli je tam hodnota výpočtu ještě v pořádku. Tímto způsobem pak poměrně rychle dokážeme najít konkrétní příkaz, kde chyba nastává.

Z popisu algoritmu je patrné, že bisekce (stejně jako všechny ostatní metody, které zde představíme) je iterativní metoda. Nikdy nedostaneme zcela přesný výsledek, ale čím více iterací provedeme, tím více se mu budeme blížit. Důležitou vlastností iterativních metod je rychlost konvergence, tedy to, jak rychle se budeme správné hodnotě blížit s rostoucím počtem iterací. Nyní se pokusme tento pojem formalizovat. Mějme posloupnost  $\{x_n\}_{n=0}^{\infty}$  konvergující k hodnotě  $s$ . Pokud existují konečné konstanty  $k \neq 0$  a  $p > 0$  takové, že

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - s|}{|x_n - s|^p} = k,$$

pak řekneme, že posloupnost konverguje s *řádem konvergence*  $p$ . Speciálně pokud  $p = 1$ , pak posloupnost konverguje lineárně, pokud  $p = 2$ , pak konverguje kvadraticky, ...

Pokud bude konvergující posloupností posloupnost chyb metody (v závislosti na počtu iterací)  $\{e_n\}_{n=0}^{\infty}$ , kde  $e_n = |x_n - s|$ , pak dle definice bude pro danou posloupnost pro dostatečně veliké  $n$  platit  $e_{n+1} \approx ke_n^p$ . Tento rekurentní vztah můžeme rozepsat do podoby

$$e_n \approx k^{1+p+\dots+p(n-n_0-1)} e_{n_0}^{p(n-n_0)},$$



kde jsme si zvolili pevné dostatečně velké  $n_0$  a uvažujeme  $n \geq n_0$ . Zlogaritmováním pak dostaneme

$$\ln e_n \approx p^{n-n_0} \ln e_{n_0} + \sum_{j=0}^{n-n_0-1} p^j \ln k.$$

Pro případ  $p > 1, k < 1, e_{n_0} < 1$  tedy platí, že závislost *logaritmu* chyby na  $n$  se bude chovat přibližně jako  $-Cp^n$ , kde  $C > 0$  je konstanta. Pro  $p = 1$  se logaritmus chyby chová jako  $-Cn$ , chyba tedy klesá exponenciálně.

S rychlostí konvergence jsme se již setkali v úloze o hledání čísla  $\pi$  v první sérii. Ve vzorovém řešení bylo zmíněno, že metoda konverguje kvadratically s počtem hran, a názorně jsme si předvedli, co to znamená. Upozorníme zde ale, že šlo o jinou definici rychlosti konvergence, která je používána spíše pro diskretizační metody (metody, kde kouskujeme nějakou oblast či čas na malé dílky). V obou definicích ale platí, že čím vyšší řád, tím lépe metoda konverguje, jen to pokaždé znamená něco trochu jiného.

Vypočtíme nyní řád konvergence pro metodu bisekce. Je zřejmé, že v  $n$ tém kroku hledáme kořen v intervalu šířky  $|b_n - a_n| = 2^{-n}|b_0 - a_0|$ . Protože v tomto intervalu kořen jistě leží, můžeme touto hodnotou shora odhadnout chybu metody  $e_n$  v  $n$ tém kroku. Posloupnost  $\{e_n\}$  konverguje k nule, dle výše uvedené definice tedy hledáme takové  $p > 0$ , aby výraz

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^p} = \lim_{n \rightarrow \infty} \frac{2^{-n-1}|b_0 - a_0|}{(2^{-n}|b_0 - a_0|)^p} = \frac{1}{2}|b_0 - a_0|^{1-p} \lim_{n \rightarrow \infty} (2^{p-1})^n$$

byl roven nenulovému reálnému číslu. Je vidět, že pro  $p > 1$  je hodnota výrazu rovna  $+\infty$ , pro  $p \in (0, 1)$  je pak rovna nule. Řešením je  $p = 1$ , kdy je hodnota výrazu rovna číslu  $1/2$ . Dokázali jsme tedy, že metoda konverguje lineárně.

Pokud si to tedy shrneme, pro metodu bisekce potřebujeme dopředu znát interval, kde se kořen vyskytuje a jde o lineárně konvergující metodu. Lze si také rozmyslet, že metoda konverguje vždy (což, jak uvidíme, není samozřejmost). Jde tedy o jednoduchý a robustní algoritmus, ne ale příliš rychlý. Představme si tedy rychleji konvergující metody.

### Metoda sečen

Myšlenka této metody je opět poměrně jednoduchá. Na začátku potřebujeme dva body  $x_0$  a  $x_1$ . Kořen nemusí tentokrát nutně ležet mezi nimi, musí ale ležet „dostatečně blízko“. Nyní proložíme body  $[x_0, f(x_0)]$  a  $[x_1, f(x_1)]$  přímkou – sečnu funkce  $f(x)$ . Ta je v jistém smyslu rozumnou aproximací funkce  $f(x)$  blízko bodů  $x_0$  a  $x_1$ , navíc je to přímka, a pro ni dokážeme najít kořen lehce. Označme tento kořen sečny  $x_2$ . Protože je ale sečna pouze aproximací,  $x_2$  (většinou) není kořenem funkce  $f(x)$ . Proto proložíme další sečnu body  $[x_1, f(x_1)]$  a  $[x_2, f(x_2)]$ , najdeme kořen této sečny  $x_3$  a dále iterujeme až do dosažení konvergence. Tu můžeme detekovat například tak, že se dvě následující hodnoty  $x_i$  v podstatě neliší.

Bez důkazu uvedme, že řád konvergence metody je roven zlatému řezu, tedy přibližně 1,62, což je lepší než u lineárně konvergující bisekce. Problém ale je, že metoda nekonverguje vždy. Jako příklad si uvedme funkci, která je na nějakém

intervalu konstantní. Pokud se v nějaké iteraci stane, že chceme proložit sečnu dvěma body ležícími v tomto intervalu, nebude mít tato sečna žádný kořen. To je samozřejmě extrémní případ, stačí, pokud je funkce někde málo strmá (v porovnání se strmostí v okolí kořenu funkce), pak sice sečna protne osu  $x$ , ale zpravidla v bodě ležícím daleko od skutečného kořenu funkce.

### Metoda regula falsi

Pokusme se modifikovat metodu sečen tak, aby byla vždy konvergentní. Na začátku si opět zvolíme  $x_0$  a  $x_1$ , tentokrát ale tak, aby kořen ležel mezi nimi (aby  $f(x_0)f(x_1) < 0$ ). Těmito dvěma body opět proložíme sečnu a najdeme její kořen  $x_2$ . Nyní ale nastane podstatná změna. Další sečnu proložíme buď body  $x_1$  a  $x_2$ , nebo  $x_0$  a  $x_2$  podle toho, mezi kterou dvojicí bodů leží skutečný kořen funkce  $f(x)$ , což, podobně jako u bisekce, ověříme podle toho, jestli  $f(x_0)f(x_2) < 0$ , nebo  $f(x_1)f(x_2) < 0$ . Poté iterujeme až do požadované přesnosti.

Vidíme, že podstatnou modifikací je to, že si hlídáme, aby skutečný kořen ležel vždy mezi průsečíky sečny s osou  $x$ . Tento požadavek nám zajistí, že se odhad kořene  $x_i$  bude přibližovat skutečnému kořenu a metoda tedy bude vždy konvergentní. Zaplatíme za to ale tím, že metoda konverguje pouze lineárně, nicméně o něco rychleji než bisekce (ne o tolik, aby se to projevilo na řádu konvergence, ale i tak je to vítané zlepšení).

### Newtonova metoda tečen

Myšlenka Newtonovy metody je taková, že zvolíme nějaký bod  $x_0$  „blízko“ hledaného kořenu funkce  $f(x)$ . Funkci poté aproximujeme tečnou v bodě  $x_0$ , což technicky provedeme rozepsáním Taylorova polynomu v okolí bodu  $x_0$  do 1. řádu

$$f(x) \approx y = f(x_0) + f'(x_0)(x - x_0).$$

Nyní, podobně jako v předchozích dvou metodách, nalezneme kořen této aproximace, tedy položíme  $y = 0$ . Pak platí

$$x = x_0 - \frac{f(x_0)}{f'(x_0)},$$

protože je ale toto  $x$  opět pouhou aproximací kořenu, označme jej jako  $x_1$  a iterujme dále. Kompletní rekurentní předpis Newtonovy metody tedy je

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Specifikem této metody je, že je potřeba znát nejen funkční hodnoty samotné funkce  $f(x)$ , ale i její derivace. Mohli byste namítnout, že pokud bychom derivaci neznali, mohli bychom použít její numerickou aproximaci. Nicméně lze ukázat, že v případě použití dopředné diference přechází Newtonova metoda na metodu sečen (zkuste si to, jde jen o dosazení do vzorce). To dává smysl, neboť numerická derivace je pouze aproximací, hledáme tedy vlastně sklon sečny procházející body  $x$

a  $x + h$ , zatímco při použití derivace je  $h$  infinitesimálně malé, sečna tedy přejde v tečnu.

S metodou sečen má Newtonova metoda společnou i tu nehezkou vlastnost, že nekonverguje vždy. Pokud ale konverguje, tak konverguje (za splnění určitých předpokladů) kvadraticky, jde tedy o nejefektivnější z metod, které jsme si zde představili.

### Banachova věta o kontrakci

Jako poslední si představme metodu, kterou nelze použít obecně, nicméně za určitých okolností se může vyplatit ji znát. Navíc jde o poměrně zajímavou matematickou větu, se kterou jste se již možná setkali, ani o tom nevíte.

Banachova věta je poměrně obecné tvrzení, které se zavádí na abstraktním matematickém objektu nazývaném metrický prostor. My se zde ale spokojíme s méně obecnou verzí této věty, kdy si ji zavedeme pouze na intervalech reálných čísel, což, pokud přirozeně použijeme absolutní hodnotu rozdílu dvou čísel jako tzv. metriku, je speciální případ metrického prostoru.

Nejprve si ale potřebujeme definovat několik pojmů. Funkce  $f : \langle a, b \rangle \rightarrow \mathbb{R}$  se nazve *lipschitzovská*, pokud existuje  $L \geq 0$  takové, že  $\forall x, y \in \langle a, b \rangle$  platí  $|f(x) - f(y)| \leq L|x - y|$ . Pokud navíc  $L < 1$ , pak funkci  $f$  na intervalu  $\langle a, b \rangle$  nazveme *kontrakcí*. Toto označení má jasný význam, neboť funkční hodnoty kontrakce jsou si blíže, než původní argumenty, kontrakce tedy interval  $\langle a, b \rangle$  zobrazuje na interval  $\langle c, d \rangle$ , který je menší, než původní interval, „zkontrahuje jej“.

Nyní již samotná Banachova věta. Nechť  $f : \langle a, b \rangle \rightarrow \langle a, b \rangle$  je kontrakce. Pak existuje jediné  $x_P \in \langle a, b \rangle$  takové, že  $f(x_P) = x_P$  ( $x_P$  je tzv. *pevný bod zobrazení*). Navíc pokud  $x_0 \in \langle a, b \rangle$  a  $x_{k+1} = f(x_k)$ , pak pro všechna  $k = 1, 2, \dots$  platí

$$|x_k - x_P| \leq \frac{L^k}{1-L} |x_0 - x_1|.$$

Takto matematicky napsaná věta nejspíš vypadá hrozně, pojďme si tedy rozebrat, co říká. První část tvrdí existenci právě jednoho bodu, který se kontrakcí zobrazí sám na sebe. Nejlépe si to představíme na příkladu. Hezkým příkladem kontrakce je turistická mapa, která zobrazuje reálné body ve světě na papír. Banachova věta pak tvrdí, že pokud mapu položíme na zem někde v oblasti, kterou zobrazuje,<sup>58</sup> pak existuje na mapě právě jeden bod, který se zobrazuje sám na sebe. Druhá část věty pak říká, že pokud vezmeme libovolný bod z intervalu  $\langle a, b \rangle$  a budeme na něj opakovaně aplikovat kontrakci  $f$ , dokonvergujeme právě do pevného bodu zobrazení. Věta navíc říká, jak rychle tam dokonvergujeme (jde vidět, že jde o lineární konvergenci).

Zde se dostáváme k aplikaci. Zkusili jste si někdy na kalkulačce zadat nějaké číslo a pak opakovaně mačkat tlačítko *cos*? Všimli jste si přitom, že ať už zadáte jakékoliv číslo, vždy nakonec dospějete k číslu přibližně 0,739? Pak jste použili právě Banachovu větu o kontrakci a našli tak kořen rovnice  $x = \cos x$ , neboť

<sup>58</sup>To je důležitý požadavek, vyjádřený matematicky  $f : \langle a, b \rangle \rightarrow \langle a, b \rangle$ . Pokud mapu Prahy položíme na zem v New Yorku, nejsou předpoklady Banachovy věty splněny.

funkce  $\cos$  je kontrakce. Počáteční hodnotu jste mohli volit kdekoliv, protože je to kontrakce na libovolném intervalu obsahujícím interval  $(-1, 1)$ .

Nevýhodou Banachovy věty je, že funguje pouze na kontrakce, musíte tedy mít štěstí na řešený problém. V praxi tedy nejspíš použijete některou z ostatních metod. Jde ale rozhodně o pozoruhodný a v určitých oblastech matematiky i velmi důležitý koncept.

### *Celulární automaty*

Mnohé matematické a fyzikální systémy jsou popsány rovnicemi, které nedokážeme analyticky řešit a zapojujeme proto do práce numeriku a simulace. Existují však i situace, kdy je systém příliš komplikovaný na to, abychom dokázali všechny potřebné rovnice vůbec sepsat. O to překvapivější je, když takový systém po čase vykazuje zřejmou symetrii, opakující se geometrické vzorce, vyvíjí se od chaosu k jednoduchému uspořádání. Příkladem může být růst sněhové vločky či krystalu obecně. Ačkoli je agregace částic velice komplikovaný a tedy těžko předvídatelný proces, výsledkem je geometricky přesný krystal. Jinými slovy, jednoduché globální uspořádání je důsledkem složitého působení lokálních interakcí.

Aniž bychom příliš zabíhali do termodynamiky, rozmyslíme si, že jestliže entropie („neuspořádanost“) izolovaného systému samovolně neklesá, musí systémy vykazující samsopřádaní být otevřené.<sup>59</sup> Naše neschopnost snadno popsat takový systém soustavou rovnic s jednoznačným řešením plyne z toho, že by tato soustava musela obsahovat časový vývoj v principu nekonečného prostředí. Chceme tedy najít model sestávající z rozumně malého počtu pravidel, který dokážeme prostudovat detailněji než velkou soustavu diferenciálních rovnic (popisující stejný problém) a zároveň bude vykazovat netriviální dynamické chování vedoucí, pro určité hodnoty parametrů, k samoorganizaci. Jak jsme již předeslali, cílem tohoto textu bude ukázat, že právě takový model nám poskytnou buněčné (celulární) automaty.

Se studiem celulárních automatů (CA) začali matematictí fyzikové Stanislav Ulam a John von Neumann během pobytu v Los Alamos za druhé světové války. Ulamovou motivací bylo zkoumání růstu krystalů pomocí mřížkových modelů, zatímco von Neumanna k CA přivedla myšlenka autoreplikace strojů. Společně pak svoje znalosti použili ke studiu pohybu tekutin pomocí metody, která spočívala v prostorové diskretizaci tekutiny a zkoumání interakce mezi sousedními buňkami. V sedmdesátých letech se do širšího povědomí dostal 2D buněčný automat zvaný Hra života, který objevil John Conway.<sup>60</sup> Tento automat je zajímavý množstvím statických i dynamických objektů, které se v něm i přes jeho jednoduchá pravidla objevují, a ještě se k němu později vrátíme. V osmdesátých letech se do práce

<sup>59</sup>Otevřený termodynamický systém je takový, který si s okolím může vyměňovat hmotu i energii. O okolí se nezajímáme (neznáme jeho extenzivní popisné veličiny jako je objem, hmotu, entropie), pouze měříme toky skrze hranice studovaného systému. Izolovaný systém si nevyměňuje ani hmotu, ani energii.

<sup>60</sup>Anglický matematik, narozen 1937.

na buněčných automatech pustil Stephen Wolfram,<sup>61</sup> přičemž rigorózně klasifikoval všechny jednorozměrné automaty a pomohl rozšířit CA jako metodu do dalších oborů, nejen přírodovědných.

### Vlastnosti a klasifikace CA

Celulární automat je jednoznačně definován pomocí tří objektů. První z nich je (pro účely simulací konečná, matematicky je možné studovat i nekonečné) *mřížka*  $\Lambda$  s pozičním indexem  $i$ , která nám říká, jakým způsobem jsme diskretizovali prostor. Může být čtvercová, hexagonální i jakákoli jiná. Dále musíme zadat *množinu stavů*  $S$ , kterých může nabývat každá jednotlivá buňka. Stav konkrétní buňky označíme  $s_i$ . Nakonec musíme znát lokální *pravidlo*, podle kterého se systém v diskrétním čase vyvíjí. Takové pravidlo lze obecně napsat ve tvaru

$$s_i(t+1) = f(\{s\}_{O(i)}(t)). \quad (67)$$

Zde máme na levé straně stav buňky na pozici  $i$  v čase  $t + \Delta t$ , přičemž jsme pro jednoduchost volili  $\Delta t = 1$ . Napravo máme funkci  $f$ , která nabývá hodnot z množiny  $S$ . Jejím argumentem jsou stavy buněk z okolí  $O(i)$  v čase  $t$ . Okolí můžeme definovat libovolně, obvykle se jedná o nejbližší sousedy, samotná buňka  $i$  může, ale nemusí být do okolí zahrnuta.

Ještě než si ukážeme příklad jednoduchého pravidla  $f$ , povězme si něco stručně o klasifikaci buněčných automatů. Podle charakteru pravidla můžeme CA dělit na *deterministické* a *stochastické*. V prvním případě je na pravé straně rovnice (67) obyčejná funkce, zatímco stochastické pravidlo má na pravé straně funkci vracející náhodnou proměnnou (která může nabývat hodnot pouze z množiny  $S$ ). Dále můžeme CA rozdělit podle toho, jak aplikují své pravidlo. Pokud při provádění jednoho časového kroku vkládáme do funkce napravo vždy hodnoty buněk, kterých nabývali před provedením tohoto kroku, jedná se o *synchronní* automat. V jakémkoli jiném případě nazýváme automat *asynchronním*. Striktně řečeno jsme tedy výše definovali pouze deterministický a synchronní automat, to proto, abychom se vyhnuli komplikovanému zápisu při větším zobecnění. Dalším možným kritériem pro klasifikaci CA je jejich komplexita, která sahá od typů, kdy vždy po pár krocích dosáhneme statického stavu, až po automaty s velmi dlouhým dynamickým vývojem. Užitečným kritériem klasifikace je také stabilita vůči poruše.

A nyní k příkladu. Uvažujme 1D automat (tvar mřížky zde není potřeba udávat) s binární množinou stavů  $S = \{0, 1\}$  s pravidlem

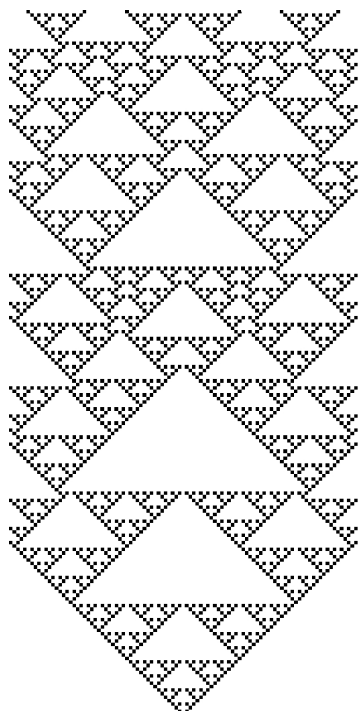
$$s_i(t+1) = s_{i-1}(t) \vee s_{i+1}(t), \quad (68)$$

kde symbol  $\vee$  označuje logický operátor výlučné nebo (exclusive or, xor).<sup>62</sup> Při časovém vývoji se tedy u každé buňky díváme, jakých hodnot nabývají nejbližší sousedi. Pokud právě jeden z nich má hodnotu 1 (uvažujeme synchronní automat,

<sup>61</sup> Anglický počítačový fyzik, narozen 1959. Ano, je to *ten* Wolfram. Jeho zájem o celulární automaty se projevili i v online algebraickém systému Wolfram|Alpha, který ve svých starších verzích mívával na čekací obrazovce během výpočtu animaci vývoje některých CA.

<sup>62</sup> Od klasického *nebo* (or) se liší tím, že pro  $A = 1, B = 1$  platí  $A \vee B = 1$ , ale  $A \vee B = 0$ .

tedy v čase  $t$ ), pak bude mít buňka na pozici  $i$  v čase  $t + 1$  hodnotu 1, jinak 0. Jelikož je mřížka  $\Lambda$  omezená, je potřeba stanovit, jak se chovají buňky na okraji. Obvykle předpokládáme periodické podmínky (sousedem poslední buňky je první buňka), ale nic nám nebrání volit i jiné. Obrázek 44 nám ukazuje prvních 200 kroků vývoje. Všimněte si, že se v obrázku opakují podobné geometrické útvary, ale nejdná se o periodické chování.

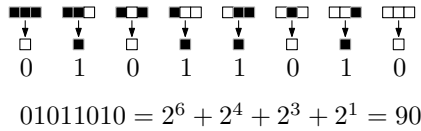


Obr. 44: 1D buněčný automat na mřížce délky 100 s periodickými podmínkami a s pravidlem (68). Na počátku má jeden bod hodnotu 1 a ostatní 0 (počáteční stav není vyobrazen). Na svislé ose je čas (rostoucí směrem nahoru), na vodorovné index  $i$ .

Jednorozměrné automaty je pro jejich jednoduchost poměrně snadné klasifikovat. Budeme uvažovat  $O(i) = \{i - 1, i, i + 1\}$ , tedy do okolí zahrnujeme i buňku na aktuální pozici. Také se omezíme na binární množinu  $S$ . Potom může celé okolí nabývat osmi možných stavů: 111, 110, 101, 100, 011, 010, 001, 000. Pro každý z těchto stavů existují dvě možnosti, jak se bude buňka  $i$  vyvíjet: buďto bude mít v příštím kroce hodnotu 1, nebo 0. Celkem tedy existuje 256 různých pravidel<sup>63</sup>

<sup>63</sup>S využitím symetrií tento počet můžeme redukovat na 32.

pro automaty uvedeného typu (občas se nazývají *elementární buněčné automaty*). Pravidlo (68) potom nazveme „pravidlo 90“, jak je názorně ukázáno na obrázku 45.



Obr. 45: Názorná ukázka klasifikace pravidla daného rovnicí (68).

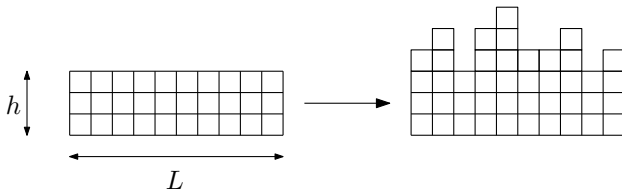
Podívejme se, jak se to má s počtem automatů ve více dimenzích. Pokud budeme za okolí  $O(i)$  opět považovat pouze nejbližší sousedy a samotnou buňku  $i$ , bude okolí v  $d$  dimenzích obsahovat  $N = 2d + 1$  buněk. Velikost množiny stavů je obecně  $|S| = q$ , počet možných konfigurací na okolí je tedy  $C = q^N$ . Z každé konfigurace může vzejít  $q$  různých stavů buňky  $i$ , a proto celkový počet pravidel bude v tomto obecném případě

$$N_{CA} = q^C = q^{2d+1}. \quad (69)$$

Tak například pro dvourozměrný binární buněčný automat existuje celkem  $2^{2^{4+1}} = 4\,294\,967\,296$  pravidel. Při takovém počtu nelze pravidla klasifikovat jedno po druhém, smysl má pouze dívat se na statistické chování. Prozatím proto zůstaneme v 1D světě a podíváme se na jednu zajímavou fyzikální aplikaci.

#### Využití CA pro modelování růstu povrchů

Představme si, že máme kovovou destičku, jejíž povrch je dokonale rovný. Tedy, na jejím povrchu jsou atomy uspořádány jeden vedle druhého, jak je ukázáno v levé části obrázku 46. Na tuto destičku naprašíme<sup>64</sup> několik vrstev stejného materiálu. Nový materiál se však neusazuje zcela rovnoměrně, po naprašování vypadá nanesená vrstva například jako na pravé části obrázku 46. Destička není již tak hladká jako na počátku – říkáme, že se zvýšila *hrubost povrchu*.



Obr. 46: Ukázka hrubnutí 1D povrchu.

<sup>64</sup>Naprašování je jednou z metod epitaxe neboli depozice tenkých vrstev. Zdrojový materiál po malém množství převádíme do plynného stavu a částice plynu pomocí elektromagnetického pole deponujeme na destičku.

Nechť kovová mřížka destičky neobsahuje žádné dislokace. Potom se na ni můžeme dívat jako na jednotlivé sloupečky. Výšku sloupečků částic značme  $h$ . Hrubost povrchu definujeme jako odmocninu druhého centrálního momentu výšky<sup>65</sup>

$$W(t, L) = \sqrt{E(h^2) - (E(h))^2}. \quad (70)$$

Rozepsáno podrobněji,

$$W(t, L) = \sqrt{\frac{1}{L^d} \sum_i h_i^2 - \left( \frac{1}{L^d} \sum_i h_i \right)^2}, \quad (71)$$

kde  $t$  je čas,  $L$  je charakteristický lineární rozměr povrchu,  $d$  je jeho dimenze a  $h_i$  výška sloupečku s indexem  $i$ . Budeme-li se dívat na řez destičkou, bude dimenze povrchu  $d = 1$  a  $L$  bude délka řezu (počet sloupečků). Pokud navíc budeme pokládat rozměr molekuly za jednotkový, udává  $h$  počet molekul v každém sloupečku a  $L$  je počet molekul v řadě.

S časem a přibývajícimi částicemi na povrchu hrubost roste. Ale jak? To záleží na způsobu depozice částic, respektive na metodě, kterou ho budeme modelovat. My budeme k simulaci používat 1D asynchronní buněčný automat, přičemž počet buněk bude  $L$  a výšky sloupců částic budou představovat jejich stavy. Formálně je tedy množina stavů  $S = \mathbb{N}$  neomezená, prakticky ale samozřejmě nevytvoříme destičku nekonečné tloušťky.

Pravidla evoluce automatu pro simulaci růstu hrubosti mohou být různá. Nejjednodušším pravidlem je *náhodná depozice*, tj. v každém kroku (máme asynchronní automat, krok znamená, že přibude jedna částice, ne celá vrstva) vybereme jednu buňku a zvýšíme její hodnotu o jedna. Náhodný výběr děláme z rovnoměrného rozdělení na  $L$ , není-li fyzikální důvod provádět výběr jinak.

Dalším používaným modelem růstu je *balistická depozice*. Opět vybíráme rovnoměrně náhodně buňky, které budou růst, ale v tomto případě se pomyslná padající částice nemusí zastavit až na vybraném sloupečku, ale může se „přilepit“ na sloupeček sousední. Při vybrání buňky  $i$  tedy můžeme formulovat evoluční vztah (s vhodnou úpravou na okraji)

$$h_i(t+1) = \max(h_{i-1}(t), h_i(t) + 1, h_{i+1}(t)). \quad (72)$$

Ve struktuře destičky tedy vznikají dutiny, automat si je ale nepoznamenává.

Názorná ukázka toho, jak se chovají uvedené dva modely, je na obrázku 47.

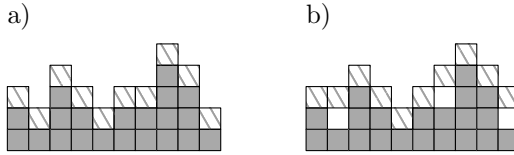
### Další příklady CA

V povídání o 1D automatech jsme se nevěnovali příliš jednotlivým pravidlům, byt jsou některé z nich nepochybně zajímavé. Například<sup>66</sup> u již zmíněného pravidla 90 existují pro každou konfiguraci čtyři možní předchůdci (zde uvažujeme mřížku bez

<sup>65</sup>Směrodatná odchylka výšky, chcete-li.

<sup>66</sup>Mějte na paměti, že ke každému příkladu existuje několik symetrických automatů, striktně řečeno tedy nejsou unikátní.





Obr. 47: a) Model náhodné depozice, b) model balistické depozice. **Nejedná se o časový vývoj.** Schémata pouze zobrazují, kam dopadne nová (šrafovaná) částice, bude-li vybrán určitý sloupec/buňka. Kdyby se jednalo o evoluci, mohly by se nové buňky přilepovat i vzájemně na sebe.

okrajů, nekonečnou; tvar okrajových podmínek má taky vliv na počet předchůdců), zatímco u některých jiných CA existují konfigurace, ke kterým se nelze evolucí dostat. Pravidlo 110 zase plynule přechází mezi chaotickými a uspořádanými oblastmi v časo-prostorovém diagramu a navíc je turingovskuy úplné.<sup>67</sup>

Mnohem zajímavější chování nalezneme mezi 2D automaty. Již výše zmíněná Hra života (Game of Life, GoL) je synchronní, deterministický, binární CA s následujícím pravidlem:

$$s_i(t+1) = 1 \quad \text{pro } 1 < |O(i)| < 4 \\ = 0 \quad \text{jinak.} \quad (73)$$

Název Hra života plyne z představy, že buňka s více než třemi sousedy umírá v důsledku „přelidnění“ a buňka s jedním nebo žádným sousedem umírá z osamění. Toto jednoduché pravidlo vede ke vzniku mnoha statických i dynamických vzorů, navíc je GoL dokáže nekonečně replikovat. Na rozdíl od pravidla 90 existují v GoL konfigurace, ke kterým není možno dospět evolucí – takové počáteční stavy se obvykle velice rychle redukují na základní objekty. Také nelze na základě počátečního stavu jednoznačně předpovědět, které objekty budou během evoluce vznikat. To mimo jiné plyne z toho, že GoL je podobně jako pravidlo 110 turingovskuy úplná (kdyby šlo bez proběhnutí evoluce předpovědět výsledek, byla by GoL „silnější“ než univerzální Turingův stroj, což není možné). Nejlépe se s GoL seznámíte, když si sami vyzkoušíte nechat vyvíjet některé počáteční stavy. Ke hraní doporučujeme online simulátor <https://bitstorm.org/gameoflife/>.

Hra života je zajímavá, ale není vidět její přímý užitek. Za užitečné můžeme považovat například modelování tvorby dálničního provozu. Úplně nejjednodušším modelem je pravidlo 184, kde buňka s hodnotou 1 představuje auto a buňka s hodnotou 0 prázdné místo. Pokud je napravo od auta prázdné místo, tak se pohne, jinak stojí (rozmyslete si, jak formálně toto pravidlo může vypadat, případně si to dopočtete pomocí schématu). Pokročilejší model je Nagelův-Schreckenbergův, stochastický 1D celulární automat s proměnlivou rychlostí „auta“. Nebudeme ho

<sup>67</sup> Být turingovskuy úplný ve stručnosti znamená být schopný řešit všechny úlohy, které dokáže řešit jakýkoli jiný stroj. Neznamená to ale řešit efektivně, viz 8-bitové procesory sestavené v Minecraftu...

zde podrobně rozebírat, pouze uvedeme zajímavý výsledek: kolona se s průběhem času posouvá proti směru rychlosti aut.

Pomocí buněčných automatů dokážeme simulovat (a bylo to nakonec i naší motivací) také problémy, které nejde dobře fyzikálně uchopit. Pěkným příkladem je CA model šíření lesních požárů. Jedná se o 2D stochastický, synchronní automat, v němž každá buňka nabývá několika stavů z množiny hodnot mladý strom, starý strom, hořící strom, spáleniště (dělení může být i podrobnější), přičemž požár se s nejvyšší pravděpodobností rozšíří na starý strom, s menší na mladý strom a nemůže se rozšířit na spáleniště. Ze spáleniště se v každém časovém kroku může s určitou pravděpodobností stát mladý strom. V případě tohoto automatu závisí výsledek velice silně na poměrech zadaných pravděpodobností. K vyhrání si doporučujeme online simulátor <http://www.eddaardvark.co.uk/svg/forest/forest.html>.

Celulární automaty se také hodí ke studiu kritických jevů, jakými jsou například fázové přechody. O tom ale až příště.

## Úloha IV.S ... kořeni a automati

10 bodů

- Nalezněte všechny (tři) reálné kořeny funkce  $\exp(x) - 5x^2$ . Výběr metody je na vás. Nezapomeňte okomentovat, jak a proč jste zvolili daný postup.
- Newtonova metoda tak, jak jsme si ji představili funguje i pro funkce komplexní proměnné. Vaším úkolem je vykreslit tzv. Newtonovy fraktály, tedy oblasti v komplexní rovině takové, že když v nich zvolíme počáteční odhad kořenu pro Newtonovu metodu, tak dokonvergujeme k určitému kořenu. Fraktál vykreslete pro funkce  $z^3 - 1$  a  $z^6 + z^3 - 1$ , kde  $z$  je komplexní číslo. Derivace těchto funkcí jsou  $3z^2$ , resp.  $6z^5 + 3z^2$ . Pro výpočet a vykreslení můžete použít Pythonní kód přiložený k zadání.

*Poznámka* Komplexní derivaci, pokud existuje, lze technicky spočítat stejně, jako reálnou derivaci, tedy pro ni platí stejné vzorce pro derivaci součtu, součinu a složené funkce.

*Bonus* Nalezněte co nejzajímavější nebo nejhezčí Newtonův fraktál.

- Simulujte na počítači (nebo napočítejte ručně) elementární buněčný automat s pravidlem 54 na mřížce délky 20 s periodickými podmínkami alespoň na 10 časových kroků (více určitě neuškodí). Na počátku má jedna libovolná buňka hodnotu 1 a zbylé 0. Výsledek zobrazte v časoprostorovém diagramu (jako na obr. 1 z textu seriálu).
- Simulujte hrubnutí 1D povrchu pomocí modelu náhodné depozice popsaném v seriálu. Povrch má rozměr  $L = 100$ , na počátku je zcela hladký. Nakreslete graf závislosti hrubosti  $W$  na čase pro alespoň  $10^8$  kroků (jeden krok = jedna nová částice), výsledek diskutujte.

(řešení str. 259)

## Kapitola 5: Rostou nám diferenciální rovnice

V tomto díle seriálu pokročíme k numerickému tématu, které je ve fyzice všudypřítomné: řešení obyčejných diferenciálních rovnic (ODR). Ačkoli nejsou tak složité

jako parciální diferenciální rovnice, vyžaduje jejich analytické řešení v mnoha případech pokročilé matematické metody. Numerika nám umožní se matematickým komplikacím vyhnout – pomocí diskretizace derivací získáme integrační schémata, pomocí nichž dokážeme vyřešit každou ODR se zvolenou přesností.

Simulační část seriálu nepokročí, ale naopak zůstane u tématu z minulého dílu. Ukážeme si, že jiné modely růstu povrchů vedou na kvalitativně odlišný vývoj hrubosti povrchu a také si předvedeme, jak použít růstové modely v mikrobiologii. Poté se vrátíme k náhodným procházkám a využijeme je k simulaci růstu fraktálního krystalu.

### Obyčejné diferenciální rovnice

V tomto díle se naučíme základy numerického řešení obyčejných diferenciálních rovnic. Jde o poměrně atraktivní problematiku, ostatně mnoha lidem se při vyslovení pojmu „počítačové simulace“ vybaví právě řešení diferenciálních rovnic. My již víme, že numerická matematika je daleko rozsáhlejším oborem, to ale nijak nesnižuje význam a potřebu numerického řešení diferenciálních rovnic. Nicméně jde o zvláště zákeřnou oblast numerické matematiky, narazit zde na nestabilní řešení není nic výjimečného a opravdu platí rčení: „Důvěřuj, ale proveď!“

Diferenciální rovnice jsou jednoduše řečeno takové rovnice, ve kterých vystupuje derivace. Na rozdíl od algebraických rovnic, jejichž řešením je číslo, řešením diferenciálních rovnic je funkce. Příkladem diferenciální rovnice je například pohybová rovnice svislého vrhu  $\ddot{y} = -g$ , kde  $\ddot{y}$  značí druhou derivaci  $y(t)$  vůči času. Řešením této rovnice pak je funkce  $y(t) = y_0 + v_0 t - \frac{1}{2} g t^2$ , kde  $y_0$  a  $v_0$  jsou tzv. integrační konstanty, tedy volné parametry řešení. V případě svislého vrhu pak mají jasný fyzikální význam počáteční polohy a rychlosti. Sami si pak můžete snadno ověřit, že druhá derivace dané funkce podle času je skutečně rovna  $-g$ . Pokud k zadání přidáme i tzv. počáteční podmínky, např.  $y(0) = 1$  a  $\dot{y}(0) = 0$ , dokážeme jednoznačně určit i hodnotu integračních konstant. V našem případě pak řešení bude  $y(t) = 1 - \frac{1}{2} g t^2$ . Platí přitom, že počátečních podmínek potřebujeme tolik, jaký je řád diferenciální rovnice, tedy řád nejvyšší derivace v ní vystupující. V našem případě rovnice obsahuje druhou derivaci, rovnice je tedy druhého řádu a potřebujeme dvě nezávislé počáteční podmínky. Takto přesně analyticky vyřešit ovšem umíme pouze některé jednoduché rovnice, na zbytek musíme použít aproximace, rozvoje, nebo právě numerické metody.

Diferenciální rovnice dělíme na *obyčejné* (zkr. ODR), které obsahují neznámou funkci pouze jedné proměnné a derivace podle této proměnné, a *parciální* (zkr. PDR), obsahující neznámou funkci více proměnných a (parciální) derivace podle nich. Řešení parciálních diferenciálních rovnic je zpravidla složitější úloha, než řešení ODR, a to i při numerickém přístupu, nadále se tedy budeme zabývat pouze řešením ODR. Konkrétně budeme hledat řešení soustavy  $N$  rovnic, na které můžeme nahlédnout jako na vektorovou rovnici

$$\dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t), t)$$

s počáteční podmínkou  $\mathbf{u}(0) = \mathbf{u}_0$ . To, že jde o soustavu rovnic, se prakticky projevuje pouze tak, že všechny kroky řešení neprovedeme jednou, ale  $N$ -krát, tedy pro

každou z rovnic soustavy. Při popisu metod si tedy nemusíte zadání představovat jako soustavu, ale jen jako jednu rovnici. Tento přístup má dokonce jisté výhody, jak uvidíme při praktické implementaci metod. Ač to tak na první pohled nevypadá, do výše uvedeného tvaru lze převést naprostou většinu soustav obyčejných diferenciálních rovnic libovolného řádu,<sup>68</sup> nijak se tedy v úloze neomezujeme. Jak tento převod prakticky provést si ukážeme na rovnici  $\ddot{y} = -g$ . Nejprve zavedeme substituci  $\dot{y}(t) = v(t)$  a dosadíme do původní rovnice, čímž získáme  $\dot{v} = -g$ . Máme tedy soustavu dvou rovnic prvního řádu  $\dot{y} = v$  a  $\dot{v} = -g$  s neznámými  $y(t)$  a  $v(t)$ . Nyní jsme připraveni k popisu samotných metod řešení, ještě předtím si ale pokusíme osvojit dovednost pro fyzika možná i důležitější, a to, jak formulovat problém ve formě diferenciální rovnice.

Derivace nějaké veličiny podle času vyjadřuje rychlost změny této veličiny. Jako příklad si můžeme vzít model radioaktivity, kdy z fyziky víme, že za jednotku času se rozpadne takové množství atomů, které je přímo úměrné počtu nerozpadlých atomů. Příslušná diferenciální rovnice tedy je

$$\frac{dN(t)}{dt} = -\lambda N(t)$$

s počáteční podmínkou  $N(0) = N_0$ , kde  $N(t)$  je počet ještě nerozpadlých atomů v čase  $t$  a  $\lambda$  je rozpadová konstanta vyjadřující počet přeměn za jednotku času přepočtený na jeden atom. Mínus pak značí, že nerozpadlé atomy postupně ubývají. Řešením této rovnice je pak známá rovnice radioaktivní přeměny  $N(t) = N_0 \exp(-\lambda t)$ .

Dalším důležitým příkladem jsou pohybové rovnice vycházející z druhého Newtonova zákona, který je vlastně diferenciální rovnicí  $\dot{\mathbf{p}} = \mathbf{F}$ , slovně: „Časová změna hybnosti hmotného bodu je v každém okamžiku rovna výslednici sil na tento bod působících“. Pokud budeme uvažovat konstantní hmotnost, pak dostáváme méně obecnou verzi  $\dot{\mathbf{p}} = (\dot{m}\mathbf{v}) = m\dot{\mathbf{v}} = m\dot{\mathbf{x}} = \mathbf{F}$ , tedy  $\dot{\mathbf{x}} = \mathbf{F}/m$ . Nyní stačí dosadit za sílu dle konkrétního fyzikálního problému a získáme soustavu pohybových rovnic daného hmotného bodu. Například pro šikmý vrh v homogenním gravitačním poli bude síla za použití kartézských souřadnic rovna  $\mathbf{F} = (0, 0, -g)$ . Musíme pak ještě samozřejmě zadat 6 počátečních podmínek, např. polohy a rychlosti na počátku a rovnice případně převést na soustavu rovnic prvního řádu, jak bylo ukázáno výše.

### Eulerova metoda

První a nejjednodušší metodou, kterou si zde představíme, je explicitní Eulerova metoda. Vyjdeme ze soustavy  $\dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t), t)$  a derivaci nahradíme dopřednou diferencí. Po úpravě pak dostaneme

$$\frac{\mathbf{u}(t+h) - \mathbf{u}(t)}{h} = \mathbf{f}(\mathbf{u}(t), t) + O(h),$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{f}(\mathbf{u}_n, t_n) + O(h^2),$$

<sup>68</sup>Snad s výjimkou některých obzvlášť ošklivých rovnic, které jsou nelineární v nejvyšším řádu derivace.

kde jsme označili  $t_n \stackrel{\text{def}}{=} nh$  (bez újmy na obecnosti můžeme začít v čase  $t_0 = 0$ ),  $\mathbf{u}_n \stackrel{\text{def}}{=} \mathbf{u}(t_n)$  a  $\mathbf{u}_{n+1} \stackrel{\text{def}}{=} \mathbf{u}(t_{n+1})$ . Nyní již máme přímočarý návod k řešení diferenciální rovnice. Vezmeme počáteční podmínku  $\mathbf{u}_0$  a dosazením  $\mathbf{u}_n = \mathbf{u}_0$  a  $t_n = 0$  do rovnice výše vypočteme  $\mathbf{u}_1$ , to opět dosadíme (nyní s časem  $t_1 = h$ ) a iterujeme, dokud nezískáme hodnoty až do kýženého času. Metodu lze implementovat například takto:

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

def euler_step(f,y,t,h):
    return y+h*f(y,t)

def ODE_solve(f,y0,t0,tf,steps):
    times = np.linspace(t0,tf,steps)
    h = (tf-t0)/steps
    result = []
    y = y0
    for t in times:
        result.append(y)
        y = euler_step(f,y,t,h)
    return times,np.array(result)

def vrh(y,t):
    return np.array((y[2],y[3],0.0,-9.81))

X0,Y0 = 0.0,0.0
VX0,VY0 = 5.0,10.0
t0,tf,steps = 0.0,10.0,1000
y0 = np.array((X0,Y0,VX0,VY0))

times, res = ODE_solve(vrh, y0, t0, tf, steps)
plt.plot(times, res[:,1]) #vykresli Y(t)
plt.show()
```

Všimněme si, že veškerá fyzika je obsažena v definici funkce  $f(u, t)$  (v našem případě za ni dosazujeme funkci `vrh()`) a v počátečních podmínkách. Toto oddělení fyziky a numerického řešiče (integrátoru) umožňuje flexibilně používat stejný integrátor pro různé problémy a naopak.

Eulerova metoda funguje na principu lineární extrapolace, kdy v bodě  $(t_n, u_n)$  sestrojíme tečnu k řešení  $u$  (její směrnice je vyjádřena hodnotou funkce  $f(u_n, t_n)$ ) a posuneme se podél této tečny o časový krok  $h$  do bodu  $(t_{n+1}, u_{n+1})$ . Je jasné, že jde pouze o hrubou aproximaci skutečného řešení, pro dostatečnou přesnost tedy musíme volit poměrně malý krok  $h$ .

### Rungovy-Kuttovy metody

Přesnost dokážeme vylepšit například s použitím Rungových-Kuttových metod vyššího řádu. Předpis R-K metody  $s$ -tého řádu má tvar

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i, \text{ kde}$$

$$k_i = f \left( u_n + h \sum_{j=1}^s a_{ij} k_j, t_n + c_i h \right),$$

přičemž  $a_{ij}$ ,  $b_i$  a  $c_i$  jsou číselné koeficienty příslušející dané metodě. Často jsou v literatuře zapisovány ve formě tzv. *Butcherovy tabulky*. Všimněme si, že pro  $j \geq i$  potřebujeme znát hodnoty  $k_j$ , které při postupném vyhodnocování ještě neznáme. Tyto metody se nazývají *implicitní* a mají určité nemalé výhody, které si zmíníme později, ty jsou však vykoupeny složitějším způsobem vyhodnocování. Tento problém ale zmizí, pokud pro  $j \geq i$  je  $a_{ij} = 0$ . Pak můžeme bez problémů vyčíslit  $k_1$ , poté  $k_2, \dots, k_s$  a nakonec  $u_{n+1}$ . Takové metody se nazývají *explicitní*. Zkusme nyní dosadit  $s = 1$ ,  $a_{11} = 0$ ,  $b_1 = 1$  a  $c_1 = 0$ . Pak dostaneme nám již známý předpis explicitní Eulerovy metody  $u_{n+1} = u_n + hf(u_n, t_n)$ . Explicitní Eulerova metoda je tedy Rungovou-Kuttovou metodou 1. řádu.

Pokusme se nyní ilustrovat, jak Rungovy-Kuttovy metody fungují, a to na příkladu R-K metody 4. řádu, která je jednou z nejpoužívanějších. Její předpis je

$$k_1 = f(u_n, t_n),$$

$$k_2 = f \left( u_n + \frac{h}{2} k_1, t_n + \frac{h}{2} \right),$$

$$k_3 = f \left( u_n + \frac{h}{2} k_2, t_n + \frac{h}{2} \right),$$

$$k_4 = f(u_n + hk_3, t_n + h),$$

$$u_{n+1} = u_n + \frac{1}{6} h k_1 + \frac{1}{3} h k_2 + \frac{1}{3} h k_3 + \frac{1}{6} h k_4.$$

Nejprve vypočteme směrnicí  $k_1$  v bodě  $(t_n, u_n)$ , podobně, jako v případě Eulerovy metody. Dále ale vypočteme další směrnice  $k_i$  v mezibodech určených předchozími směrnicemi. Nakonec všechny směrnice zprůměrujeme s různými vahami a posuneme se podél této výsledné směrnice o krok  $h$  z počátečního bodu kroku  $(t_n, u_n)$ . Lze intuitivně předpokládat, že toto odhadování v mezibodech a následné průměrování je přesnější, než prostý dlouhý skok jako v Eulerově metodě, matematický důkaz zde ale nevedeme. Dále by se mohlo zdát, že by bylo jednodušší a možná i přesnější použít Eulerovu metodu s třeba čtyřikrát kratším krokem. Vtip je v tom, že R-K metoda  $s$ -tého řádu má globální zaokrouhlovací chybu (chybu nasčítanou přes všechny kroky) řádu  $O(h^s)$ . Pro velmi velký krok tedy bude skutečně Eulerova metoda lepší,<sup>69</sup> ale pro menší krok (prakticky pro každý rozumně velký krok) budou

<sup>69</sup>V takovém případě by byla lepší už z toho důvodu, že by R-K metody vyššího řádu byly nestabilní, tj. nevedly by ke správnému řešení.

R-K metody vyššího řádu přesnější. V praxi se pak často používá zmíněná R-K metoda 4. řádu, která představuje rozumný kompromis mezi přesností (zlepšující se s řádem) a stabilitou a počtem vyčíslení funkce  $f$  v jednom kroku (což jsou s řádem se zhoršující vlastnosti). Dále je používána například Dormandova-Princeova metoda, což je jedna z adaptivních R-K metod, tedy metod, které dokáží odhadnout chybu v daném kroku a na jejím základě adaptivně upravovat délku kroku v průběhu výpočtu.

Tab. 14: Butcherova tabulka

$c_1$	$a_{11}$	$a_{12}$	$\cdots$	$a_{1s}$
$c_2$	$a_{21}$	$a_{22}$	$\cdots$	$a_{2s}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$c_s$	$a_{s1}$	$a_{s2}$	$\cdots$	$a_{ss}$
	$b_1$	$b_2$	$\cdots$	$b_s$

### Lineární víceukrové metody

Dalším typem pokročilejších metod jsou lineární víceukrové metody. Metody, které jsme doposud poznali, byly jednokrové, protože pro výpočet stavu  $u_{n+1}$  stačilo znát pouze stav  $u_n$ , starší stavy jsme znát nepotřebovali. Myšlenka víceukrových metod pak spočívá v tom, že tyto starší stavy známe a s jejich pomocí lépe odhadneme vývoj do následujícího kroku než pouze se znalostí stavu a jeho derivace v jednom bodě. Předpis obecné lineární  $s$ -krové metody je

$$\sum_{j=0}^s \alpha_j u_{n+j} = h \sum_{j=0}^s \beta_j f(u_{n+j}, t_{n+j}),$$

kde  $\alpha_j$  a  $\beta_j$  jsou koeficienty a platí  $\alpha_s = 1$ ,  $\alpha_0 \neq 0$  a  $\beta_0 \neq 0$  (pomocí  $u_n, \dots, u_{n+s-1}$  a  $f(u_n, t_n), \dots, f(u_{n+s}, t_{n+s})$  tedy chceme spočítat  $u_{n+s}$ ). Pokud navíc platí  $\beta_s = 0$ , je metoda explicitní, jinak je implicitní.

Praktickým problémem při implementaci víceukrových metod je jejich nastartování. Dokud jsme totiž neudělali  $s - 1$  kroků, nemůžeme metodu použít pro nedostatek známých  $u_i$ . Prvním řešením je, že máme k dispozici nadbytek počátečních podmínek, ideálně přímo ve formě potřebných  $u_i$ . To se stává zřídka, protože to většinou znamená znalost řešení ODR, kterou chceme řešit. Častějším způsobem je použití nějaké jednokrové metody pro prvních pár kroků. Dalším, hůře řešitelným problémem je, že víceukrové metody předpokládají stálý krok  $h$  po celou dobu výpočtu, nelze tedy použít adaptivní změnu kroku dle aktuálního odhadu chyby.

Ne všechny metody splňující toto schéma jsou automaticky použitelné. Samozřejmým požadavkem je, aby metoda dostatečně přesně aproximovala danou ODR, musí tedy mít malou chybu metody (chyba musí být dostatečného řádu  $v$   $h$ ). Ovšem

ani taková metoda nemusí automaticky fungovat dobře, protože metoda může neúměrně zvětšovat zaokrouhlovací chybu danou diskretizací problému, a to až tak, že řešení metody nebude vůbec kopírovat skutečné řešení. Říkáme pak, že je metoda *nestabilní*. Vyšetřování stability metod sahá nad rámec tohoto textu, spokojme se s tvrzením, že často používanými zástupci lineárních vícekrokových metod jsou explicitní Adamsovy-Bashforthovy metody a implicitní Adamsovy-Moultonovy metody, jejichž předpisy a vlastnosti (např. řád) najdete v literatuře.

### Problémy s problémy se silným tlumením

Uvažujme diferenciální rovnici ve tvaru

$$y' = -cy,$$

kde  $c > 0$  je reálná konstanta. Analytické řešení této rovnice je

$$y(x) = Ae^{-cx},$$

kde  $A$  je integrační konstanta, kterou určíme z počátečních podmínek. Vidíme, že funkce  $y(x)$  rychle klesá k nule. Takovému problému se říká problém se silným tlumením,<sup>70</sup> anglicky *stiff equation*. Pokusme se jej nyní vyřešit pomocí explicitní Eulerovy metody. Přímým dosazením za  $f(u, t)$  dostaneme

$$u_{n+1} = (1 - ch)u_n.$$

Pokud budeme volit  $h > 2/c$ , pak posloupnost  $|u_n| \rightarrow +\infty$  pro  $n \rightarrow +\infty$  pro libovolné počáteční podmínky, což rozhodně nedává správné řešení, metoda je *nestabilní*. Pokusme se nyní ten samý problém vyřešit s pomocí implicitní Eulerovy metody s předpisem  $u_{n+1} = u_n + hf(u_{n+1}, t_{n+1})$ , kde ale můžeme vyjádřit  $u_{n+1}$  explicitně díky tomu, že známe vztah pro  $f()$ . Po dosazení dostáváme

$$u_{n+1} = \frac{u_n}{1 + ch}.$$

V tomto případě pro libovolně velký krok  $h$  posloupnost  $u_n$  klesá k nule. Zdá se tedy (a prakticky to tak je), že použití implicitních metod by mohlo být univerzálním lékem na problém silného tlumení.

Ukázali jsme tedy, že za určitých (ne zcela vzácných) okolností jsou implicitní metody nedocenitelné. Zbývá vyřešit problém, jak je řešit, když se zdá, že potřebujeme znát  $u_{n+1}$  dříve, než jej vypočítáme (a nemůžeme využít trik s dosazením za  $f()$ ). První, co nás napadne, je použít iterativní metody, kdy nastřelíme nějaký odhad  $u_{n+1}^0$  a iterujeme předpisem implicitní metody, dokud  $u_{n+1}^i$  není dostatečně blízko  $u_{n+1}$  (horní index je jen index iterace, ne mocnina). Hodilo by se, aby už prvotní odhad byl dostatečně blízko, abychom dokonvergovali rychle, nabízí se tedy pro prvotní odhad použít explicitní metodu. Tím se dostáváme k metodám *prediktor-korektor*, které fungují přesně na tomto principu. Nejprve použijeme explicitní metodu (prediktor) pro určení hrubého nástřelu  $u_{n+1}$ , ten poté zpřesníme použitím jedné, nebo více iterací implicitní metody (korektor). Díky použití

---

<sup>70</sup>Název pochází z tlumení v harmonickém oscilátoru, které se projevuje právě takto.



implicitní metody máme prakticky zajištěnou stabilitu, nezdržujeme se ale příliš mnoha iteracemi (většinou stačí jedna), metoda je tedy rychlá. Při jejich použití je třeba dát pozor na správné namixování prediktoru a korektoru, mimo jiné na řád metod. Doporučujeme tedy používat nějaké standardní kombinace, či ještě lépe použít numerickou knihovnu, která má metody implementovány. V případě kombinace Adamsovy-Bashforthovy a Adamsovy-Moultonovy metody je takovou standardní kombinací, například

$$u_{n+2} = u_{n+1} + \frac{1}{2}h(3f(u_{n+1}, t_{n+1}) - f(u_n, t_n)),$$

$$u_{n+2} = u_{n+1} + \frac{h}{12}(5f(u_{n+2}, t_{n+2}) + 8f(u_{n+1}, t_{n+1}) - f(u_n, t_n)),$$

čímž dostaneme prediktor-korektor 3. řádu.

### Zachování energie a Verletův algoritmus

Jak jsme zmínili v úvodu (a po diskusi stability je nejspíš jasné), je třeba v průběhu výpočtu kontrolovat, jestli nám příliš neroste chyba. U fyzikálních dějů se nám často nabízejí různé zachovávající se veličiny, například u oběhu planety kolem Slunce se (mimo jiné) zachovává celková energie a moment hybnosti. Pokud si budeme tyto veličiny v průběhu výpočtu vypisovat, měly by být konstantní, samozřejmě až na nějakou numerickou chybu, která ale nesmí být velká.

Pokud ale řešíme diferenciální rovnici typu  $u'' = f(u, t)$ , což je mj. i častý případ mechanických pohybů, pak můžeme s výhodou použít metodu, která automaticky ze své podstaty zachovává energii. Napišme si Taylorův rozvoj  $u(t+h)$  a  $u(t-h)$  a sečteme je.

$$u(t+h) = u(t) + hu'(t) + \frac{h^2}{2}u''(t) + \frac{h^3}{6}u'''(t) + O(h^4)$$

$$u(t-h) = u(t) - hu'(t) + \frac{h^2}{2}u''(t) - \frac{h^3}{6}u'''(t) + O(h^4)$$

$$u(t+h) = 2u(t) - u(t-h) + h^2u''(t) + O(h^4)$$

Poslední rovnice nám již dává předpis pro Verletovu metodu

$$u_{n+1} = 2u_n - u_{n-1} + h^2f(u_n, t_n).$$

Důležitou vlastností Verletovy metody je, že je symetrická vůči směru toku času, neboli *časově reverzibilní*. Skutečně, pokud zaměníme  $h$  za  $-h$ , dostaneme

$$u_{n-1} = 2u_n - u_{n+1} + h^2f(u_n, t_n),$$

což po úpravě dává vztah výše. Díky této vlastnosti pak z důležité matematické věty, které se říká *teorém Noetherové*, plyne, že Verletův vztah zachovává energii, ovšem samozřejmě pouze za předpokladu, že energii zachovává i simulovaný děj, reprezentovaný funkcí  $f$ .

Nejspíš jste se již s teorémem Noetherové setkali v populární literatuře. Pak si nejspíš vzpomínáte na tvrzení, že každé symetrii odpovídá nějaká zachovávající se veličina. Zde je tou symetrií invariance vůči směru toku času, odpovídající se zachovávající veličinou je pak energie.

## Růst povrchů II – balistická depozice a Edenův model

Než začneme vysvětlovat nové pojmy, zopakujme si nejprve ve stručnosti poznatky z minulého dílu seriálu. Říkali jsme si, že růst povrchu lze modelovat pomocí stochastického vícestavového buněčného automatu. Automat je stochastický, protože buňku, která v daném kroce bude růst, vybíráme náhodně. Buňkou je zde myšlen jeden sloupeček molekul na mřížce. Že je automat vícestavový, znamená, že výška  $h_i$  sloupečku, kterou ztotožňujeme se stavem buňky, může nabývat během vývoje více než dvou různých hodnot ( $h_i \in \mathbb{N}$ ). Hrubost povrchu v čase  $t$  (po  $t$  krocích) na 1D mřížce velikosti  $L$  definujeme jako

$$W(t, L) = \sqrt{\frac{1}{L} \sum_i h_i^2 - \left( \frac{1}{L} \sum_i h_i \right)^2}. \quad (74)$$

Model náhodné depozice náhodně vybere buňku  $i$  a zvětší hodnotu jejího stavu o jedna. Balistická depozice má mírně složitější pravidlo,

$$h_i(t+1) = \max(h_{i-1}(t), h_i(t) + 1, h_{i+1}(t)). \quad (75)$$

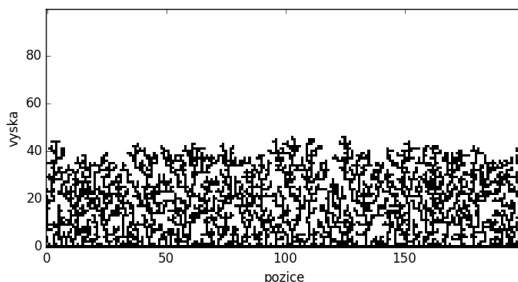
Růst povrchu modelovaný metodou balistické depozice si můžeme představit tak, že částice sestupují z nekonečna na povrch a zastaví se tehdy, když je nějaká částice ihned pod nimi, ale také tehdy, nachází-li se nějaká částice v sousední buňce nalevo nebo napravo. V materiálu tak vznikají dutiny, což se děje i ve skutečných experimentech, například při růstu povrchu pomocí naprašování (epitaxe). Porózní materiál má pak samozřejmě jiné vlastnosti. Ukázka materiálu vzniklého balistickou depozicí je na obrázku 48.

Balistická depozice<sup>71</sup> se však od náhodné depozice neliší pouze porozitou. Podívejme se, jak se vyvíjí hrubost povrchu. Na obrázku 49 je vykreslen vývoj hrubosti vygenerovaný pomocí kódu níže. Vidíme, že na počátku hrubost poměrně rychle narůstá, ale růst zpomaluje a po cca 10 000 krocích se zastaví – došlo k saturaci (nasyčení). Tento efekt jsme nepozorovali u náhodné depozice (viz řešení úlohy ke 4. dílu seriálu), kdy hrubost neustále rostla jako mocnina  $t^{1/2}$ . Obecně můžeme rozdělit vývoj hrubosti na tři časové úseky na základě času přechodu  $t_x$ :

1.  $t \ll t_x \rightarrow W(t, L) \sim t^\beta$ ;
2.  $t \approx t_x \rightarrow$  přechodová oblast, růst se zpomaluje;
3.  $t \gg t_x \rightarrow W(t, L) \sim L^\alpha$ .

Parametry  $\alpha$  a  $\beta$  se nazývají *kritické škálovací parametry*. Dále se zavádí *dynamický škálovací parametr*  $z = \alpha/\beta$ , který popisuje chování času přechodu jako  $t_x \sim L^z$ . Nalezení parametrů  $\alpha$  a  $\beta$  pro model balistické depozice bude vaším úkolem v seriálové úloze.

<sup>71</sup> Pokud budete hledat o balistické depozici něco na internetu, možná narazíte na jinou definici. Zde jsme definovali NN (nearest neighbour) depozici.



Obr. 48: Struktura povrchu získaného na základě modelu balistické depozice. Všimněte si vysoké porozity.

```

# BALISTICKA DEPOZICE
# nacteme grafickou knihovnu a numerickou knihovnu
import matplotlib as mpl
from matplotlib import pyplot as plt
import numpy as np

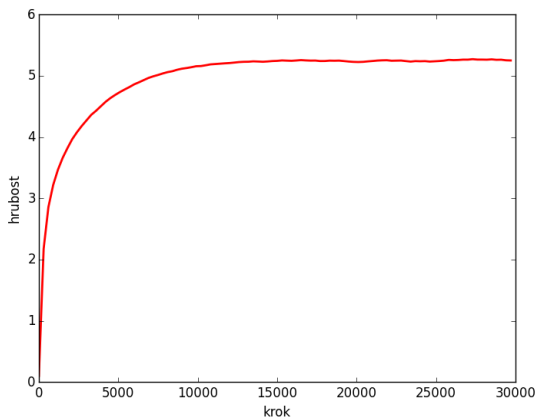
# definujeme rozmer mridky a pocet kroku a interval pro vypocet hrubosti
pocet = 100
roz = 100
krok = 30000
inter = 30
# vytvorime pole naplnene nulami pro stavy a pro hrubost
stav = np.zeros(roz,dtype=float)
hrub = np.zeros(krok/inter,dtype=float)
sumhrub = hrub

for j in range(pocet):
    # v kazdem kroku nahodne vybereme bunku a zvysime její hodnotu o 1
    hrub = np.zeros(krok/inter,dtype=float)
    stav = np.zeros(roz,dtype=float)
    for i in range(krok):
        nahod = np.random.randint(roz)
        stav[nahod] = max(stav[nahod]+1.,stav[(nahod-1) % roz],stav[(
            nahod+1) % roz])
        if i % inter == 0:
            hrub[i/inter] = np.sqrt(1./roz*np.sum(stav**2) - (1./
                roz*np.sum(stav))**2)
    sumhrub += hrub

sumhrub *= 1./pocet

# nakreslime graf vyvoje hrubosti
# funkce linspace rovnomerne rozdeli pocet kroku na intervaly

```

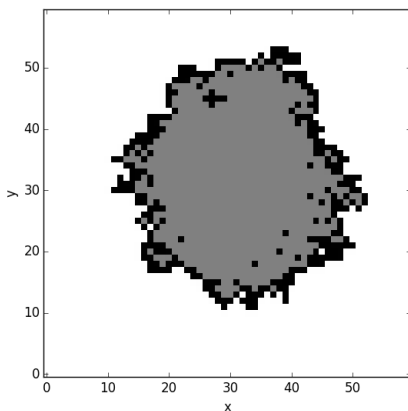


Obr. 49: Příklad vývoje hrubosti povrchu pro simulaci růstu metodou balistické depozice. Průměrováno přes 1000 běhů.

```
body = np.linspace(0,krok-inter,krok/inter)
hrubplot = plt.plot(body,sumhrub,'r',linewidth=2,label='hrubost')
plt.xlabel('krok')
plt.ylabel('hrubost')
plt.show()
```

Druhým růstovým modelem, o kterém se zde blíže zmíníme, je Edenův model. Jeho princip je jednoduchý – částice se může přilepit na libovolné místo na povrchu (tj. seshora i z boku sloupečku). Ale protože zde bude naším cílem simulovat růst bakteriálních kolonií, jak jsme již v úvodu předeslali, nemůžeme použít 1D mřížku. Model si proto představíme jako binární automat (pouze stavy 0 a 1) na 2D mřížce a povolíme růst ze všech směrů. Během simulace je tedy potřeba pamatovat si celý povrch, a to včetně dutin uvnitř (bakterie se mohou množit kdekoli, kde mají životní prostor). Správným přístupem je ukládat si souřadnice buněk na povrchu, ale programátorsky jednodušší je vytvořit pole  $N \times N$ , kde  $N$  je zvolený rozměr mřížky, a ukládat hodnotu každé buňky. Výpočet je pak samozřejmě pomalý a musíme se omezit na malé mřížky. Příklad povrchu vytvořeného pomocí Edenova modelu vidíte na obrázku 50. K vygenerování byl použit kód níže.

Edenův model přiřazuje růstu uvnitř dutin v materiálu stejnou pravděpodobnost jako růstu na volném povrchu. Porozita tak jednak zvyšuje celkovou délku povrchu, kterou musíme udržovat v paměti, a také nemusí odpovídat skutečnému chování fyzikálního systému, který studujeme. Proto se často zavádějí úpravy modelu, které zvyšují pravděpodobnost zaplňování dutiny úměrně její vzdálenosti od povrchu. Tato úprava samozřejmě ovlivní hodnoty škálovacích koeficientů ve vztazích pro hrubost, ale ne příliš výrazně.



Obr. 50: Výsledek simulace růstu podle Edenova modelu po 2000 krocích. Šedou barvou jsou označeny ty buňky, které byly náhodně vybrány, ale už neměly neobsazené sousedy (bylo by možné vykreslit je také černou barvou, jde jen o ukázkou toho, jak funguje použitý algoritmus). Porozita je v porovnání s balistickou depozicí minimální.

```
# EDENUV MODEL
# nacteme grafickou knihovnu a numerickou knihovnu
import matplotlib as mpl
from matplotlib import pyplot as plt
import numpy as np

# definujeme rozmer mřížky a počet kroku
roz = 200
krok = 20000
# vytvořime pole nul, do kterého budeme ukládat stavy bunek
# zavorka (kroky, rozmer) udava velikost 2D pole
pole = np.zeros((roz, roz), dtype=np.int)
# inicializujeme
pole[100,100] = 1

for i in range(krok):
    # funkce where vrati indexy prvku, které mají hodnotu jedna,
    # ve formě seznamu obsahujícího dvě pole, kde každé z těchto poli
    # obsahuje příslušné x,y souřadnice
    indexy = np.where(pole == 1)
    # náhodně vybereme index
    rand = np.random.randint(len(indexy[0]))
    ind = [(indexy[0])[rand], (indexy[1])[rand]]
    # prozkoumáme okolí odpovídajícího prvku
    soused1 = pole[(ind[0]+1) % roz, ind[1]]
    soused2 = pole[ind[0], (ind[1]+1) % roz]
    soused3 = pole[(ind[0]-1) % roz, ind[1]]
    soused4 = pole[ind[0], (ind[1]-1) % roz]
```

```

sousedni = np.array([soused1,soused2,soused3,soused4])
sous_indexy = np.where(sousedni == 0)
# pokud jsou sousedi zaplneni, nastavime pole[ind] na 2,
# abychom ho jiz priste neprohledavali
if len(sous_indexy[0]) == 0:
    pole[ind[0],ind[1]] = 2
# nahodne vybereme prazdnou sousedni bunku a zaplnime ji
else:
    sous_ind = (sous_indexy[0])[np.random.randint(len(sous_indexy
[0]))]
    if sous_ind == 0:
        pole[(ind[0]+1) % roz,ind[1]] = 1
    elif sous_ind == 1:
        pole[ind[0],(ind[1]+1) % roz] = 1
    elif sous_ind == 2:
        pole[(ind[0]-1) % roz,ind[1]] = 1
    elif sous_ind == 3:
        pole[ind[0],(ind[1]-1) % roz] = 1

# definujeme barevne schema pomoci vycytu, v nasem pripade cernobile
barvy = mpl.colors.ListedColormap(['white','black','grey'])
# definujeme prechod mezi barvami kdekoli mezi 0 a 1
hranice=[0,0.5,1.5,2]
# pouzijeme barvy a hranice k normovani barevne mapy
norma = mpl.colors.BoundaryNorm(hranice, barvy.N)
# vykreslime barevnou mapu a zobrazime ji
mapa = plt.imshow(pole, cmap=barvy, interpolation='None', norm=norma, origin='
lower')
plt.xlabel('x')
plt.ylabel('y')
plt.show()

```

Kvůli radiální geometrii nelze přímo aplikovat vzorec pro hrubost (74). Proto se nebudeme hrubostí zabývat, ale zavedeme do modelu malou změnu: s pravděpodobností  $p_1$  změníme stav buňky na rozhraní z 0 (prázdná) na 1 (obsazená) a s pravděpodobností  $p_2$  naopak změníme stav buňky na rozhraní z 1 na 0. Tento model, někdy nazývaný Williams-Bjerknesův, je vhodný pro simulaci růstu nádorových buněk nebo bakterií v Petriho misce. V seriálové úloze bude vaším úkolem zkoumat vývoj nádorových buněk pomocí právě popsaného modelu v závislosti na  $p_1, p_2$ .

Pro úplnost uvedme, že modelovat růst povrchů lze i jinými metodami než pomocí buněčných automatů. Ke každému zmíněnému růstovému modelu existuje ekvivalentní diferenciální rovnice, kterou lze v některých případech řešit analyticky a získat tak stejné škálovací parametry jako numerickým modelováním. Jedná se však o stochastické diferenciální rovnice, tj. rovnice obsahující člen, který reprezentuje šum. K řešení takových rovnic je potřeba zavést speciální integrální počet, tzv. Itův stochastický kalkulus.

### DLA – difúzí limitovaná agregace

Zatím zmíněné růstové modely dokázaly poměrně dobře simulovat laboratorní tvorbu materiálů nebo růst bakteriálních kolonií. Pokud bychom ale chtěli modelovat

růst krystalu vznikajícího v roztocích, např. pomocí elektrolýzy nebo sedimentací, bude lepší volbou model DLA (Diffusion limited aggregation).

Fyzikální představa procesu je následující: částice z nekonečna se brownovsky pohybují prostorem, dokud nedojde k jejich kontaktu s krystalizačním jádrem, ke kterému se přilepí. Hustota částic je malá, vzájemně spolu nekolidují (brownovský pohyb zajišťují molekuly v pozadí, které na jádru neulpívají). V počítačové praxi nemůžeme poslat částici z nekonečna, vytvoříme ji proto v určité vzdálenosti od krystalu a necháme ji pohybovat se tak dlouho, dokud nedojde ke kolizi s krystalem, nebo se nedostane za určitou radiální hranici – potom částici přestaneme sledovat a vytvoříme novou.

Níže je příklad kódu (výsledek simulace je na obrázku 51), kde máme jednu částici v počátku hexagonální mřížky a nové částice generujeme ve vzdálenosti, která je rovna radiální vzdálenosti od počátku nejbližší částice krystalu (plus jedna). Částici přestaneme sledovat, pokud se vzdálí na dvojnásobek počáteční vzdálenosti. Snížíme-li maximální povolenou vzdálenost, běh kódu se výrazně zrychlí, ale krystal potom začne vykazovat preferovaný směr růstu. Kód je poměrně komplikovaný kvůli implementaci šestiúhelníkové mřížky. V seriálové úloze budete simulovat růst na čtvercové mřížce, tím se kód podstatně zjednoduší.

```
# DIFFUSION LIMITED AGGREGATION
# nacteme grafickou knihovnu a numerickou knihovnu
import matplotlib as mpl
from matplotlib import pyplot as plt
import numpy as np

# definujeme rozmer mrizky a maximalni pocet castic
roz = 20
mass_max = 100
# pole pro spirální 1D index, bezpecne vetsi nez je potreba
roz_id = 6*roz**2
# pole, do kterzch budeme ukladat prepoctena kartezske souradnice
# definujeme vektory pomahajici pri prepoctu z heza na ctvercovou mrizku
xs = np.zeros(mass_max, dtype=float)
ys = np.zeros(mass_max, dtype=float)
xhelp = [np.sqrt(3.)/2., 0., -np.sqrt(3.)/2., -np.sqrt(3.)/2., 0., np.sqrt(3.)/2.]
yhelp = [-0.5, -1., -0.5, 0.5, 1., 0.5]
# pole nul, do ktereho budeme ukladat stavy bunek
pole = np.zeros(roz_id, dtype=np.int)
# inicializujeme bunky, hmotnost a nejvetsi vzdalenost od stredu
pole[0] = 1
mass = 1
rmax = 0

while mass < mass_max:
    # pocatecni vzdalenost castice
    r = rmax+1
    # pricist k hranici v radialnim smeru, za kterou castice nesmi utect
    rkill = rmax
    # vyber nahodny prvek z sestiuhelniku o rozmeru r
    i = int(6*r*np.random.random() + 1)
    nn = 0
    # pohyb difundujici castice
    while nn == 0:
        i_full = i+3*r*(r-1) # pozice v poli 'pole'
        # nejsme-li ve vrcholu sestiuhelniku
```

```

if (i % r) != 0:
    # nalezneme sousedy
    if i == 1:
        ileft = 6*r
        idownleft = 3*r*(r-1)
    else:
        ileft = i-1
        idownleft = (i*(r-1))/r + 3*(r-1)*(r-2)
    # dvojitě lomítka vrací floor
    pole_nn = [i+1+3*r*(r-1), ileft+3*r*(r-1), idownleft,
               (i*(r-1))/r+3*(r-1)*(r-2)+1,
               (i*(r+1))/r+3*r*(r+1),
               (i*(r+1))/r+3*r*(r+1)+1]
    obsazeno = np.where(pole[pole_nn] == 1)
    nn = len(obsazeno[0])
    # je-li nějaký soused obsazen, přilep se
    if nn > 0:
        pole[i_full] = 1
        rmax = max([rmax, r])
        mass += 1
    # jinak difunduj dál
    else:
        step = int(np.random.random()*6)
        # radialní sestup
        if (step == 2 or step == 3):
            r -= 1
        # radialní vzestup
        elif (step == 4 or step == 5):
            r += 1
        i = pole_nn[step] - 3*r*(r-1)
# pokud jsme ve vrcholu sestihelníku
else:
    if i == 6*r:
        irect = 1
        iupright = 1
    else:
        irect = i+1
        iupright = i*(r+1)/r+1
    if i == 1:
        ileft = 6
    else:
        ileft = i-1
    pole_nn = [irect+3*r*(r-1), ileft+3*r*(r-1),
               i*(r-1)/r+3*(r-1)*(r-2), i*(r+1)/r+3*r*(r+1),
               iupright+3*r*(r+1), i*(r+1)/r-1+3*r*(r+1)]
    obsazeno = np.where(pole[pole_nn] == 1)
    nn = len(obsazeno[0])
    if nn > 0:
        pole[i_full] = 1
        rmax = max([rmax, r])
        mass += 1
    else:
        step = int(np.random.random()*6)
        if step == 2:
            r -= 1
        elif ((step == 3 or step == 4) or step == 5):
            r += 1
        i = pole_nn[step] - 3*r*(r-1)
# pokud částice utekla, začneme znovu
if r > (rmax+rkill):
    r = rmax+1

```

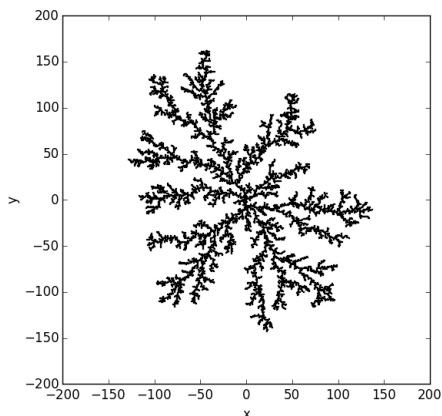


```

        i = int(6*r*np.random.random()+1)
        continue
    x0 = r*np.sin(2.*np.pi/(6.*r)*(i - (i % r)))
    y0 = r*np.cos(2.*np.pi/(6.*r)*(i - (i % r)))
    x = x0 + (i % r)*xhhelp[min(int(i//r),5)]
    y = y0 + (i % r)*yhhelp[min(int(i//r),5)]
    xs[mass-1] = x
    ys[mass-1] = y

# vykreslime castice pomoci sestihelnicovych symbolu
dlaplot = plt.scatter(xs,ys,c='k',s=30000/(roz**2),marker=(6,0,30))
plt.xlim(-roz, roz)
plt.ylim(-roz, roz)
plt.axes().set_aspect('equal')
plt.xlabel('x')
plt.ylabel('y')
plt.show()

```



Obr. 51: Výsledek simulace růstu podle DLA modelu na hexagonální mřížce. Počet částic v krystalu je 8000. Všimněte si, že i přes poměrně benevolentní limit na maximální vzdálenost difundující částice je na krystalu vidět preferovaný směr růstu.

Na vzniklý krystal (shluk částic) můžeme pohlížet jako na fraktál. Nebudeme zde zabíhat do hluboké teorie, fraktál si prostě představíme jako objekt, který při bližším pohledu (myšleno ve větším měřítku) vykazuje stále stejnou strukturu – je soběpodobný. Příkladem z přírody je například list kapradiny, který se skládá z mnoha menších listů, a ty zase z menších lístků. V matematické abstrakci lze na objekt „zoomovat“ donekonečna, vždy budeme nacházet stále stejnou geometrickou strukturu. Podobně na obrázku 51 najdeme na každé větvi rostoucí z počátku další větve, ze které rostou další větve atd.

Pro fraktály je typické, že jejich dimenze je větší než u „obyčejných“ geometrických objektů. Například kružnice má dimenzi 1, ale Kochova vločka (obr. 52), což je vlastně jenom jinak zakřivená čára, má dimenzi přibližně 1,26. Na Kochově vločce si vysvětlíme definici dimenze.<sup>72</sup> Necht' má strana počátečního trojúhelníku délku  $\varepsilon = 1$ . Počet úseček v jedné straně je samozřejmě  $N = 1$ . V dalším kroku již jedna strana obsahuje  $N = 4$  úsečky délky  $\varepsilon = 1/3$ , v dalším kroku  $N = 16$  úseček délky  $\varepsilon = 1/9$  atd. Fraktální dimenze je pak definována jako

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}}, \quad (76)$$

přičemž pro vločku můžeme psát  $\varepsilon(n) = 1/3^n$ ,  $N(n) = 4^n$ , a tedy

$$D_{\text{Koch}} = \lim_{n \rightarrow \infty} \frac{\log 4^n}{\log 3^n} = \frac{\log 4}{\log 3} \doteq 1,26.$$

Všimněte si, že se ve výpočtu mocniny  $n$  zkrátily. To je dáno faktem, že princip výstavby Kochovy vločky spočívá v aplikaci zcela identické geometrické deformace na jednotlivé hrany vločky v každém kroku. Limitní proces tedy nemá v případě matematicky exaktní vločky smysl (tj. dimenzi by šlo definovat jinak, např. jako  $\log N(\varepsilon)/\log(1/\varepsilon)$  bez limity), ale má smysl v případě fraktálu vytvořeného DLA. Při „zoomování“ zde totiž nepozorujeme zcela identické obrazce, ale pouze velmi podobné, protože výstavba krystalu probíhá náhodně. Navíc krystal není nekonečně velký, resp. nemá nekonečně detailní strukturu, a také v důsledku nutného omezení délky náhodné procházky vstupují do procesu další odchylky. Při výpočtu dimenze budeme nyní místo strany trojúhelníku měřit lineární rozměr krystalu (radiální vzdálenost nejvzdálenější částice od počátku), místo počtu úseček máme počet částic. Přítomnost limity nám říká, že pro co nejpřesnější výsledek chceme nechat krystal růst co nejdéle. Pro krystal na obrázku 51 vychází dimenze  $D \doteq 1,75$ .

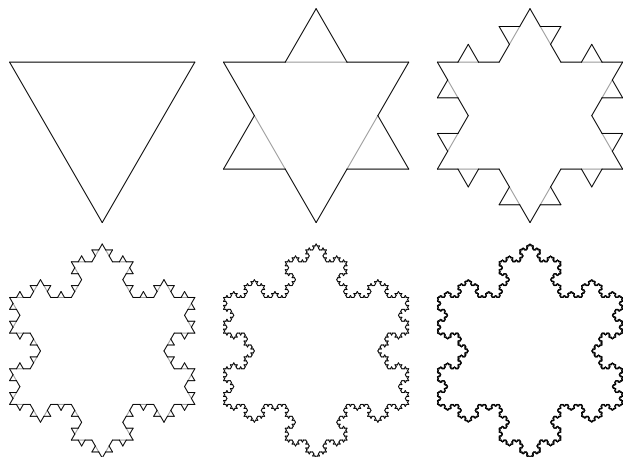
Nakonec ještě dodejme, že v případě materiálu získaného na základě Edenova modelu můžeme pohlížet na jeho povrch jako na fraktál, tělo takového materiálu ale není fraktální.

## Úloha V.S ... rostou nám diferenciální rovnice

10 bodů

- Řešte problém dvou těles pomocí Verletovy a Rungovy-Kuttovy metody 4. řádu přes několik (mnoho) period. Krok přitom volte tak velký, aby se projevily numerické chyby, a pozorujte, jakým způsobem se chyby v obou případech projevují na tvaru trajektorie.
- Řešte pohyb tlumeného lineárního harmonického oscilátoru, který je dán rovnicí  $\ddot{x} + 2\delta\omega\dot{x} + \omega^2x = 0$ , kde  $\omega$  je úhlová frekvence a  $\delta$  tlumící člen. Parametry měňte a sledujte změny v chování oscilátoru. Pro jaké hodnoty parametrů se oscilátor utlumí nejrychleji?
- Modelujte růst povrchu metodou balistické depozice a studujte statistické chování hrubosti povrchu. Nalezněte mocniny  $\alpha$  a  $\beta$  popisující růst před saturací

<sup>72</sup>Existuje mnoho způsobů, jak definovat fraktální dimenzi, zde jsme vybrali jeden z nich.



Obr. 52: Prvních šest iterací Kochovy vločky. Tenké čáry jsou vždy z předchozího kroku.

a po saturaci (viz seriál). Vyjděte z kódu v seriálu. Volte takový počet kroků, abyste byli schopni dobře studovat oba režimy hrubnutí. Lineární rozměr povrchu volte alespoň  $L = 256$ . (Upozornění: simulace mohou trvat i několik hodin.)

- d) Simulujte na čtvercové mřížce šíření zhoubného nádoru pomocí Edenova modelu. Uvažujte přitom následující obměnu: s pravděpodobností  $p_1$  dojde k nákaze zdravé buňky v kontaktu s nádorovou a s pravděpodobností  $p_2$  dojde k uzdravení nakažené. Volte nejprve  $p_1 \gg p_2$ , pak  $p_1 > p_2$  a nakonec  $p_1 < p_2$ . Na počátku nechtě je nakaženo pět buněk do tvaru kříže. Kvalitativně popište, co pozorujete.

- e) Přepište kód ze seriálu pro růst fraktálního krystalu (DLA model) na hexagonální mřížce na růst na čtvercové mřížce a spočtěte dimenzi výsledného fraktálu.

*Poznámka* Využití kódy přiložené k seriálu není nutné, ale doporučeno.

(řešení str. 263)

## Kapitola 6: Matice a populace

Třicátý první ročník FYKOSu se blíží ke konci a s ním i seriál o numerických metodách a počítačových simulacích. V tomto díle se rozpojeneme na úplný začátek seriálu, kdy jsme klasifikovali simulace, a povíme si něco o populačních modelech. V numerické části se krátce vrátíme k diferenciálním rovnicím, pohovoříme o oblastech stability a nakonec přehledově pojednáme o numerické lineární algebře.

### Oblasti stability

V řešení minulého dílu se objevilo tvrzení, že explicitní Eulerova metoda je zcela nevhodná pro řešení netlumeného harmonického oscilátoru a že to vyplývá z její oblasti stability. Pojdme si proto stručně objasnit, o co se jedná. Vezměme si jednoduchou diferenciální rovnici  $\dot{y} = \lambda y$ , kde  $\lambda$  je obecně komplexní číslo (a tedy i  $y$  je komplexní funkce reálné proměnné  $t$ ). Analytické řešení této rovnice je

$$y(t) = y_0 e^{\lambda t}.$$

Pro  $\operatorname{Re}(\lambda) < 0$  řešení s časem klesá k nule, pro  $\operatorname{Re}(\lambda) > 0$  naopak exponenciálně roste. Řešme tuto rovnici numericky, pro začátek explicitní Eulerovou metodou. Platí tedy

$$y_{n+1} = y_n + h\lambda y_n = y_n(1 + h\lambda) = y_0(1 + h\lambda)^{n+1}. \quad (77)$$

Rozumným požadavkem je, aby numerické řešení nebylo příliš vzdáleno od řešení skutečného, přesněji budeme požadovat, aby pro dané  $h\lambda$  bylo řešení omezené<sup>73</sup> (tj. neutíkalo do nekonečna) pro  $n \rightarrow +\infty$ . Tato požadovaná vlastnost se nazývá *absolutní stabilita*. Konkrétně pro explicitní Eulerovu metodu je splněna, pokud  $|1 + h\lambda| \leq 1$ , jak je vidět z rovnice (77). Body  $z = h\lambda$  v komplexní rovině, které splňují podmínku absolutní stability, tvoří tzv. *oblast (absolutní) stability*. Explicitní Eulerova metoda má tedy oblast stability tvořenou kruhem o středu  $-1$  a jednotkovém poloměru. Skutečně, pokud si vzpomenete na diskusi o řešení problémů se silným tlumením v minulém díle, zjistili jsme tehdy, že pro rovnici  $\dot{y} = -cy$  musíme volit  $h < 2/c$ . To přesně odpovídá podmínce na okraj oblasti stability na reálné ose, kde  $h\lambda = -hc \geq -2$ .

Spočítejme si ještě oblast stability pro implicitní Eulerovu metodu. Z jejího předpisu máme

$$y_{n+1} = y_n + h\lambda y_{n+1} = \frac{y_n}{1 - h\lambda} = \frac{y_0}{(1 - h\lambda)^{n+1}}.$$

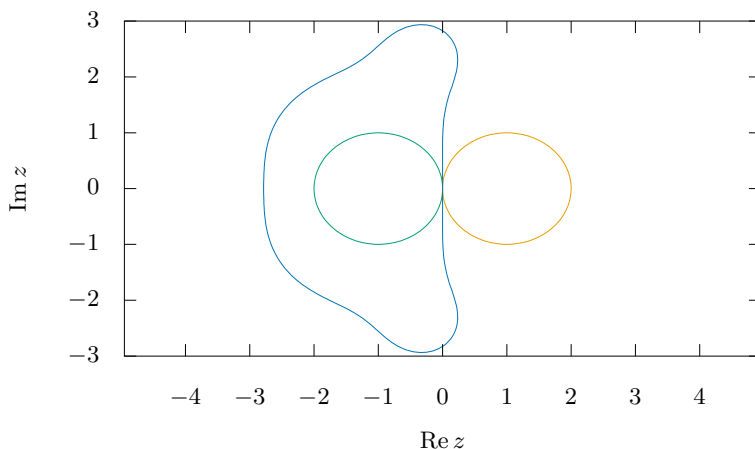
Aby  $|y_n|$  bylo konečné, musí platit  $|h\lambda - 1| \geq 1$ . Oblast stability implicitní Eulerovy metody je tedy celá komplexní rovina až na jednotkový kruh o středu 1. Oblasti stability pro metody vyšších řádů a pro více krokové metody se formálně získávají stejným způsobem, technicky jde ale o náročnější výpočty, proto zde ukážeme pouze výsledek ve formě grafu 53.

Zjistili jsme tedy, jaký musíme volit krok, abychom měli stabilní řešení problému  $\dot{y} = \lambda y$ . Co ale dělat, pokud máme obecnější problém? Zobecnění mohou být dvou typů. Zaprvé můžeme mít jednu nelineární rovnici, zadruhé můžeme mít soustavu lineárních rovnic. Nejobecnější je pak kombinace obojího. Pokud máme jednu (skalární) nelineární rovnici ve tvaru  $\dot{y} = f(y, t)$ , můžeme ji rozvést do prvního řádu<sup>74</sup>

$$f(y, t) \approx f(y_0, t_0) + \left. \frac{\partial f}{\partial y} \right|_{y=y_0} (y - y_0) + \left. \frac{\partial f}{\partial t} \right|_{t=t_0} (t - t_0).$$

<sup>73</sup>Není nyní příliš jasné, proč toto požadujeme i pro  $\operatorname{Re}(\lambda) > 0$ , kde ani skutečné řešení omezené není. Ukazuje se ale, že tento požadavek i zde nadělá více užítku než škody, protože zamezíme tomu, aby numerické řešení divergovalo ještě rychleji, než skutečné. Hlavní význam této konstrukce je ale nepochybně pro  $\operatorname{Re}(\lambda) \leq 0$ .

<sup>74</sup>Jde vlastně o Taylorův rozvoj funkce více proměnných.



Obr. 53: Hranice oblastí stability pro explicitní (vnitřek levého kruhu) a implicitní (vnějšek pravého kruhu) Eulerovu metodu a Rungovu-Kuttovu metodu 4. řádu (vnitřek „šišoidu“).

Lze přitom ukázat, že na konstantním členu a na členu lineárním v  $t$  analýza absolutní stability nezávisí, můžeme tedy pro účely vyšetřování stability aproximovat  $\lambda \approx \left. \frac{\partial f}{\partial y} \right|_{y=y_0}$ .

Pokud máme lineární soustavu rovnic  $\dot{y} = Ay$ , kde  $A$  je matice a  $y$  a  $\dot{y}$  jsou vektory, je třeba nalézt tzv. vlastní čísla  $\lambda_i$  matice  $A$  a krok  $h$  zvolit tak, aby pro všechna tato vlastní čísla platilo, že  $h\lambda_i$  leží v oblasti stability. Proč je třeba použít zrovna vlastní čísla je možné nahlédnout například z toho, že pokud s maticí  $A$  provedeme operaci zvanou diagonalizace,<sup>75</sup> pak bude mít tato diagonalizovaná matice  $A'$  na své hlavní diagonále právě vlastní čísla a jinde budou nuly. Soustava diferenciálních rovnic tedy formálně přejde do tvaru  $\frac{d}{dt}\tilde{y}_i = \lambda_i\tilde{y}_i$ , což již připomíná původně vyšetřovanou rovnici. Značky  $\tilde{y}$  zde pouze značí, že jde pouze o formální převod a  $\tilde{y}$  neodpovídají původním  $y$ .

Tento případ si můžeme ilustrovat. Mějme diferenciální rovnici lineárního harmonického oscilátoru  $\ddot{y} + \omega^2 y = 0$ . Po převodu rovnice na soustavu rovnic prvního řádu máme  $\dot{\mathbf{y}} = A\mathbf{y}$ , kde

$$A = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix}.$$

<sup>75</sup>ne každá matice je diagonalizovatelná; zde se zaměříme pouze na ty, které jsou – např. všechny symetrické matice

Vlastní čísla této matice<sup>76</sup> jsou  $\lambda_{1,2} = \pm i\omega$ , jsou tedy ryze imaginární. Odtud už okamžitě vidíme, že řešení této soustavy explicitní Eulerovou metodou bude pro libovolný krok  $h \neq 0$  nestabilní, protože obě  $h\lambda_i$  pak neleží v oblasti stability. Naopak např. u Rungovy-Kuttovy metody 4. řádu můžeme volit libovolný krok  $h \leq h_{\max} \doteq 2,8/\omega$  a metoda bude stabilní.

Poslední, nejsložitější možností je pak kombinace předchozích dvou zobecnění, máme tedy nelineární soustavu rovnic. Jako v prvním případě, i nyní provedeme Taylorův rozvoj do prvního řádu

$$f_i(\mathbf{y}, t) \approx f_i(\mathbf{y}_0, t_0) + \sum_{j=1}^N \left. \frac{\partial f_i}{\partial y_j} \right|_{\mathbf{y}=\mathbf{y}_0} (y_j - y_{j,0}) + \left. \frac{\partial f_i}{\partial t} \right|_{t=t_0} (t - t_0).$$

Pokud opět škrtneme všechny členy kromě těch, které jsou lineární v  $y_j$ , dostaneme soustavu rovnic  $\dot{y}_i = J_{ij}y_j$ , kde  $J_{ij} = \left. \frac{\partial f_i}{\partial y_j} \right|_{\mathbf{y}=\mathbf{y}_0}$  jsou složky matice zvané jakobián. Nyní opět nalezneme vlastní čísla matice  $J_{ij}$  a aplikujeme stejnou podmínku, jako při zobecnění na soustavu lineárních rovnic.

Tímto bychom ukončili téma řešení obyčejných diferenciálních rovnic. Probrali jsme jej dostatečně podrobně na to, abyste nyní byli schopni numericky vyřešit takřka libovolnou soustavu obyčejných diferenciálních rovnic. Pokud byste se však chtěli dozvědět více, či znovu a jinak slyšet už jednou řečené, podívejte se na záznam přednášky *Počítačové simulace ve fyzice*, která byla přednesena v rámci letošního cyklu FYKOSích přednášek pro středoškolačky a najdete ji na našem Youtube kanálu.<sup>77</sup>

## Numerická algebra

Posledním tématem, který si v numerické části seriálu představíme, je numerická lineární algebra. Konkrétně se podíváme na to, jak řešit soustavy lineárních algebraických rovnic. Algebraické rovnice jsou takové ty klasické, které jste znali ze školy jen jako „rovnice“, než jsme vám představili rovnice diferenciální. Lineární jsou, pokud jdou zapsat ve tvaru

$$\sum_{i=1}^N a_i x_i = b,$$

kde  $a_i, b$  jsou konstanty a  $x_i$  jsou neznámé. Soustavu  $M$  takových lineárních rovnic pak můžeme zapsat ve vektorové formě  $\mathbf{Ax} = \mathbf{b}$ , neboli

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{pmatrix}.$$

<sup>76</sup>To mi v této chvíli musíte věřit.

<sup>77</sup><https://www.youtube.com/fykosak>

Připomeňme, že maticové násobení funguje s pravidlem „řádek  $\times$  sloupec“, tedy např. z prvního řádku matice máme rovnici

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N = b_1.$$

Matici  $A$  nazveme *maticí soustavy*, matici

$$(A|b) \stackrel{\text{def}}{=} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2N} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} & b_M \end{pmatrix}$$

pak nazveme *rozšířenou maticí soustavy*. K dalšímu výkladu nyní potřebujeme vysvětlit pojmy lineární nezávislost a řád matice. Řekneme, že množina vektorů  $\{\mathbf{v}_1, \dots, \mathbf{v}_N\}$  je *lineárně nezávislá*, pokud rovnici  $\sum_{i=1}^N c_i \mathbf{v}_i = \mathbf{0}$ , kde  $c_i$  jsou číselné konstanty, které můžeme zvolit libovolně, splníme pouze tak, že zvolíme všechna  $c_i = 0$ . V opačném případě jsou vektory *lineárně závislé*. Například vektory  $(1, 0, 2)$ ,  $(2, 1, 5)$  a  $(11, 3, 25)$  jsou lineárně závislé, neboť

$$5 \cdot (1, 0, 2) + 3 \cdot (2, 1, 5) - 1 \cdot (11, 3, 25) = (0, 0, 0).$$

V případě matice lze také na jednotlivé sloupce, či řádky pohlížet jako na vektory. *Řád matice* (označme jej  $\text{rank}$ ) je pak maximální počet lineárně nezávislých sloupců (nebo řádků, lze dokázat, že to je jedno) matice. Například matice

$$\begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 5 \\ 11 & 3 & 25 \end{pmatrix}$$

má řád 2, neboť všechny tři řádky jsou lineárně závislé. Pokud ale libovolný z nich škrtneme, bude mít výsledná matice už oba dva řádky lineárně nezávislé. Sami si pak můžete ověřit, že to platí i pro její sloupce.

Důvod, proč jsme se zabývali těmito pojmy je ten, že na jejich základě dokážeme určit počet řešení soustavy lineárních rovnic. Pokud je totiž řád rozšířené matice soustavy  $(A|b)$  větší než řád matice soustavy  $A$ , pak soustava nemá řešení. Naopak, <sup>78</sup> pokud  $\text{rank}(A|b) = \text{rank}(A)$ , mohou nastat dva případy. Pokud zároveň počet neznámých  $N = \text{rank}(A)$ , pak má soustava právě jedno řešení. Tímto případem se zde budeme zabývat. Nakonec, <sup>79</sup> pokud  $N > \text{rank}(A) = \text{rank}(A|b)$ , soustava má nekonečně mnoho řešení.<sup>80</sup> Tyto případy se zpravidla neřeší numericky, proto se jimi nebudeme zabývat.

Možná vás napadá, jestli nejdeme s kanónem na vrabce. Vždyť soustavu lineárních algebraických rovnic zvládneme vyřešit ručně, například Gaussovou eliminací a nepotřebujeme speciální numerické metody. Problém ale nastane, pokud naše

<sup>78</sup>Případ  $\text{rank}(A|b) < \text{rank}(A)$  nastat nikdy nemůže.

<sup>79</sup>Případ  $N < \text{rank}(A)$  totiž také nemůže nastat.

<sup>80</sup>Libovolná  $N$ -tice čísel ale nemusí být řešením, neboť řešení stále musí splňovat onu soustavu rovnic.

soustava rovnic bude velká, metodami, které si představíme lze řešit soustavy až do  $N \sim 100\,000$ , větší soustavy se nám jednoduše už nevejdou do paměti počítače. Za určitých okolností (matice soustavy je řídká tj. má skoro všechny prvky nulové) lze speciálními metodami řešit i větší soustavy. Jistě uznáte, že takové soustavy bychom už ručně řešili těžko. Takto velké soustavy se v praxi často vyskytují, například při řešení parciálních diferenciálních rovnic někdy nastane krok, kdy získáme obří soustavu lineárních algebraických rovnic, kterou potřebujeme vyřešit. Dalším příkladem může být složitý elektrický obvod řešený pomocí Kirchhoffových zákonů.

Samotné numerické řešení zpravidla probíhá tak, že si matici soustavy rozložíme na součin matic se speciálními vlastnostmi, které nám pak pomohou s maticovou rovnicí lépe pracovat. Správné technické provedení rozkladů je zpravidla složité na vysvětlení, takřka vždy je ale navíc náročné a zdlouhavé na implementaci. V praxi proto (minimálně pro rozklady, spíše však pro celou úlohu) použijte některou z numerických knihoven. My si zde představíme algoritmus pouze pro tzv. LU rozklad, zbytek textu se pak bude zabývat tím, jak nám tyto rozklady pomohou.

### LU dekompozice

Nejjednodušším rozkladem, o kterém si zde řekneme, je rozklad matice  $A$  na součin horní trojúhelníkové matice  $U$  (z angl. upper) a dolní trojúhelníkové matice<sup>81</sup>  $L$  (z angl. lower). Horní, resp. dolní trojúhelníková matice je taková, která má nad, resp. pod hlavní diagonálou všechny prvky nulové.

Algoritmus pro LU dekompozici, který si zde představíme, se nazývá Doolittlův a jde v podstatě o Gaussovu eliminaci. Pokud matici  $A$  zleva vynásobíme maticí

$$L_1 = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -l_{n1} & 0 & 0 & \cdots & 1 \end{pmatrix},$$

kde  $l_{k1} = a_{k1}/a_{11}$ , dostaneme matici

$$A^{(1)} = L_1 A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix},$$

---

<sup>81</sup>V tomto textu nebude značena  $L$ , ale  $L^{-1}$ .



zbavili jsme se tedy v prvním sloupci hodnot pod hlavní diagonálou. Pokud dále zkonstruujeme matice

$$L_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -l_{k+1,k} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -l_{n,k} & \cdots & 1 \end{pmatrix},$$

kde  $l_{jk} = a_{jk}^{(k-1)} / a_{kk}^{(k-1)}$  pro  $j > k$ , každou maticí  $L_k$  „vyřešíme“ jeden sloupec. Platí tedy ( $L_n = I$  vynecháme)

$$L_{n-1} \dots L_2 L_1 A = LA = U,$$

kde jsme součin matic  $L_k$  označili  $L$ . Pokud nyní obě strany vynásobíme inverzní maticí<sup>82</sup>  $L^{-1}$ , dostaneme

$$L^{-1}LA = IA = A = L^{-1}U,$$

k dokončení LU rozkladu tedy potřebujeme nalézt dolní trojúhelníkovou<sup>83</sup> matici  $L^{-1}$ . Inverze matice je obecně výpočetně náročná operace<sup>84</sup>, zde ale máme štěstí, neboť lze ukázat, že matice

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & 1 & \cdots & 0 \\ -l_{31} & -l_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -l_{n1} & -l_{n2} & \cdots & 1 \end{pmatrix}$$

a její inverze pak je

$$L^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ l_{31} & l_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix}.$$

Tím máme LU dekompozici zdánlivě zdárně provedenu. Vraťme se nyní k výpočtu  $l_{jk}$ . V něm dělíme hodnotami  $a_{kk}^{(k-1)}$ , které nesmí být nulové, ani kdybychom výpočet prováděli přesně. Při numerickém výpočtu navíc z důvodu redukce zaokrouhlovacích chyb požadujeme, aby tyto hodnoty byly srovnatelné s největšími

<sup>82</sup>Inverzní matice  $C^{-1}$  k matici  $C$  je taková, pro kterou platí  $C^{-1}C = CC^{-1} = I$ , kde  $I$  je jednotková matice.

<sup>83</sup>Což ale zatím nevíme, jestli  $L^{-1}$  bude splňovat.

<sup>84</sup>Ostatně jedna z cest, jak inverzi numericky efektivně provést, je založena právě na LU dekompozici.

hodnotami v matici. Řešením je před výpočtem  $L_k$  přeházet prvky  $A^{(k-1)}$  tak, aby se na  $k$ -tém prvku hlavní diagonály vyskytovalo co možná největší číslo. Většinou postačí, pokud prohazujeme řádky, takže hledáme maximum pouze v rámci  $k$ -tého sloupce. Pak mluvíme o *částečné pivotizaci*. U opravdu nehezkyých matic toto nemusí stačit, pak musíme použít *úplnou pivotizaci*, prohazovat řádky i sloupce a hledat tak maximum v rámci celé matice. Tato prohození si navíc rozkládáme. To je samo o sobě technicky náročnější na implementaci, proto opět doporučujeme použití numerických knihoven. Ovšem i při jejich použití je třeba dát pozor na to, co vrací, neboť většinou provedou rozklad  $A = PL^{-1}U$ , kde  $P$  je právě matice permutací řádků/sloupců vzniklá v důsledku pivotizace. Není tedy možné vzít slepě pouze matice  $L^{-1}$  a  $U$  vpadlé z metody!

Předpokládejme tedy, že máme úspěšně provedenu dekompozici  $A = PL^{-1}U$ , řešíme tedy maticovou rovnici  $PL^{-1}U\mathbf{x} = \mathbf{b}$ . Výpočet nyní rozložíme do dvou kroků. Nejprve vyřešíme rovnici  $PL^{-1}\mathbf{y} = \mathbf{b}$  pro neznámé  $\mathbf{y}$ . Protože inverze permutační matice je rovna její transpozici,<sup>85</sup> můžeme rovnici přepsat do tvaru  $L^{-1}\mathbf{y} = P^T\mathbf{b}$ . Druhým krokem je pak vyřešení rovnice  $U\mathbf{x} = \mathbf{y}$  pro neznámé  $\mathbf{x}$ . Protože  $L^{-1}$  i  $U$  jsou trojúhelníkové matice, lze maticové rovnice řešit dopřednou a zpětnou substitucí. Tedy například u rovnice  $U\mathbf{x} = \mathbf{y}$  má poslední řádek  $U$  pouze jednu nenulovou hodnotu  $u_{nn}$ , rovnou tedy určíme  $x_n = y_n/u_{nn}$ . V předposledním řádku jsou pouze maximálně dvě nenulové hodnoty, z nichž jedna se násobí s již známou hodnotou  $x_n$ , rovnou tedy vypočítáme hodnotu  $x_{n-1}$  a tak postupujeme dále.

První rovnici lze také vyřešit úpravou do tvaru  $\mathbf{y} = LP^T\mathbf{b}$ , musíme si ale dát pozor, abychom násobili v pořadí  $L(P^T\mathbf{b})$ , a ne  $(LP^T)\mathbf{b}$ . V prvním případě totiž z prvního násobení vznikne vektor, provádíme tedy dvě násobení matice a vektoru po sobě, což má složitost  $O(n^2)$ , zatímco násobení matic ve druhém případě má složitost<sup>86</sup>  $O(n^3)$ . Možná vás napadne, proč rovnou rovnici neupravíme do tvaru  $\mathbf{x} = A^{-1}\mathbf{b}$  a neprovedeme maticové násobení. Důvod je, jak jsme již poznamenali, že inverze matice je obecně velmi časově náročná.

### Ostatní metody a aplikace

Představili jsme si LU dekompozici, což je nejjednodušší z používaných rozkladů matic. Ostatní používané rozklady, jako např. QR či SVD rozklad, jsou složitější a vyžadují znalosti pokročilých pojmů lineární algebry, proto se jimi zde zabývat nebudeme. Tyto rozklady mají mnoho aplikací, kromě již zmíněného řešení soustav lineárních rovnic jde např. o hledání inverze matice, či jejího determinantu. Jak již bylo řečeno, tyto metody jsou (z paměťových důvodů) aplikovatelné pouze na poměrně malé<sup>87</sup> matice, často je ale potřeba řešit daleko větší soustavy rovnic. Pokud máme štěstí, a tato matice je řídká, tj. má skoro všechny prvky nulové, je možné si zapamatovat pouze kde se nachází tyto nenulové prvky a jaké jsou jejich hodnoty, čímž docílíme značné paměťové úspory. S takto reprezentovanou maticí

<sup>85</sup> Prohození řádků za sloupce a naopak.

<sup>86</sup> Existují pokročilé algoritmy, které to zvládnou o chlup rychleji.

<sup>87</sup>  $n = 10\,000$  je malá matice, jak se přesvědčíte v úloze.

ale zpravidla nedokážeme efektivně provádět libovolnou operaci, speciálně nechceme provádět rozklad na součin matic, neboť tyto matice by již nemusely být řídké. V takovém případě pak používáme různé iterační metody, tedy vyjdeme z nějakého odhadu  $\mathbf{x}^{(0)}$ , na který opakovaně aplikujeme algoritmus metody a určujeme přesnější odhady  $\mathbf{x}^{(k)}$  dokud nedokonvergujeme s dostatečnou přesností k výsledku. Tyto metody pro řešení řídkých soustav opět najdete v numerických knihovnách.

Zajímavou aplikací je taktéž fitování (experimentálních) dat polynomiální závislostí. Pokud totiž máme  $m$  naměřených dat  $(x_i, y_i)$  a teoretickou závislost  $y(x) = b_0 + \sum_{j=1}^n b_j x^j$ , máme soustavu rovnic

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{pmatrix},$$

neboli  $\mathbf{y} = \mathbf{X}\mathbf{b} + \boldsymbol{\varepsilon}$ , kde  $\mathbf{b}$  jsou hledané regresní koeficienty a  $\boldsymbol{\varepsilon}$  jsou skutečné experimentální chyby. Protože naměřených hodnot je víc než hledaných koeficientů, řešíme přeúřčenou soustavu, která pro  $\boldsymbol{\varepsilon} = 0$  nemá řešení. Ostatně když provedeme regresi, závislost nám nikdy přesně neprotíná všechny experimentální body, ty jsou místo toho rozmístěny okolo regresní závislosti. Místo přesného řešení se tedy snažíme najít takové koeficienty, pro které budeme řešení v jistém smyslu nejbližší, snažíme se tedy minimalizovat velikost vektoru  $\|\boldsymbol{\varepsilon}\|$ . Takový problém lze<sup>88</sup> převést na (nepřeúřčenou) soustavu lineárních rovnic pro  $\mathbf{b}$ . Vidíme tedy, že na aplikace numerické lineární algebry narazíme téměř na každém kroku.

### Řešení Poissonovy rovnice

V již odkazované přednášce pro středoškoláky jsme vyřešili Laplaceovu rovnici ve 2D pomocí Excelu. Vaší úlohou bude totéž<sup>89</sup> provést v Pythonu. Pojdme si proto zrekapitulovat potřebnou teorii.

Jedna z Maxwellových rovnic má podobu  $\operatorname{div} \mathbf{E} = \rho/\varepsilon_0$ , kde  $\mathbf{E}$  je intenzita elektrického pole,  $\rho$  je hustota náboje a  $\varepsilon_0$  je permitivita vakua. Pokud do ní dosadíme definici elektrického potenciálu  $\mathbf{E} = -\operatorname{grad} \varphi$ , obdržíme Poissonovu rovnici

$$\Delta \varphi = -\frac{\rho}{\varepsilon_0}.$$

V případě, že je všude  $\rho = 0$ , dostaneme tzv. Laplaceovu rovnici

$$\Delta \varphi = 0.$$

Pokud tedy dostaneme zadanou hustotu náboje ve všech bodech oblasti a navíc tzv. okrajové podmínky, řešením této rovnice obdržíme potenciál  $\varphi$  ve všech bodech prostoru.

<sup>88</sup>Postup ale vyžaduje znalost pokročilé matematiky, nebudeme jej tedy zde rozebírat.

<sup>89</sup>Excel vnitřně používal nějakou z iteračních metod, zatímco my budeme soustavu rovnic řešit přesně.

Poissonova rovnice je ukázkou parciální diferenciální rovnice, neboť Laplaceův operátor (zde ve 2D)  $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  obsahuje parciální derivace. Potenciál  $\varphi(x, y)$  je totiž funkcí více proměnných, musíme tedy specifikovat, podle které zrovna derivujeme. Všechny ostatní v tu chvíli (pokud derivujeme parciálně) považujeme za konstantu.

Poissonova rovnice je PDR takového typu, ve které nevystupuje derivace podle času. Nehledáme tedy časový vývoj, ale stacionární řešení. Z tohoto důvodu také nespecifikujeme počáteční podmínku, jako u ODR, musíme ale specifikovat tzv. okrajové podmínky. Rovnici totiž řešíme na nějaké konečné oblasti, derivace ale vyžadují znalost hodnot v okolí daného bodu. Na hranici této oblasti tedy musíme předepsat dodatečné informace, kterým se říká okrajové podmínky.

Nejjednodušší okrajovou podmínkou je *Dirichletova podmínka*, která je ve tvaru  $\varphi(x, y) = 0$ , resp. obecněji  $\varphi(x, y) = C(x, y)$ , kde  $C$  je pevně daná (většinou konstantní) funkce. Dirichletova podmínka tedy říká, jaká je hodnota potenciálu v daném bodě hranice. Druhou používanou podmínkou je *von Neumannova podmínka*, která je ve tvaru  $\frac{\partial \varphi}{\partial x} = 0$ , kdy derivaci provádíme vždy ve směru kolmo k hranici. Tato podmínka říká, že v daném bodě se hodnota potenciálu v daném směru nemění (určuje vlastně hodnotu elektrického pole).

Platí přitom, že v každém bodě hranice stačí specifikovat jednu z těchto dvou podmínek, aby bylo řešení jednoznačné. Dirichletovu podmínku použijeme, pokud víme hodnotu potenciálu, například na elektrodách kondenzátoru. Von Neumannovu podmínku pak použijeme, pokud jsme na hranici, kde předpokládáme v daném směru homogenní řešení. Použijeme je tedy např. ve volném prostoru daleko od všech zdrojů. Okrajové podmínky přitom nemusí být specifikovány pouze na samém okraji oblasti, ale kdekoliv, kde tyto podmínky potřebujeme specifikovat (a v těchto bodech pak neřešíme samotnou PDR). Pokud tedy např. do deskového kondenzátoru vložíme další elektrodu, specifikujeme ji pomocí Dirichletovy podmínky v oblasti, kde se elektroda nachází.

Nyní si povíme, jak Poissonovu rovnici vyřešit numericky. Použijeme k tomu metodu konečných diferencí, kdy budeme hledat řešení na nějaké (v našem případě pravoúhlé a pravidelné) mřížce. Laplaceův operátor obsahuje druhé derivace ve směrech  $x$  a  $y$ , pokud tedy najdeme numerickou aproximaci druhé derivace ve směrech  $x$  a  $y$ , najdeme i numerickou aproximaci Laplaceova operátoru. Obdobným postupem, jako v kapitole o numerické derivaci si napíšeme dva Taylory a sečteme je. Dostaneme

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f^{(3)}(x) + O(h^4), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f^{(3)}(x) + O(h^4), \\ f(x+h) + f(x-h) &= 2f(x) + h^2f''(x) + O(h^4). \end{aligned}$$

Druhou derivaci tedy můžeme aproximovat vztahem

$$f''_i = \frac{f_{i+1} + f_{i-1} - 2f_i}{h^2} + O(h^2),$$

kde  $h$  je vzdálenost dvou sousedních bodů mřížky. Za předpokladu, že krok  $h$  je stejný ve směru  $x$  i  $y$ , tedy můžeme pro Laplaceův operátor v bodě  $(i, j)$  mřížky, kde má funkce  $f(x, y)$  hodnotu  $f_{i,j}$ , psát

$$\Delta f_{i,j} = \frac{f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j}}{h^2} + O(h^2).$$

Tímto problém převedeme na řešení soustavy rovnic  $A\boldsymbol{\phi} = \mathbf{b}$ , kde matice  $A$  na jednotlivých řádcích (odpovídajících rovnicím pro jednotlivé body mřížky) obsahuje buď aproximaci Laplaceova operátoru výše, nebo některou z okrajových podmínek. Vektor  $\mathbf{b}$  pak obsahuje hodnoty  $-\rho/\varepsilon_0$  v jednotlivých bodech mřížky, případně pravé strany okrajových podmínek. Řešením této soustavy pak získáme hodnoty potenciálu  $\boldsymbol{\phi}$  ve všech bodech mřížky.

### Populační modely

Prakticky ve všech úlohách, které jsme v předešlých dílech seriálu řešili, jsme se věnovali pouze diskretním simulacím. Nyní využijeme nabyté znalosti z minula a podíváme se na spojité simulace, tj. takové, které vyžadují numerické řešení diferenciálních rovnic.

Diferenciální rovnice mají nepřeberné množství aplikací. Abychom si trochu odpočinuli od fyziky a podporovali mezioborové vzdělávání, zaměříme se na (v biologii hojně používané) modely vývoje populace.<sup>90</sup>

#### Základní populační modely

Není potřeba dlouze zmiňovat, že časový vývoj velikosti populace je ovlivněn širokou paletou různých faktorů a není tedy možné jeho dynamiku exaktně postihnout. O to se ani nebudeme pokoušet. Místo toho se zaměříme jen na pár důležitých vlivů a budeme sledovat, co se děje s populací, když má navrch ten či onen faktor.

Nejjednodušší je *exponenciální růstový model*. Ten je založen na předpokladu, že produkce potomků není nijak regulována, tj. prostředí poskytuje neomezený přísun potravy, prostoru a nežijí v něm predátoři. Potom můžeme vývoj počtu jedinců  $n(t)$  zapsat pomocí diferenciální rovnice

$$\frac{dn}{dt} = rn. \quad (78)$$

Tato rovnice má jednoduché řešení

$$n(t) = n(t_0)e^{r(t-t_0)}. \quad (79)$$

Parametr  $r$  se nazývá *růstový koeficient*,  $t_0$  je počáteční čas. Často budeme pokládat  $t_0 = 0$  a psát  $n_0 \equiv n(t_0 = 0)$ .

Pro  $r > 0$  populace roste, pro  $r < 0$  asymptoticky klesá k nule, pro  $r = 0$  vývoj stagnuje (populace je konstantní). Pro praktické účely je potřeba buďto koeficient

<sup>90</sup> V principu nám nic nebrání použít diskretní simulace (diferenční rovnice), ale jelikož se děti/mládata nerodí skokově, ale neustále a nepravidelně, dává lepší smysl použít diferenciální rovnice.

extrapolovat ze starších dat, nebo mít teorii pro jeho odhad<sup>91</sup>. S exponenciálním růstem či poklesem se samozřejmě setkáváme i ve fyzice, příkladem budíž ubývání částic radioaktivního nuklidu v důsledku rozpadu na lehčí částice – viz minulý díl seriálu, první kapitola.

Za jeden z prvních příkladů aplikace exponenciálního modelu v biologii můžeme považovat Fibonacciho úlohu s králíky. Fibonacci uvažoval model, v němž máme na počátku dva mladé králíky, přičemž mladý králík se stane dospělým za jeden měsíc (jednotka času, není podstatná pro teorii), a dospělý pár má každý měsíc dva nové potomky (pro štouraly: necht je v každém páru jeden samec a jedna samice). Počet všech párů králíků v jednotlivých časových krocích je v takovém případě dán sekvencí 1, 1, 2, 3, 5, 8, 13, ... Posloupnost, jejíž každý člen je dán součtem dvou předchozích, je na počest autora nazývána Fibonacciho. Pro nás je podstatné, že tento diskrétní model se chová v limitě velkých časů podle vztahu  $n(t) = \varphi^t / \sqrt{5}$ , kde  $\varphi = (1 + \sqrt{5})/2$  je zlatý řez, což volbou  $r = \ln \varphi$  a  $n_0 = 1/\sqrt{5}$  převedeme na rovnici (79).

Druhým významným populačním modelem, který je zároveň stále jednoduchý, je *logistický růst*. Je vyjádřen diferenciální rovnicí

$$\frac{dn}{dt} = rn \left(1 - \frac{n}{k}\right). \quad (80)$$

Koeficient  $k$  zde představuje tzv. *nosnou kapacitu prostředí*. Pokud bude kapacita nekonečná, hodnota závorky bude 1 a dostaneme zpět neregulovaný, exponenciální růst. Na hodnotu v závorce lze pohlížet jako na faktor, který reguluje velikost růstového koeficientu  $r$ . Pokud je velikost populace významně menší než je kapacita prostředí,  $n \ll k$ , je hodnota v závorce zhruba rovna 1 a dostáváme tak zmíněný neregulovaný růst. Jak se poměr  $n/k$  blíží zdola jedné, růst postupně zpomaluje, až se velikost populace zastaví na hodnotě  $n = k$ ; prostředí je saturováno.

S tímto chování se opět setkáváme i ve fyzice, a to u fermionového plynu. Zatímco ideální plyn složený z klasických částic lze komprimovat do libovolně malého objemu, u fermionů (např. elektrony) tomu tak není. Pauliho vylučovací princip totiž říká, že žádný kvantový stav nemůže být okupován dvěma či více fermiony s identickou sadou kvantových čísel. Proto může dojít k nasycení kvantových stavů, což má za následek například zastavení kolapsu hvězdy po dohoření paliva a následný vznik bílého trpaslíka.  
92

Jelikož je tento model mírně komplikovanější než předchozí, rozepíšeme zde postup řešení a výsledek rozebereme. Rovnici (80) separujeme a budeme integrovat,

$$\int_{n_0}^n \frac{dn}{n \left(1 - \frac{n}{k}\right)} = \int_0^t r dt.$$

---

<sup>91</sup>Takový teoretický vztah může obsahovat parametry jako počet potomků v jednom vrhu, dobu mezi vrhy, délku života atp.

<sup>92</sup>Jedná se ovšem o aproximaci. Tlak elektronového plynu může být překonán, což vede ke vzniku neutronové hvězdy nebo dokonce černé díry.

Na pravé straně dostáváme triviálně  $rt$ . Na levé straně použijeme substituci  $n = n_0/k$  a provedeme rozklad na parciální zlomky

$$\frac{1}{n'(1-n')} = \frac{1}{n'} - \frac{1}{n'-1}.$$

Tyto zlomky se integrují na logaritmy, a tak vrácením substituce a převedením rozdílu logaritmů na logaritmus podílu postupně dostaneme

$$\ln\left(\frac{n'}{n'_0}\right) - \ln\left(\frac{1-n'}{1-n'_0}\right) = \ln\left(\frac{n}{n_0}\right) - \ln\left(\frac{1-\frac{n}{k}}{1-\frac{n_0}{k}}\right) = \ln\left(\frac{n}{n_0} \frac{1-\frac{n_0}{k}}{1-\frac{n}{k}}\right).$$

Výsledek

$$\ln\left(\frac{n}{n_0} \frac{1-\frac{n_0}{k}}{1-\frac{n}{k}}\right) = rt$$

převědeme do exponenciálního tvaru

$$\frac{n}{1-\frac{n}{k}} = ce^{rt},$$

kde jsme použili substituci  $c = n_0/(1 - n_0/k)$ . Jelikož se jedná o rovnici lineární v  $n$ , můžeme snadno vyjádřit

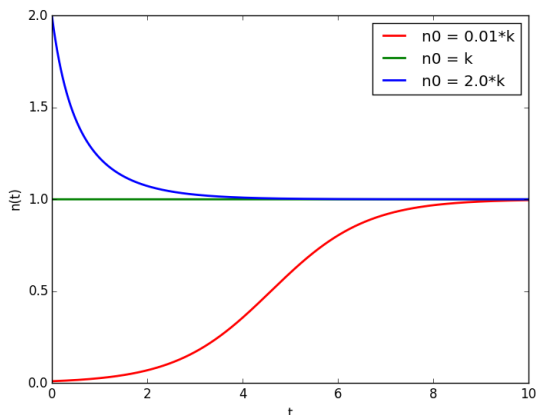
$$n(t) = \frac{k}{1 + \frac{k}{c}e^{-rt}}. \quad (81)$$

Funkce (81) se nazývá *logistická funkce*. Na základě hodnoty parametru  $c$  rozlišujeme tři kvalitativně odlišná řešení. Pro  $n_0 = k$  je  $c = \infty$  a funkce se tak redukuje na  $n = k$ , jak jsme si všimli výše. Pro  $n_0 < k$  je  $c > 0$ , z čehož plyne  $n(t) < k \forall t$ , přičemž díky členu  $e^{-rt}$  máme limitu  $\lim_{t \rightarrow \infty} n(t) = k$ . Třetím řešením je zavedení přesycené populace  $n_0 > k$ . Potom máme  $c < 0$ , což má za důsledek  $n(t) > k \forall t$  a limita nekonečného času je opět  $k$ . Populace, která je větší než nosná kapacita prostředí, postupně konverguje k této kapacitě. Všechna tři řešení naleznete na obrázku 54. Ještě uvedme, že model lze dále vylepšit například zavedením časové závislosti nosné kapacity či přičtením flukтуаčního členu, řešení se pak ovšem může podstatně zkomplikovat.

### Lotkûv–Volterrûv model

Oba výše zmíněné modely předpokládaly, že se ve zkoumaném prostředí vyskytuje pouze jedna populace. Takový případ můžeme zřídka pozorovat např. u některých autotrofû–extremofilû, typicky ovšem pozorujeme dvě a více populací, které spolu interagují. V následujících odstavcích zmíníme dva typy interakce a rozebereme modely, kterými je můžeme popsat.

Prvním typem je interakce predátor–kořist. Kořistí je nejčastěji býložravec, například již zmíněný králík, který získává potravu (trávu) z prostředí. Budeme uvažovat, že trávy je neomezeně mnoho, populace králíků se tedy sama o sobě bude vyvíjet exponenciálně. Predátor, například liška, se živí lovem králíků. Čím větší



Obr. 54: Logistický vývoj populace pro různé počáteční hodnoty. Zelená křivka představuje případ, kdy je populace již od začátku na nosné kapacitě, červená křivka odpovídá růstu z hodnot zanedbatelných vůči kapacitě a modrá křivka je vývoj populace nad nosnou kapacitou prostředí. Pro jednoduchost jsme volili  $k = 1$ .

je populace králíků, tím rychleji roste populace lišek. Naopak čím větší je populace lišek, tím pomaleji roste (rychleji klesá) populace králíků.

Již z uvedeného popisu si můžeme rozmyslet, že populace budou nejspíše oscilovat, přičemž zde bude určitý fázový posun, neboť když je jedna populace na maximu, nemůže být v maximu i druhá. Matematický model vývoje populací se nazývá *Lotkuv–Volterrov model predátor–kořist* a je shrnut soustavou diferenciálních rovnic

$$\begin{aligned}\frac{dx}{dt} &= r_x x - D_x xy, \\ \frac{dy}{dt} &= r_y xy - D_y y.\end{aligned}\quad (82)$$

Koeficienty  $r_x$ ,  $r_y$  jsou růstové, koeficienty  $D_x$ ,  $D_y$  jsou extinkční.<sup>93</sup> Všechny tyto koeficienty jsou kladné.

Rovnice dobře postihují fakt, že zatímco predátora  $y$  interakce (člen  $xy$ ) živí, kořist zase hubí. Špatnou zprávou je, že tyto rovnice nám neumožňují najít časovou závislost velikosti jednotlivých populací v uzavřeném tvaru. Soustavu nelineárních diferenciálních rovnic (82) je možné linearizovat, dostaneme tak ale pouze aproximační řešení pro malé relativní změny v populacích, což není uspokojivé. Exaktně

<sup>93</sup>Koeficienty v Lotkově–Volterrově modelu se často uvádějí bez konkrétních názvů, vybrali jsme jim proto nějaké příhodné.



dokážeme nalézt například stacionární bod, tj. velikosti populací predátora a kořisti, při kterých dojde ke stagnaci. Stačí prostě položit  $\dot{x} = 0$ ,  $\dot{y} = 0$  a vyřešit soustavu dvou lineárních algebraických rovnic. Výsledkem jsou kořeny  $(x_1, y_1) = (0, 0)$  a  $(x_2, y_2) = (D_y/r_y, r_x/D_x)$ . Lze ukázat, že bod  $(x_1, y_1)$  je sedlový, a tedy nestabilní, a tedy že populace nemají tendenci vymřít, což ukazuje na kvalitu modelu. Perturbace okolo  $(x_2, y_2)$  vedou k oscilacím ve vývoji populace (zmíněné linearizované řešení). Vyšetřování stability řešení diferenciálních rovnic je velice zajímavá disciplína, ale nemáme zde na ni ani prostor, ani matematický aparát.

Je tu ale i dobrá zpráva, a sice, že pokud nám bude stačit sledovat vývoj pouze během několika mála period, vystačíme si při integraci zcela bezpečně s oblíbenou explicitní metodou RK4. Níže je přiložen kód, který řeší Lotkovy-Volterrovy rovnice. Získaný vývoj je vykreslen na obrázku (55). V obrázku si všimněte stavu, kdy se populace kořisti přiblíží k nulové hodnotě. Při aplikaci modelu v biologii musíme mít na paměti, že aby se kořist mohla rozmnožovat, musí přežít alespoň jeden jedinec. Pokud bychom hodnotu  $x$  považovali za počet jedinců, je z obrázku 55 zřejmé, že pro zvolené parametry model selhal. Tento jev se v literatuře označuje jako „atto-fox problem“ (při modelování vývoje populací králíků a lišek v Británii klesla populace predátorů na jednu attolišku, tj.  $10^{-18}$  lišky).

```
# LOTKUV-VOLTERRUV MODEL PREDATOR KORIST
# Importujeme knihovny
import numpy as np
from matplotlib import pyplot as plt

# metoda Runge-Kutta 4. radu pro funkce bez explicitni casove zavislosti
def rk4(f, y, h):
    k1 = f(y)
    k2 = f(y + (h/2.)*k1)
    k3 = f(y + (h/2.)*k2)
    k4 = f(y + h*k3)
    y = y + (h/6.)*(k1 + 2.*k2 + 2.*k3 + k4)
    return y

# Lotkuv-Volterruv model, r rust, D extinkce,
# pruni prvek pole korist, druhy predator
def lv_pp(y):
    r = np.array([0.5,0.1])
    D = np.array([0.1,0.3])
    return np.array([r[0]*y[0] - D[0]*y[0]*y[1], r[1]*y[0]*y[1] - D[1]*y[1]])

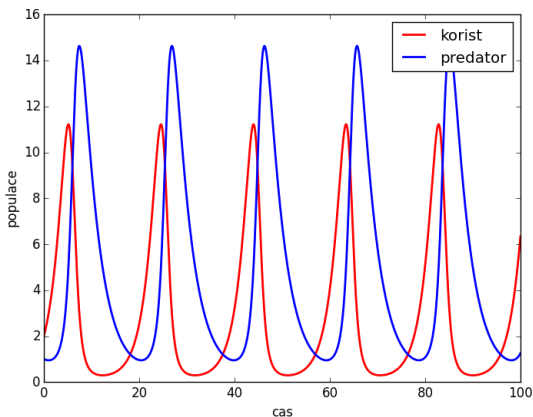
# definujeme delku casoveho kroku, pocet kroku, pocatecni velikost populaci
h = 0.01
kroky_max = 9999
krok = 0
cas_pole = np.zeros(kroky_max+1)
korist_pole = np.zeros(kroky_max+1)
predator_pole = np.zeros(kroky_max+1)
y = np.array([2.,1.])
korist_pole[krok] = y[0]
predator_pole[krok] = y[1]
# integrujeme pomoci rk4, ukladame do pole
while krok < kroky_max:
    y = rk4(lv_pp,y,h)
    krok += 1
    korist_pole[krok] = y[0]
```

```

    predator_pole[krok] = y[1]
    cas_pole[krok] = krok*h

# vykreslim casovou zavislost
koristplot, = plt.plot(cas_pole, korist_pole, 'r', linewidth=2, label='korist')
predatorplot, = plt.plot(cas_pole, predator_pole, 'b', linewidth=2, label='
    predator')
plt.xlabel('cas')
plt.ylabel('populace')
plt.legend(handles=[koristplot, predatorplot])
plt.show()

```



Obr. 55: Vývoj populací podle Lotkova–Volterrova modelu predátor–kořist.

Červená čára představuje kořist, modrá predátora. Volili jsme koeficienty  $r_x = 0,5$ ,  $r_y = 0,1$ ,  $D_x = 0,1$ ,  $D_y = 0,3$  a počáteční populace  $x = 2$ ,  $y = 1$ . Velikost populace považujeme za normalizovanou, tzn. nepředstavuje počet jedinců.

Další a poslední populační model, který v tomto textu rozebereme, je *kompetitivní Lotkův–Volterrův model*. Zatímco model predátor–kořist přidával interakci k exponenciálnímu modelu, kompetitivní model přidává interakci k logistickému modelu. Příslušná soustava diferenciálních rovnic má pro dvě populace tvar

$$\begin{aligned}
 \frac{dx}{dt} &= r_x x \left( 1 - \left( \frac{x + I_{xy}y}{k_x} \right) \right), \\
 \frac{dy}{dt} &= r_y y \left( 1 - \left( \frac{y + I_{yx}x}{k_y} \right) \right).
 \end{aligned} \tag{83}$$

V tomto modelu nedochází k predaci, populace však soupeří o společný zdroj potravy, což je vyjádřeno členy  $I_{xy}y$  a  $I_{yx}x$ , kde interakční koeficienty  $I$  jsou kladné. Na závorku se opět můžeme dívat jako na parametr upravující růstový koefici-

ent, interakční koeficienty pak můžeme interpretovat jako poměr, který nám říká, kolika jedincům v populaci  $x$  zabere životní prostor jedinec z populace  $y$ .

Podobně jako v případě modelu predátor–kořist neexistuje analytické řešení a musíme se spolehnout na numeriku. Co se týče stacionárních bodů, řešíme soustavu dvou kvadratických rovnic a nalezneme kořeny

$$\begin{aligned}x_1 &= 0, y_1 = 0, \\x_2 &= 0, y_2 = k_y, \\x_3 &= k_x, y_3 = 0, \\x_4 &= \frac{k_x - I_{xy}k_y}{1 - I_{xy}I_{yx}}, y_4 = \frac{k_y - I_{yx}k_x}{1 - I_{xy}I_{yx}}.\end{aligned}$$

Zde by byl další rozbor podstatně komplikovanější než u modelu s predací. Pouze si proto povšimněme, že čtvrtý kořen<sup>94</sup> může v případě silných interakcí nabývat záporných hodnot, tj. v kladné oblasti parametrů  $x, y$  (záporné nemají v biologii smysl) neexistuje žádný bod, okolo kterého by mohlo řešení oscilovat, nebo k němu konvergovat.

Pro řešení kompetitivního Lotkova–Volterrova modelu lze opět využít metodu RK4. Studování dynamiky modelu bude vaším úkolem v seriálové úloze.

Na závěr ještě dodejme, že Lotkovy–Volterrovy diferenciální rovnice lze zobecnit a psát jako soustavu<sup>95</sup>

$$\frac{dn_i}{dt} = n_i f_i(\mathbf{n}),$$

kde

$$\mathbf{f}(\mathbf{x}) = \mathbf{r} + A\mathbf{x}.$$

Složka  $n_i$  vektoru  $\mathbf{n}$  udává velikost  $i$ -té populace. Složky  $r_i$  vektoru  $\mathbf{r}$  jsou růstové koeficienty jednotlivých populací a matice  $A$  popisuje (potenciálně nesymetrické) interakce mezi populacemi. V tomto tvaru můžeme tedy do modelu zavést libovolný počet populací s různými typy interakcí. Například můžeme vyjít z kompetitivního modelu a některým populacím dát záporné interakční koeficienty, čímž budeme simulovat parazitické chování (pokud bude  $I_{ij} < 0$  a zároveň  $I_{ji} < 0$ , půjde o mutualismus), a k tomu přidat populace, které se živí jen predací těch, na nichž jsou závislí paraziti. Tak získáme nepřímou interakci mezi predátory a parazity.

### Co se nevešlo

V průběhu roku jsme vám v seriálu slíbili probrat několik témat, která jsme později jen stručně odbyli, nebo se na ně vůbec nedostalo. Bohužel jsme s tím nemohli nic udělat, protože už v současném stavu je tento seriál zdaleka nejdlejší v historii FYKOSu, a díky častým požadavkům na psaní vlastních počítačových skriptů je

<sup>94</sup>všimněte si, že explicitně neobsahuje nulu (vyhynulou populaci) – pokud  $k_x - I_{xy}k_y = k_y - I_{yx}k_x = 0$ , je i jmenovatel  $1 - I_{xy}I_{yx} = 0$ ; dosazením do (83) zjistíme, že pak je stabilní libovolná populace s  $x/k_x + y/k_y = 1$

<sup>95</sup>Pro ty, kteří již slyšeli o Einsteinově sumační konvenci, pro jistotu uvádíme, že zde se přes stejné indexy nesčítá.

také jedním z nejnáročnějších seriálů, které jsme kdy mezi naše řešitele vypustili. Pokusíme se zde proto vysvětlit, co nás vedlo k vynechání těch konkrétních témat.

V simulační části zcela chybělo pojednání o termodynamickém Monte Carlu (TMC). Tato hojně využívaná metoda by vyžadovala shrnout základy termodynamiky a statistické fyziky, s nimiž se potýkal seriál 29. ročníku, a ještě je trochu rozšířit. A tím jsme vás nechtěli trápit. Zároveň jsme vás proto museli ochudit o rozbor Isingova modelu ferromagnetismu, který se nejlépe řeší právě pomocí TMC.<sup>96</sup> Alternativní přístupy pomocí buněčných automatů bohužel nejsou na pochopení o moc jednodušší. Abyste si nemysleli, že Isingův model složí jenom k modelování ferromagnetismu, tak vězte, že pomocí něj lze předpovídat i úrodu pěstícií.<sup>97</sup>

Stejně tak jsme se z důvodů zvýšené složitosti a protahování délky textu nevěnovali simulacím kritických jevů pomocí buněčných automatů a jevu perkolace. Také jsme vynechali poměrně zábavné simulace přírodních katastrof. Více času jsme také mohli věnovat studiu Brownova pohybu. Celkově lze říct, že prakticky na každém tématu, kterého jsme se během simulací dotkli, si lze založit vědeckou kariéru. Pokud to ovšem člověk s modelováním přírodních jevů myslí vážně, je potřeba strávit pár let studiem matematiky a statistiky. Jak jsme již dříve zmínili, statistické zpracování simulací jsme v seriálu rychle vzdali, neboť činilo řešitelům problémy a ubíralo prostor přitažlivějším tématům. Není však pravda, že by nebylo bez hlubokých matematických znalostí něco pěkného nasimulovat – to jsme se ostatně snažili v tomto textu dokázat. Pokud máte chuť se věnovat simulacím i bez toho, abyste byli vedeni FYKOsem, zkuste se podívat třeba na evoluční teorii her.<sup>98</sup> Pokud jste trochu více pragmatičtí, můžete si zkusit nasimulovat některé jevy v ekonomii, zkuste hledat heslo *Black-Scholes equations*. Pokud se vám líbil aktuální biologicky zaměřený díl, zkuste vyhledat difúzní populační modely či populační modely založené na celulárních automatech. Možností je spousta, stačí si jen vybrat.

### Poděkování

Chtěli bychom vyjádřit velký dík vám všem, kteří jste dočetli až sem, a zároveň vám poblahopřát k vašemu neochvějnému zájmu o počítačovou fyziku. Během psaní seriálu jsme seznali, že je pro většinu řešitelů dosti náročný, což se negativně projevilo na počtu odevzdaných řešení. Ačkoli jsme se postupně snažili psát text více srozumitelně a s menším množstvím matematiky, rostoucí nároky na samostatné programování byly bohužel pro mnohé FYKOSáky velkou překážkou, takže vás do konce vydržela jen hrstka. Avšak o to víc na sebe můžete být hrdí, a věz-

<sup>96</sup> Lze ho řešit i analyticky ale v jedné dimenzi neobsahuje žádné fyzikálně zajímavé chování, ve třech a více dimenzích není exaktně řešitelný a ve dvou dimenzích existuje exaktní Onsagerovo řešení. K tomu řekněme jen tolik, že Landau údajně kdysi prohlásil: „Celou fyziku jsem si přepočítal od základů sám, jen Onsagerovo řešení Isingova modelu jsem si musel přečíst.“

<sup>97</sup> <https://www.matfyz.cz/clanky/>

1111-aktualita-z-fyziky-uroda-pistacii-podle-isingova-modelu

<sup>98</sup> Pěkná a přitažlivě graficky zpracovaná ukázka aplikace této teorie je například na webu <http://ncase.me/trust/>.

te, že nabyté programovací dovednosti v životě ještě mnohokrát zúročíte. Mnoho zdaru s šestou sérií a neutuchající nadšení pro fyziku vám přejí

Mirek Hanzelka a Lukáš Tímko

## Úloha VI.S . . . matice a populace

10 bodů

- a) Na základě Lotkova-Volterrova modelu simulujte vývoj populace predátora a kořisti (např. sluněčka sedmítečného a mšice makové) pro následující hodnoty parametrů:  $r_m = 0,8$ ,  $D_m = 1,0$ ,  $r_s = 0,75$ ,  $D_s = 1,5$ . Počáteční populace volte po dvojicích jako  $m = 0,5$  a  $s = 2,0$ ;  $m = 1,5$  a  $s = 0,5$ ;  $m = 1,95$  a  $s = 0,75$ . Výsledek zanešte do grafu závislosti populace predátora na populaci kořisti. Výsledky diskutujte.
- Bonus* Nalezněte tvar křivek v grafu pomocí analytických metod (integrací diferenciální rovnice).
- b) Použitím kompetitivního Lotkova-Volterrova modelu simulujte vývoj dvou soupeřících populací s omezenou populační kapacitou (např. káně lesní a poštolka obecná) pro tyto hodnoty parametrů:  $r_k = 0,8$ ,  $I_{kp} = 0,2$ ,  $k_k = 2,0$ ,  $r_p = 0,6$ ,  $I_{pk} = 0,3$ ,  $k_p = 1,0$ . Počáteční populace volte jako  $k = 0,01$ ,  $p = 1,0$ . Poté změňte interakční koeficienty na  $I_{kp} = 1,5$  a  $I_{pk} = 0,6$ , zbytek ponechejte. Výsledky zanešte do jednoho grafu závislosti velikosti populací na čase, diskutujte.
- c) Ověřte důležitost pivotizace. Vyřešte soustavu

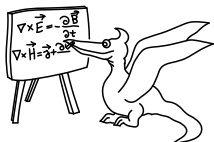
$$\begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

nejprve přesně (na papíře), poté s využitím LU dekompozice s (částečnou) pivotizací (využijte nějakou knihovní funkci, např. `scipy.linalg.lu()`), a nakonec pomocí LU dekompozice bez pivotizace (to si budete muset sami naprogramovat). Porovnejte výsledky  $\mathbf{x}$  z jednotlivých metod a výsledky zpětného vynásobení matic  $L^{-1} \cdot U$  (resp.  $P \cdot L^{-1} \cdot U$  v případě s pivotizací).

- d) Mějme nekonečný deskový kondenzátor se vzdáleností desek  $L = 10$  cm a napětím mezi deskami  $U = 5$  V. Do kondenzátoru vložíme uzemněnou elektrodu ve tvaru nekonečně dlouhého hranolu s čtvercovou podstavou o hraně  $a = 2$  cm, jejíž střed leží  $l = 6,5$  cm od uzemněné desky původního kondenzátoru (tak, že leží mezi deskami). Hranol je orientován tak, že jedna z jeho kratších hran je kolmá k deskám kondenzátoru. Nalezněte průběh elektrického potenciálu v kondenzátoru. Protože je problém symetrický vůči posunu v ose rovnoběžné s nekonečnou hranou hranolu, stačí jej řešit v řezu kolmém k této ose, jde tedy o 2D problém. V této rovině pak získaný průběh potenciálu také vykreslete. K řešení můžete využít program přiložený k zadání.

*Bonus* Vypočtete a vykreslete také průběh velikosti intenzity el. pole  $\mathbf{E}$ .

(řešení str. 275)



## Řešení úloh ze seriálu

### Úloha I.S ... rozjezdová

- a) Upravte výraz  $\sqrt{x+1} - \sqrt{x}$  tak, aby nebyl náchylný k problémům cancellation, ordering a smearing. Ke kterým z těchto problémů byl původně náchylný a proč? Jaký je rozdíl ve výsledku původního a opraveného výrazu, pokud jej vyčíslíme v double precision pro  $x = 1,0 \cdot 10^{10}$ ?
- b) Popište funkci následujícího kódu. Jaký je rozdíl mezi funkcemi `a()` a `b()`? Pro jaké hodnoty  $x$  je lze použít? Nebojte se kód spustit a hrát si s hodnotou proměnné  $x$ . Určete také asymptotickou časovou složitost programu v závislosti na proměnné  $x$ .

```
def a(n):
    if n == 0:
        return 1
    else:
        return n*a(n-1)
def b(n):
    if n == 0:
        return 1.0
    else:
        return n*b(n-1)
x=10
print("{} {} {}".format(x, a(x), b(x)))
```

- c) Označme  $o_k$  a  $O_k$  obvod vepsaného a opsaného pravidelného  $k$ -úhelníku ke kružnici. Pak pro ně platí rekurentní vztahy

$$O_{2k} = \frac{2o_k O_k}{o_k + O_k}, \quad o_{2k} = \sqrt{o_k O_{2k}}.$$

Napište program, který pomocí těchto vztahů vypočítá hodnotu  $\pi$ , začněte přitom s opsaným a vepsaným čtvercem. S jakou přesností dokážete  $\pi$  takto aproximovat? Obdobu tohoto postupu původně navrhl a použil Archimedes.

- d) Lukáš a Mirek hrají hru. Házejí férovou mincí a když padne orel, dá Mirek Lukášovi jedno FYKOSí tričko, když padne panna, dá jedno tričko Lukáš Mirkovi. Oba dohromady mají  $t$  triček, z toho  $l$  patří Lukášovi a  $m$  Mirkovi. Pokud jednomu z hráčů dojdou trička, hra končí.
1. Necht  $m = 3$  a Lukášova zásoba triček je nekonečná. Určete nejpravděpodobnější dobu trvání hry, tedy počet hodů mincí, po nichž hra skončí (protože Mirkovi dojdou trička).

2. Necht  $m = 10$ ,  $l = 20$ . Proveďte simulaci pomocí generátoru pseudonáhodných čísel a nalezněte pravděpodobnost, že Mirek vyhraje všechna Lukášova trička. Celou hru nechejte proběhnout alespoň 100krát (čím více opakování, tím lépe).

3. Jak se změní výsledek předchozí úlohy, jestliže Mirek minci „vylepší“ a panna nyní padá s pravděpodobností  $5/9$ ?

Bonus Vypočtete pravděpodobnosti analyticky a porovnejte výsledek se simulací.

e) Mějme lineární kongruenční generátor s parametry  $a = 65\,539$ ,  $m = 2^{31}$ ,  $c = 0$ .

1. Vygenerujte alespoň 1 000 čísel a spočtete jejich střední hodnotu a rozptyl. Porovnejte se střední hodnotou a rozptylem rovnoměrného rozdělení na stejném intervalu.

2. Nalezněte vztah, který vyjádří číslo v generované sekvenci jako lineární kombinaci čísel na dvou předchozích pozicích, tj. nalezněte koeficienty  $A$ ,  $B$  v rekurentním vztahu  $x_{k+2} = Ax_{k+1} + Bx_k$ . Pokud budeme považovat každá tři po sobě následující čísla za souřadnice bodu ve trojrozměrném prostoru, jak rekurentní vztah ovlivní prostorové rozložení těchto bodů?

Bonus Vygenerujte sekvenci alespoň 10 000 čísel a vykreslete 3D bodový graf, který ilustruje význam uvedeného rekurentního vztahu.

a) Výraz nemůže být náchylný k problémům ordering a smearing, neboť se v něm nevyskytují žádné mnohokrát se opakující operace. Je ale náchylný ke cancellation, protože se v něm odečítají dvě čísla, která jsou si pro velká  $x$  poměrně blízká. Výraz se pokusíme upravit tak, aby se v něm toto odečítání nevyskytovalo. Platí

$$\begin{aligned}\sqrt{x+1} - \sqrt{x} &= \frac{(\sqrt{x+1} - \sqrt{x})(\sqrt{x+1} + \sqrt{x})}{\sqrt{x+1} + \sqrt{x}} \\ &= \frac{x+1-x}{\sqrt{x+1} + \sqrt{x}} = \frac{1}{\sqrt{x+1} + \sqrt{x}},\end{aligned}$$

kde jsme použili vztah  $(a+b)(a-b) = a^2 - b^2$ . Pokud výrazy vyčíslíme pro  $x = 1,0 \cdot 10^{10}$ , dostaneme výsledek  $4,999994416721165 \cdot 10^{-06}$  pro původní výraz a  $4,99999999875 \cdot 10^{-06}$  pro opravený výraz. Jejich rozdíl tedy je zaokrouhleně  $5,58 \cdot 10^{-12}$ . Pokud bychom  $x$  zvětšovali, chyba by ještě rostla, až by při  $x = 1,0 \cdot 10^{16}$  byla chyba stejného řádu, jako výsledek.

b) Funkce  $\mathbf{a}()$  i  $\mathbf{b}()$  počítají faktoriál svého argumentu pomocí rekurentních vztahů  $0! = 1$  a  $n! = n(n-1)!$ . Obě funkce pochopitelně fungují pouze pro celá  $n \geq 0$ . Jediný rozdíl mezi funkcemi je v tom, že zatímco funkce  $\mathbf{a}()$  počítá mezivýsledky i vrací konečný výsledek jako celé číslo, funkce  $\mathbf{b}()$  místo toho používá čísla s plovoucí desetinnou čárkou. Odtud pak plynou veškeré odlišnosti v chování. Pokud program spustíme pro  $x \geq 19$ , zjistíme, že  $x! \gg 10^{16}$ . Takto velké číslo již nelze přesně vyjádřit pomocí typu `double`, výsledek vrácený funkcí  $\mathbf{b}()$  už nemusí tedy být přesný. Oproti tomu celé číslo vrácené funkcí  $\mathbf{a}()$  je stále reprezentováno přesně, jazyk Python totiž dokáže pracovat s neomezeně

velkými celými čísly. Sami si můžete ověřit, že již neplatí podmínka  $a(x)+1 == b(x)+1$ .

Nicméně máme štěstí, faktoriál „velkých“ čísel končí na několik nul<sup>1</sup>, které, když je reprezentace čísla s plovoucí desetinnou čárkou ořízne, výsledek nijak nezmění. Stále tedy platí  $a(x) == b(x)$ . Tato podmínka přestane platit až pro  $x \geq 23$ , kdy se na jednu z ořezávaných pozic dostane nenulová číslice.

Další změna nastane pro  $x \geq 171$ , kdy  $x! \gg 10^{308}$ . Takto velká čísla již vůbec nelze uložit do typu `double`<sup>2</sup> a funkce `b()` místo čísla začne vracet speciální hodnotu `'inf'`, tedy kladné nekonečno. To můžeme vidět ve výpisu na obrazovku, případně si můžeme otestovat platnost podmínky  $b(x) == \text{float}('inf')$ .

Python je v tom, že dokáže ukládat libovolně velká celá čísla, poněkud výjimečný. Většina programovacích jazyků má pevně daný (v daném standardu) počet bytů a tedy pevně danou maximální velikost celočíselných datových typů. Asi nejběžnějším typem je typ o délce 4 byty (32 bitů), do něj lze uložit čísla v rozsahu  $-2\,147\,483\,648 - 2\,147\,483\,647$  (znaménkový typ) nebo  $0 - 4\,294\,967\,295$  (bezznaménkový typ). Pokud bychom implementovali funkci `a()` s použitím 32bitového bezznaménkového celočíselného datového typu, pak by se pro  $x \geq 13$  již výsledek nevešel do rozsahu daného typu<sup>3</sup> a došlo by k tzv. přetečení (integer overflow). To, co se přesně s číslem stane, závisí na konkrétní bitové reprezentaci čísla v počítači a bitové implementaci operace, která přetečení způsobila (nebo to nemusí být definováno vůbec). Nejjednodušší situace je při sčítání, kdy po překročení horní hranice rozsahu datového typu pokračujeme opět od spodní hranice.<sup>4</sup> Obecně ale dostaneme číslo v rozsahu daného datového typu, které nemá žádnou blízkou spojitost se skutečným výsledkem. Protože stále dostáváme nějaké číslo a výpočet dále pokračuje, jako by se nic nestalo, je přetečení obzvlášť zákeřné a při definici proměnných musíme volit takový typ, aby se do nich vešla všechna čísla, která se v nich mohou v průběhu výpočtu vyskytnout.

Všimněme si také, že v tomto případě přetečení celočíselného typu nastane dříve, než typ s plovoucí desetinnou čárkou začne ztrácet přesnost, a o hodně dříve, než se v něm vyskytne hodnota nekonečno. V jazycích, kde nemáme libovolně velké celočíselné typy jako v Pythonu, se tedy mnohdy vyplatí pro výpočty s hodně velkými celými čísly používat typ s plovoucí desetinnou čárkou, pokud nepotřebujeme znát výsledek zcela přesně.<sup>5</sup> Jednou z alternativ je napsat si vlastní neomezený datový typ, se kterým jsou výpočty ale pomalé a žerou hodně paměti.

Na závěr zmiňme, že pro  $x \geq 998$ <sup>6</sup> program selže s chybou `RuntimeError: maximum recursion depth exceeded in comparison`. V těle funkcí `a()` i `b()` totiž voláme (pro  $n \neq 0$ ) tu samou funkci s argumentem o jedna menším, ta

<sup>1</sup>otázka na zamyšlení: jak rychle zjistit, kolik nul?

<sup>2</sup>Takto velký exponent se již do tohoto typu nevejde.

<sup>3</sup> $13! \doteq 6 \cdot 10^9$

<sup>4</sup>Odtud název přetečení.

<sup>5</sup>Tento trik se tedy nehodí např. pro kryptografické operace, pro numerické fyzikální výpočty jej ale můžeme použít téměř vždy.

<sup>6</sup>Hodnota se může lišit v závislosti na konkrétním nastavení prostředí.



zase volá sama sebe, . . . , až se zavolá funkce s argumentem 0. Toto se nazývá *rekurzivní volání funkce*. Program si při zavolání funkce musí zapamatovat, odkud ji zavolal a kam se má po jejím vykonání vrátit.<sup>7</sup> Všechny tyto informace si ukládá do oblasti paměti zvané zásobník (stack, tedy konkrétně call stack), který má omezenou velikost. Při hodně hlubokém rekurzivním volání se pak může stát, že se zásobník zcela naplní a není už kam uložit další iteraci. Tomuto stavu se říká přetečení zásobníku (stack overflow) a jde o kritickou chybu, která nastala i zde.<sup>8</sup> Přetečení zásobníku lze v tomto případě zabránit dvěma způsoby. První spočívá v nastavení větší velikosti zásobníku. Druhý spočívá v přepsání programu tak, aby se v něm nevyskytovaly příliš hluboké rekurze. Například v našem programu lze rekurzi snadno nahradit cyklem, jak vidíme níže.

```
def a_opravena(n):
    faktorial = 1
    for i in range(1,n+1):
        fact *= i
    return faktorial
```

Takto přepsaná funkce by měla fungovat pro libovolně velký argument.

Zmíňme ještě časovou složitost algoritmu. Pro funkce `a()` a `b()` je potřeba pro vstup  $n$  danou funkci rekurzivně zavolat  $(n + 1)$ krát a v každé iteraci provést konstantní počet operací (1 násobení a 1 odčítání). Asymptotická časová složitost je tedy  $O(n)$ . Situace je obdobná i u funkce `a_opravena()`, kde je pro vstup  $n$  potřeba provést  $n$  iterací cyklu, v každé iteraci pak provést konstantní počet operací. Složitost je tedy také lineární. Pro úplnost dodejme, že složitost celého programu je také lineární v závislosti na velikosti  $x$ , protože program jen jednou zavolá funkce `a()` a `b()`, a ty jsou lineární.

- c) Víceméně přesným přepisem rekurentního postupu ze zadání dostaneme kód níže.

```
import math
pocet_iteraci=100
# inicializace ops. a veps. ctvercem
r = 0.5 # polomer kruznice, s touto hodnotou bude obvod = pi
O = 4*2*r
o = 4*r*math.sqrt(2)
k = 4
# rekurentni vypocet
for i in range(pocet_iteraci):
    print("{} {} {}".format(k,o,O))
    O = (2.*o*O)/(o+O)
    o = math.sqrt(o*O)
    k *= 2
```

Může být matoucí, jak v těle cyklu používáme pod stejným označením  $o$  a  $O$  hod-

<sup>7</sup>A vedle toho si musí pamatovat i kontext, např. hodnoty lokálních proměnných ve volajícím místě.

<sup>8</sup>Přesněji zde byla překročena maximální hloubka rekurze, což je hodnota, kterou Python hlídá, aby nenastalo právě přetečení zásobníku.

noty z minulého i aktuálního kroku. Vězte, že lze program přepsat i následovně.

```
import math
pocet_iteraci=100
r = 0.5
O_k = 4*2*r
o_k = 4*r*math.sqrt(2)
k = 4
for i in range(pocet_iteraci):
    print("{} {} {}".format(k,o_k,O_k))
    O_2k = (2.*o_k*O_k)/(o_k+O_k)
    o_2k = math.sqrt(o_k*O_2k)
    o_k = o_2k
    O_k = O_2k
    k *= 2
```

Takto ale zbytečně potřebujeme dvě proměnné navíc.

Pokud program spustíme, zjistíme, že nám skutečně vypisuje hodnoty blízké hodnotě  $\pi$ , což je dobrá zpráva. Vypišme si nyní rozdíly vypočtených hodnot od skutečné hodnoty  $\pi$ . Toho docílíme modifikací řádku s výpisem na obrazovku do podoby `print("{} {} {}".format(k,math.pi-o,math.pi-0))`. Vidíme, že se zvětšujícím se  $k$  oba rozdíly velmi rychle klesají, pro hodnoty  $k \geq 268\,435\,456$  konvergence ustane a rozdíly se zastaví na přibližné hodnotě  $4 \cdot 10^{-16}$ . Tato hodnota odpovídá naakumulované numerické chybě. Jde ale o chybu v řádu strojové přesnosti, což lze při numerickém výpočtu jednoznačně považovat za úspěch.

Možná jste si všimli, že v jedné iteraci je hodnota  $\pi - O$  přesně nulová. To, že je přesně nulová, lze nejspíš považovat za náhodu (stejně to je nula jenom v rámci strojové přesnosti). Náhoda ale není, že jde o nejmenší hodnotu rozdílu, že před touto hodnotou rozdíl klesá a po ní roste či stagnuje, a konečně není náhoda, že se tak děje zrovna pro toto  $k$ . Došlo zde totiž ke kombinaci nízké chyby metody, která klesá s rostoucím  $k$ , a nízké zaokrouhlovací chyby, která s rostoucím počtem operací, a tedy i s rostoucím  $k$ , roste. Tímto jevem se budeme podrobněji zabývat v příštím díle.

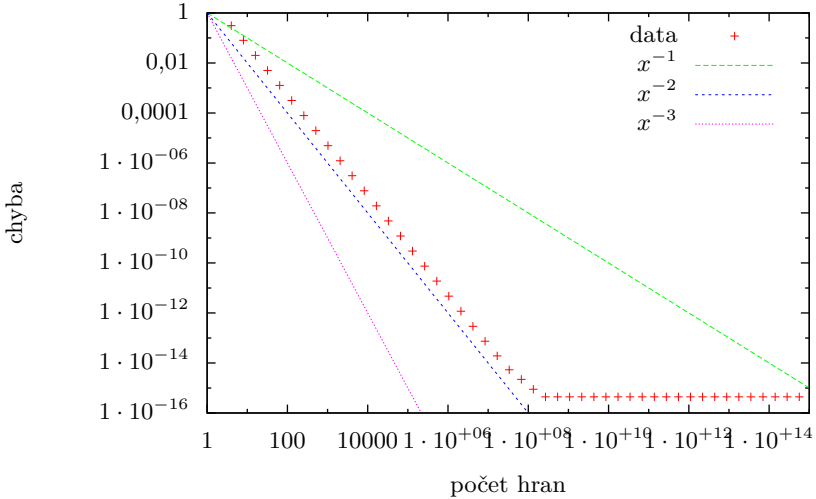
Vykresleme si nyní chyby v závislosti na počtu hran  $k$ , jak je ukázáno v log-log grafu<sup>9</sup> 56. Vidíme, že body v log-log grafu leží na přímce, což znamená, že chyba klesá polynomiálně. Tuto užitečnou transformaci si hned vysvětlíme. Uvažujme polynomiální závislost

$$y = ax^b.$$

Pokud rovnici zlogaritmujeme a označíme  $X = \log x$  a  $Y = \log y$ , dostaneme

$$\begin{aligned}\log y &= \log(ax^b) \\ \log y &= \log a + b \log x \\ Y &= bX + \log a,\end{aligned}$$

<sup>9</sup>Log-log graf je graf, který má na obou osách logaritmickou škálu.



Obr. 56: Závislost chyby metody na počtu hran použitého mnohoúhelníka.

což je rovnice přímky. Ukázali jsme tedy, že v log-log grafu se polynomiální závislosti zobrazují jako přímky, jejichž směrnice odpovídá exponentu závislosti. Poznamenejme, že obdobně lze převést na přímku i exponenciální závislost, jen použijeme graf, který má logaritmickou škálu pouze na svislé ose.

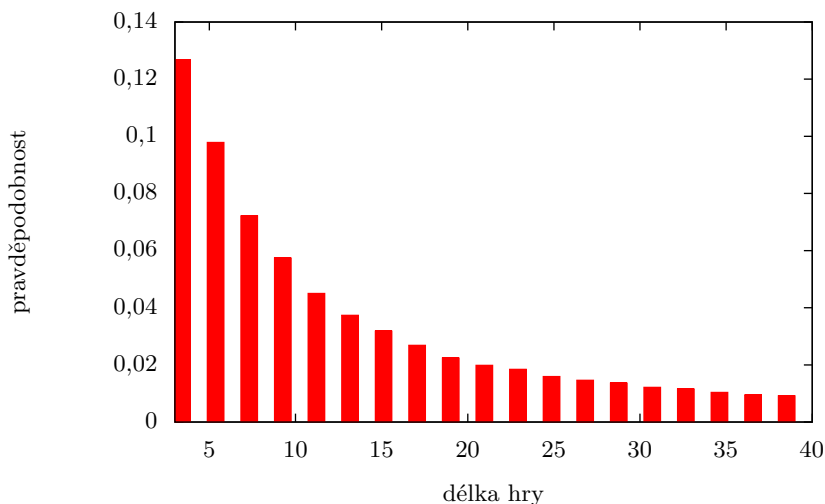
Spolu s daty jsou v grafu 56 vyneseny i závislosti  $y = x^{-1}$ ,  $y = x^{-2}$  a  $y = x^{-3}$ . Vidíme, že data klesají s přibližně stejnou směrnicí jako  $y = x^{-2}$ ,<sup>10</sup> což znamená, že tato metoda je tzv. *kvadratická* v závislosti na počtu hran  $k$ .

Zdůraznit závislost na  $k$  je v tomto případě důležité, neboť běžně se rychlost konvergence udává v závislosti na počtu iterací. Snadným přepočtem zjistíme, že tato rychlost by byla v našem případě exponenciální. V každém případě jde o velmi rychle konvergující metodu, neboť mnoho metod, se kterými se v praxi setkáte, konverguje lineárně nebo pomaleji. Tomuto tvrzení odpovídá fakt, že jsme se k výsledku na úrovni strojové přesnosti dobrali po několika málo iteracích.

- d) 1. Tuto úlohu lze poměrně snadno vyřešit pomocí simulace pro libovolné  $m$ . Necháme počítač odehrát velký počet partií a u každé partie zaznamenáme její délku. Ze získaných délek potom vykreslíme normovaný histogram, tj. graf zobrazující pravděpodobnosti ukončení hry po daném počtu hodů. V našem případě jsme použili  $m = 3$  a  $m = 5$  a celkový počet partií 100 000. Výsledné histogramy jsou na obrázcích 57 a 58. Vidíme, že pro zadané  $m = 3$  skončí hra s největší pravděpodobností již po třech hodech, což je také

<sup>10</sup>Zdůrazňuji, že nám jde o směrnici, ne o přesné proložení dat, neboť závislost může být posunuta o konstantu.

nejmenší počet hodů potřebných k ukončení hry. Oproti tomu pro  $m = 5$  je nejpravděpodobnější délkou hry 7 hodů – to již není triviálně výsledkem. Také si všimneme, že hra musí skončit vždy po lichém počtu hodů. Protože po každém kroku Mirek buď jedno tričko získá, nebo jedno ztratí, má celkový počet jeho triček vždy opačnou paritu než v předchozím kroku. Jestliže začíná s lichým počtem triček, dostane se na nulu (sudé číslo) jedině po lichém počtu hodů.



Obr. 57: Histogram pravděpodobnosti ukončení hry po určitém počtu hodů. Počet Mirkových triček na začátku hry je 3.

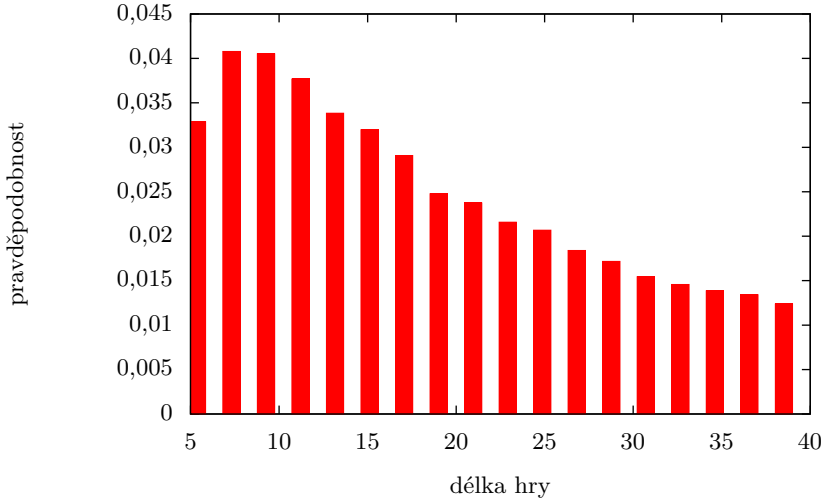
Řešení této úlohy bez pomoci počítače již není tak triviální. Omezíme se pouze na zadané  $m = 3$  a pokusíme se o kombinatorický odhad – pokud si s kombinatorikou netykáte, můžete zbytek řešení přeskočit.

Nejprve si uvědomme následující: Pravděpodobnost, že dojde k realizaci jedné konkrétní hry o délce  $n$  hodů, je  $1/2^n$ . Realizací hry zde myslíme unikátní sekvenci zaznamenávající počet Mirkových triček v každém hodě, která končí první nulou. Např.

$$3 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0. \quad (84)$$

V této konkrétní hře Mirek získal tričko v druhém hodě, v ostatních pouze ztrácel. Jelikož pravděpodobnost hození panny, nebo orla, je  $1/2$ , násobíme za každou šipku tímto faktorem – kdyby byla mince upravená jako v další úloze, výsledek  $1/2^n$  by neplatil.

Nyní předpokládejme, že Mirek může mít i záporný počet triček. Poté si všechny možné realizace hry můžeme zakreslit pomocí orientovaného grafu



Obr. 58: Histogram pravděpodobnosti ukončení hry po určitém počtu hodů. Počet Mirkových triček na začátku hry je 5.

jako na obrázku 59. Vrcholy grafu jsou označeny čísly, která říkají, kolika možnými způsoby je možné se dostat do toho daného stavu hry (stavem rozumíme počet triček a počet odehraných hodů, viz osy grafu v obrázku 59). Hodnoty ve vrcholech odpovídají číslům z Pascalova trojúhelníku. Zaměříme se nyní na vrcholy odpovídající stavu, kdy má Mirek jen jedno tričko. Pokud opět zakážeme záporná trička, tak můžeme tvrdit, že hodnoty v těchto vrcholech budou menší nebo rovny těm vyobrazeným, protože přijdeme o některé hrany grafu (pro hru o více než třech hodech jde o ostrou nerovnost). A pokud budeme chtít, aby hra v následujícím kroku skončila, tak existuje jen jedna hrana vedoucí k tomuto cíli. A jelikož *kté* číslo na *ntém* řádku Pascalova trojúhelníku je určeno binomickým koeficientem

$$\binom{n}{k},$$

bude hodnota ve výše zmíněných vrcholech

$$\binom{n-1}{(n-3)/2}$$

pro délku hry  $n$ . Dále budeme uvažovat následovně: jestliže ukážeme, že pro  $n+2$  je hodnota ve zkoumaných vrcholech menší než čtyřnásobek hodnoty pro  $n$ , a to pro každé  $n \geq 3$ , pak nejpravděpodobnější délkou hry musí

být  $n = 3$ . Čtyřnásobek zde vystupuje proto, že horní odhad pravděpodobnosti získáme vynásobením hodnoty kombinačního čísla pro  $n$  pravděpodobností  $1/2^n$ , s každým prodloužením hry o dva hody se tedy ještě sníží pravděpodobnost jednotlivých realizací hry násobným faktorem  $1/2^2$ . Protože skutečná pravděpodobnost je pro  $n = 3$  stejná jako tento horní odhad, dokážeme tím, že skutečná pravděpodobnost pro obecné  $n > 3$  je taky menší než pro  $n = 3$ .

Pokud ještě provedeme substituci  $n = r + 3$  ( $r$  je sudé), tak se snažíme o důkaz nerovnosti

$$\frac{\binom{r+4}{(r+2)/2}}{\binom{r+2}{r/2}} \leq 4 \quad \forall r \geq 0.$$

Binomické koeficienty rozepíšeme do tvaru

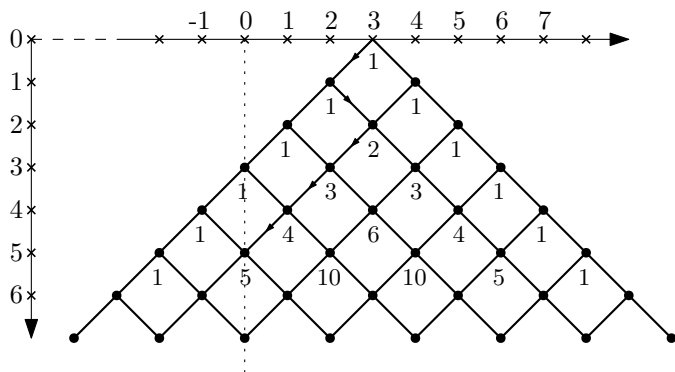
$$\frac{(r+4)!(r/2)!(r/2+2)!}{(r/2+1)!(r/2+3)!(r+2)!} = \frac{(r+3)(r+4)}{(r/2+1)(r/2+3)}$$

dosazením do nerovnosti a odstraněním jmenovatele získáme

$$r^2 + 7r + 12 \leq r^2 + 8r + 12,$$

z čehož už ihned plyne  $r \geq 0$

□.



Obr. 59: Graf, z něž můžeme vyčíst pravděpodobnosti jednotlivých realizací hry za předpokladu, že je povolen záporný počet triček. Vodorovná osa zobrazuje aktuální počet triček, svislá osa počet odehraných hodů. Graf je zkonstruován pro počáteční hodnotu 3 trička. Malými šipkami je v grafu naznačen příklad hry (84).

2. Tato úloha byla velice přímočará. Drobný problém při jejím řešení vám mohla způsobit skutečnost, že hra může skončit po libovolném počtu kroků, takže

některé partie mohou být poměrně dlouhé – pokud jste tedy ve svém kódu nastavili velký počet opakování, je možné, že jste si na výsledek museli počkat.

My jsme volili délky her 100, 1 000, 10 000 a 100 000. Pro tyto hodnoty jsme postupně dostali pravděpodobnost výhry 0,37, 0,342, 0,3366 a 0,33293. Ačkoli při každém běhu programu dostaneme trochu jiný výsledek, lze z výsledků simulace odhadnout, že přesná hodnota hledané pravděpodobnosti bude  $1/3$ . Kód použitý k řešení úlohy naleznete na našich webových stránkách (psán v Pythonu 3).

K vyřešení bonusu použijeme Větu o úplné pravděpodobnosti, která je uvedena v prvním dílu seriálu. Značme  $a = m + l$  a dále si zavedme pravděpodobnost, že Mirek vyhraje ze stavu, kdy má právě  $m$  triček, označme ji  $P_M(m)$ . Zřejmě platí  $P_M(0) = 0$  a  $P_M(a) = 1$ , jde jen o symbolickou formulaci podmínky výhry a prohry (matematicky bychom řekli, že jde o okrajové podmínky). Věta o úplné pravděpodobnosti nyní říká, že pravděpodobnost výhry  $P_M(m)$  je rovna součtu pravděpodobností výhry za podmínky, že padne panna (krát pravděpodobnost, že padne panna) s pravděpodobností výhry za podmínky, že padne orel (krát pravděpodobnost, že padne orel). Toto tvrzení zapíšeme ve tvaru diferenční rovnice

$$P_M(m) = \mu P_M(m+1) + \lambda P_M(m-1),$$

kde  $\mu = 1/2$ ,  $\lambda = 1/2$  označují pravděpodobnosti, že padne panna, resp. orel. Řešení této rovnice snadno uhodneme, když si dosadíme za  $\mu$  a  $\lambda$ . Poté dostaneme

$$P_M(m) = \frac{P_M(m+1) + P_M(m-1)}{2},$$

Jinými slovy: pravděpodobnost, že Mirek vyhraje s  $m$  tričky, že dána aritmetickým průměrem pravděpodobností, že vyhraje s  $m+1$  tričky a s  $m-1$ . Funkce  $P_M(m)$  by tedy mohla ležet na přímce, což zapíšeme formálně jako

$$P_M(m) = A + Bm,$$

přičemž neznámé koeficienty  $A$ ,  $B$  určíme podmínkou výhry/prohry, tj. dosazením  $m = 0$  a  $m = a$  a vyřešením soustavy dvou rovnic o dvou neznámých. Získáme  $A = 0$ ,  $B = 1/a$ . Takže

$$P_M(n) = \frac{n}{a} = \frac{m}{m+l}$$

a pro zadané hodnoty  $m = 10$ ,  $l = 20$  dostaneme pravděpodobnost výhry  $P_M(10) = 1/3$ , což je v souladu s naším předpokladem založeným na výsledcích simulace.

- Úpravou jedné hodnoty v kódu, tedy přepsáním pravděpodobnosti padnutí panny z  $1/2$  na  $5/9$ , jsme dostali výsledek 0,89276 (pro 100 000 her). Na této hodnotě by nás mělo zaujmout především to, že ačkoli se pravděpodobnost padnutí panny zvýšila pouze o  $1/18$ , Mirkova šance na výhru markantně

vzrostla. Tohoto faktu využívají např. kasina v ruletě, kdy šance na výhru při sázce na černou (nebo červenou) je jen o fous menší než  $1/2$ , kvůli zelené nule. Proto, i kdyby mělo kasino méně peněz než hráč (ale řádově více, než je v jedné sázce), tak hráče přesto s velmi vysokou pravděpodobností zruinuje. K řešení bonusové části zde využijeme jeden matematiky hojně používaný trik, se kterým se ale možná ještě někteří z vás nesetkali. Máme opět rovnici

$$P_M(m) = \mu P_M(m+1) + \lambda P_M(m-1),$$

ale nyní jsou  $\mu$  a  $\lambda$  různá, existuje mezi nimi pouze vztah  $\lambda = 1 - \mu$ . Tento typ rovnic se řeší pomocí metody charakteristického polynomu. Budeme předpokládat obecné exponenciální řešení ve tvaru  $\alpha^m$  a dosazením do rovnice a krácením  $\alpha^{m-1}$  získáme kvadratickou rovnici

$$\mu\alpha^2 - \alpha + \lambda = 0,$$

jejíž levá strana se nazývá charakteristický polynom. Řešením této rovnice jsou  $\alpha_1 = 1$ ,  $\alpha_2 = \lambda/\mu$ . Obecné řešení diferenční rovnice získáme jako lineární kombinaci<sup>11</sup>

$$A\alpha_1^m + B\alpha_2^m,$$

tedy

$$P_M(m) = A + B \left(\frac{\lambda}{\mu}\right)^m.$$

Podobně jako výše získáme pomocí okrajových podmínek řešení

$$P_M(m) = \frac{1 - \left(\frac{1-\mu}{\mu}\right)^m}{1 - \left(\frac{1-\mu}{\mu}\right)^{(m+l)}}.$$

Dosazením  $m = 10$ ,  $l = 20$ ,  $\mu = 5/9$  a  $\lambda = 4/9$  získáme číselný výsledek  $P_M(10) \doteq 0,89373$ .

- e) 1. Lineární kongruenční generátor, který jsme použili k řešení úlohy, najdete na našem webu. Pro ty z vás, co programovat ještě neumějí, zde uvedeme, jak takový generátor sestrojít v Excelu.

Do buňky A1 vepíšeme seed, do B1 až D1 postupně hodnoty  $a$ ,  $c$ ,  $m$ . Do buňky A2 poté vepíšeme vzorec

$$=MOD((\$B\$1*A1 + \$C\$1);\$D\$1)$$

a „vytáhneme“ sloupec A tak daleko, jak potřebujeme. Toť vše.

Pomocí generátoru sepsaného v Pythonu jsme z náhodného seedu  $x_0$  (získaného ze systémového času) vygenerovali  $10^6$  pseudonáhodných čísel a vypočetli průměrnou hodnotu  $\bar{x} \doteq 0,499596$  a rozptyl  $\sigma^2 \doteq 0,083209$  (směrodatná odchylka je potom  $\sigma \doteq 0,28846$ ).

<sup>11</sup> pokud se vám nezdá, že pro  $\lambda = \mu$  jsme dosazovali lineární funkci a tady exponenciální, vezte, že případ se stejnými kořeny je speciální – právě díky tomu, že z nich neuděláme lineární kombinaci



Srovnajme nyní výsledky simulace s přesnými výrazy pro střední hodnotu a rozptyl uvedenými v seriálu. Střední hodnota je

$$\frac{0 + (2^{31} - 1)}{2},$$

po normalizaci  $N = 2^{31}$

$$\bar{x} = \frac{0 + (2^{31} - 1)}{2 \cdot 2^{31}} \doteq 0.49999999977.$$

Když do vztahu pro rozptyl

$$\sigma^2 = \sum_i (x_i - \mu)^2 p_i$$

dosadíme normalizované hodnoty, vidíme, že normalizovat rozptyl znamená dělit  $N^2$ , takže pro rovnoměrné rozdělení máme

$$\sigma^2 = \frac{(2^{31} - 1)^2}{12 \cdot 2^{62}} \doteq 0.08333333325572312.$$

Ověřili jsme tedy, že vygenerované rozdělení má podobnou střední hodnotu a rozptyl jako rovnoměrné rozdělení. Ještě poznamenejme, že střední hodnota a rozptyl obecně nestačí pro jednoznačné určení pravděpodobnostního rozdělení (museli bychom znát všechny jeho momenty). A už vůbec nám shoda v těchto dvou veličinách nezaručuje, že generátor má dobré náhodné vlastnosti. Pokud jste si nechali vypsát pseudonáhodná čísla na obrazovku, tak jste si určitě všimli, že jsou všechna lichá, což nám indikuje, že se nejedná o dobrý generátor. Více viz následující úloha.

2. Zásadním krokem v této úloze je uvědomit si, že číslo 65 539 lze zapsat ve tvaru  $2^{16} + 3$ . Poté můžeme libovolné číslo v sekvenci napsat ve tvaru

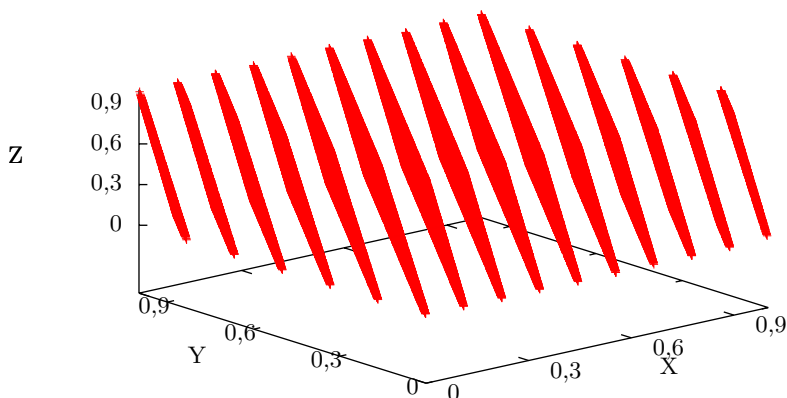
$$\begin{aligned} x_{i+2} &= ((2^{16} + 3) \cdot x_{i+1}) \bmod 2^{31} \\ &= ((2^{16} + 3) \cdot ((2^{16} + 3) x_i) \bmod 2^{31}) \bmod 2^{31} \\ &= (2^{16} + 3)^2 x_i \bmod 2^{31}. \end{aligned}$$

Postupnými úpravami dostaneme, že modulo  $2^{31}$  platí

$$\begin{aligned} x_{i+2} &= (2^{32} + 6 \cdot 2^{16} + 9) x_i = (6 \cdot 2^{16} + 9) x_i \\ &= (6(2^{16} + 3) - 9) x_i = 6x_{i+1} - 9x_i, \end{aligned}$$

kde jsme v druhém kroku využili  $2^{32} \bmod 2^{31} = 0$ . Pokud nyní budeme tři následující čísla v sekvenci chápat jako kartézské souřadnice  $x = x_i$ ,  $y = x_{i+1}$ ,  $z = x_{i+2}$ , pak je odvozený rekurentní vztah rovnicí roviny

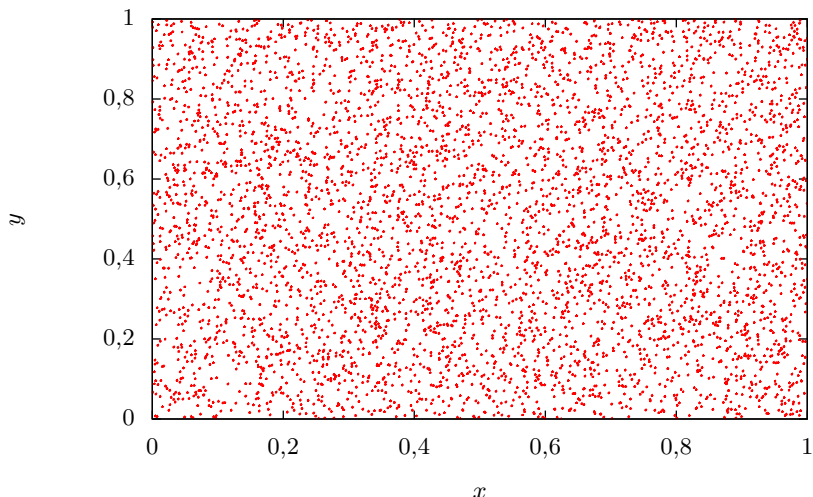
$$z = 6y - 9x.$$



Obr. 60: 1 000 000 pseudonáhodných čísel vykreslených ve 3D. Jistě byste nechtěli, aby takovýto generátor používala v bezpečnostních systémech vaše banka.

V trojrozměrném prostoru tedy budou všechna vygenerovaná pseudonáhodná čísla ležet v rovině. Ještě si můžeme rozmyslet, kolikrát se v boxu  $1 \times 1 \times 1$  rovina díky modulární aritmetice „zlomí“. V průmětech do rovin  $yz$  ( $x = 0$ ) a  $xz$  ( $y = 0$ ) dostaneme rovnice přímek  $z = 6y$  a  $z = -9x$ . Stěna boxu ležící na ose  $y$  bude tedy protnuta šestkrát a stěna ležící na ose  $x$  devětkrát. Celkem tedy bude v boxu patnáct rovin. To si konečkonců můžete potvrdit na řešení bonusové úlohy, které vidíte na obrázku 60.

Možná vás napadlo, že jelikož sekvence generovaná libovolným LCG (lineárním kongruenčním generátorem) má konečnou periodou, tak vždy bude možné najít nějakou korelaci mezi následujícími členy, která významně naruší náhodnost. Tzv. Marsagliaův teorém říká, že pokud čísla generovaná lineárním kongruenčním generátorem vykreslíme jako body  $n$ dimenzionálního prostoru, tak budou spadat do nejvýše  $(n!)^{1/n}$  nadrovin o dimenzi  $n - 1$ . Proto, zatímco v nízkých dimenzích jsou LCG obvykle spolehlivé (i zde používaný generátor, známý pod názvem RANDU, se chová dobře ve dvou dimenzích, viz obr. 61), ve vyšších dimenzích postupně selhávají.



Obr. 61: 10 000 pseudonáhodných čísel vykreslených ve 2D. V této dimenzi se již polohy bodů nezdaří být předvídatelné.

## Úloha II.S . . . derivace a Monte Carlo integrace

a) Vykreslete závislost chyby na velikosti kroku pro metodu

$$f'(x) \approx \frac{-f(x+2h) + f(x-2h) + 8f(x+h) - 8f(x-h)}{12h}$$

odvozenou pomocí Richardsonovy extrapolace v textu seriálu. Jaký je optimální krok a minimální chyba? Porovnejte s centrovanou a dopřednou diferencí. Jako derivovanou funkci použijte  $\exp(\sin(x))$  v bodě  $x = 1$ .

Bonus Vypočtěte pro tuto metodu teoretickou velikost optimálního kroku pomocí odhadu chyb.

b) Na webu se nachází soubor s experimentálně zjištěnými  $t$ ,  $x$  a  $y$  souřadnicemi poloh hmotného bodu. Pomocí numerické derivace nalezněte časovou závislost složek rychlosti a zrychlení a vynesete obě závislosti do grafu. Jaký fyzikální děj bod nejspíše konal? Numerickou metodu si zvolte sami, svoji volbu ale odůvodněte.

Bonus Existuje v tomto případě přesnější varianta získání rychlosti a zrychlení, než přímočará aplikace numerické derivace?

c) Máme zadán integrál  $\int_0^\pi \sin^2 x dx$ .

1. Nalezněte hodnotu integrálu z geometrické úvahy za pomoci Pythagorovy věty.

2. Nalezněte hodnotu integrálu pomocí Monte Carlo simulace. Určete směrodatnou odchylku výsledku.

Bonus Vyřešte Buffonovu úlohu ze seriálu (odhad hodnoty čísla  $\pi$ ) pomocí MC simulace.

d) Nalezněte vztah pro výpočet objemu šestidimenzionální koule pomocí metody Monte Carlo.

Nápověda Pythagorovu větu lze využít k měření vzdáleností i ve vyšších dimenzích.

a) Abychom měli chybu podle čeho určovat, potřebujeme znát analyticky vyjádřenou derivaci testovací funkce.<sup>12</sup> Z pravidla o derivaci složené funkce snadno zjistíme, že platí

$$(\exp(\sin(x)))' = \cos(x) \exp(\sin(x)).$$

Program, který vypíše různé kroky  $h$  a příslušné chyby tedy vypadá např. takto.

```
import math
def diff4(f,x,h):
    return(-f(x+2*h)+f(x-2*h)+8*f(x+h)-8*f(x-h))/(12*h)
h=1.
x=1.
points=100
f=lambda x: math.exp(math.sin(x))
df=lambda x: math.cos(x)*math.exp(math.sin(x))
for i in range(points):
    print("{} {}".format(h, math.fabs(df(x)-diff4(f,x,h))))
    h/=2
```

Závislost chyby na velikosti kroku je pak vynesena v grafu 62. Z grafu vidíme, že optimální krok je  $h_{\text{opt}} \doteq 5 \cdot 10^{-4}$  a odpovídající chyba numerické derivace  $1 \cdot 10^{-13}$ , což je méně než pro centrovanou a dopřednou diferenci. Optimální krok je naopak v souladu s tvrzením v seriálu větší než u dopředné a centrované diference. Také si všimněme, že chyba metody klesá jako  $O(h^4)$ , což jsme očekávali.

Nyní zmiňme řešení bonusu. Zaokrouhlovací chybu spočítáme obdobně jako v textu seriálu. Platí pro ni

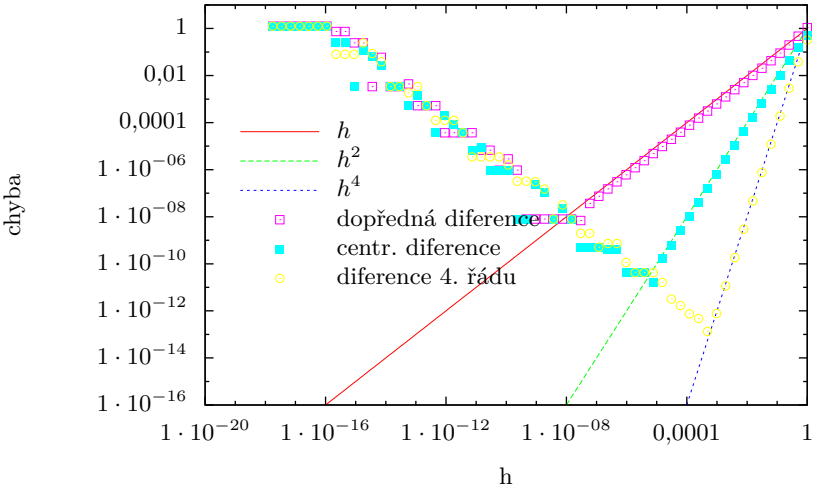
$$\Delta f'(x) = \frac{-f(x+2h)\varepsilon_1 + f(x-2h)\varepsilon_2 + 8f(x+h)\varepsilon_3 - 8f(x-h)\varepsilon_4}{12h},$$

$$\Delta f'(x) \approx \frac{18\varepsilon|f(x)|}{12h} = \frac{3\varepsilon|f(x)|}{2h},$$

kde  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\varepsilon_3$  a  $\varepsilon_4$  jsou skutečné relativní chyby vyčíslení funkce a  $\varepsilon \approx \varepsilon_{1..4}$  je strojová přesnost. Povšimněme si, že pro zaokrouhlovací chybu znova platí  $\Delta f'(x) \propto h^{-1}$ .

---

<sup>12</sup>Pokud bychom ji neznali, lze jako odhad použít numerickou metodu vyššího řádu.



Obr. 62: Závislost chyby derivace na kroku  $h$  pro různé metody. Jako testovací funkce byla zvolena  $f(x) = \exp(\sin(x))$  v bodě  $x = 1$ .

Chyba metody pak z Taylorova rozvoje je

$$\delta = \frac{48|f^{(5)}(\xi)|}{5! \cdot 12} h^4 \approx \frac{|f^{(5)}(x)|}{30} h^4,$$

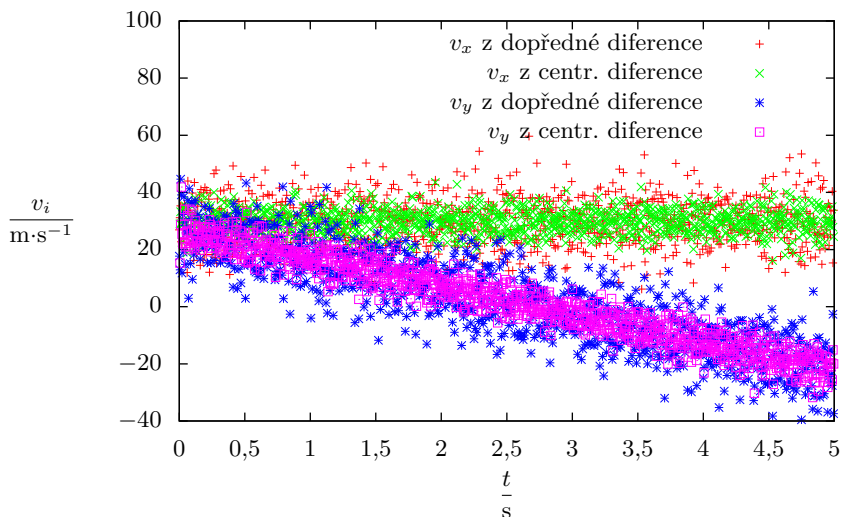
kde  $\xi$  je nějaké (správné) číslo mezi  $x$  a  $x+h$ . Z podmínky pro minimální chybu  $\delta \approx \Delta f'(x)$  máme

$$h_{\text{opt}} \approx \sqrt[5]{45\varepsilon \frac{|f(x)|}{|f^{(5)}(x)|}} \sim \varepsilon^{1/5} \sim 10^{-3},$$

což řádově odpovídá našemu „experimentálně“ získanému výsledku. Konkrétně pro  $x = 1$  (v radiánech) ještě můžeme dopočítat  $f(x) \doteq 2,3$ ,  $f^{(5)}(x) \doteq 23,8$  a  $h_{\text{opt}} \doteq 8,5 \cdot 10^{-4}$ .

- b) Přístupme k řešení problému obráceně, nejprve vypočtíme rychlost a zrychlení pomocí dopředné a centrované diference a poté se výsledky pokusme interpretovat.

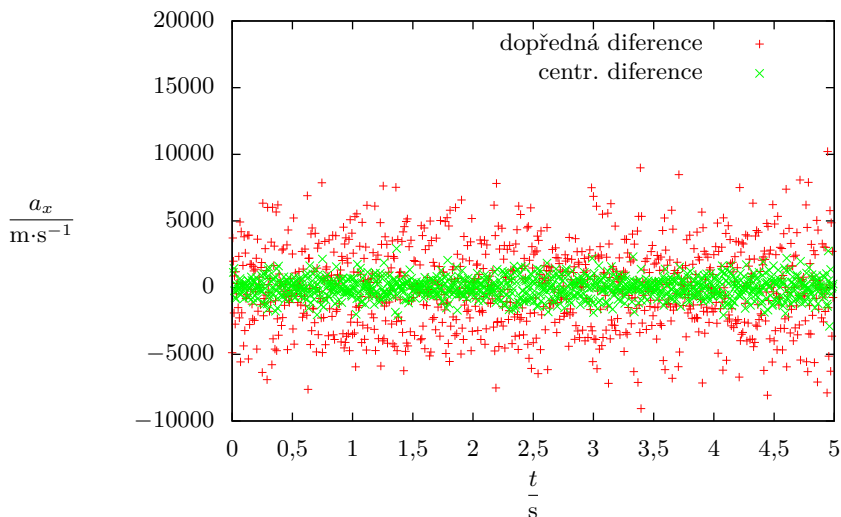
Jak vidíme z grafů rychlosti 63 a zrychlení 64, 65, v *x*ovém směru bod vykonává rovnoměrný pohyb rychlostí asi  $30 \text{ m}\cdot\text{s}^{-1}$  a v *y*ovém směru rovnoměrně zrychlený přímočarý pohyb s počáteční rychlostí asi  $27 \text{ m}\cdot\text{s}^{-1}$  a zrychlením asi  $-10 \text{ m}\cdot\text{s}^{-2}$ , přičemž zrychlení nedokážeme díky šumu vyčíst rovnou z grafu zrychlení, ale dokážeme jej určit výpočtem z grafu rychlosti. Vidíme tedy, že se jedná o případ vrhu v homogenním tíhovém poli.



Obr. 63: Závislost složek rychlosti na čase získaná dopřednou a centrovanou diferencí.

Také vidíme, že centrovaná diference má menší rozptyl než dopředná diference. Protože druhá a vyšší derivace funkce  $x(t) = 30 \cdot t$  podle času je rovna nule, dělíme ve vztahu pro optimální krok dopředné i centrované diference nulou a optimální krok tedy neexistuje, vždy jsme v režimu převládající zaokrouhlovací chyby. Obdobně pro  $y(t)$  je nulová až třetí derivace, dopředná diference tedy má (pro  $t = 1$  s)  $h_{\text{opt}} \approx 10^{-8}$  s, ale centrovaná diference opět optimální krok nemá. To si můžete lehce ověřit vyřešením předchozí podúlohy pro testovací funkce  $x(t) = v_x t$  a  $y(t) = v_y t - 1/2gt^2$ , kde  $v_x$ ,  $v_y$  a  $g$  jsou konstanty.

Protože pro  $x(t)$  jsou obě metody v režimu s převládající zaokrouhlovací chybou, měly by mít srovnatelnou přesnost, což ale očividně neplatí. To napovídá, že zde existuje nějaký efekt, který je daleko větší než vliv zaokrouhlovací chyby a chyby metody. Pokud se pozorně podíváme na předchozí data či si pořádně přečteme zadání, zjistíme, že neleží přesně na teoretické závislosti, ale, jako správná experimentální data, se kolem ní vyskytují s nějakou nepřesností měření. Centrovaná diference je pak přesnější, jak je vidět z následujícího příkladu. Představte si, že by naše závislost polohy byla konstantní (ale stále s „experimentálním“ rozptylem  $\sigma$ ). Pevně nyní zvolme nějakou vysokou pravděpodobnost  $P$ , třeba 99%. Pak existuje číslo  $n > 0$  takové, že dva po sobě následující body leží s pravděpodobností  $P$  na bočních stranách obdélníku o podstavě  $h$  a výšce  $n\sigma$ . Pokud je spojíme úsečkou, bude mít tato úsečka směrnici, tím pádem i chybu derivace, menší než  $n\sigma/h$ . Pokud ale použijeme centrovanou



Obr. 64: Závislost  $x$ ové složky zrychlení na čase získaná dopřednou a centrovanou diferencí.

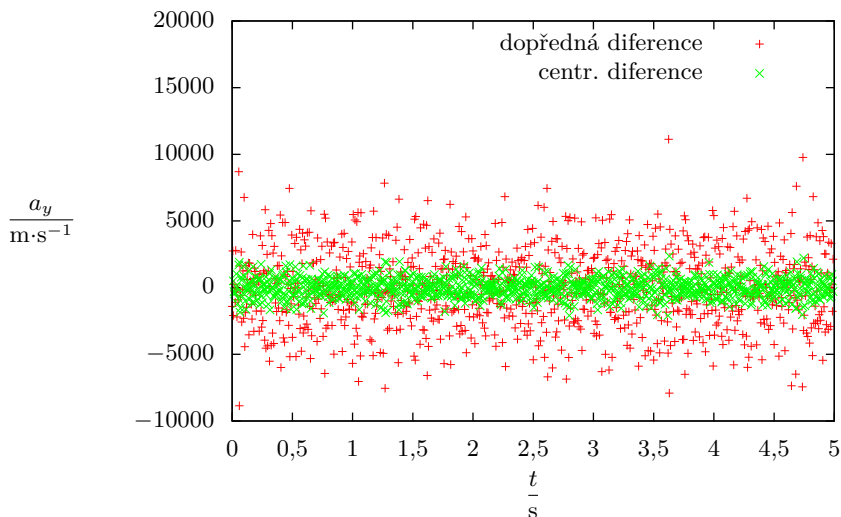
diferenci,<sup>13</sup> používáme body „ob jedna“, které s pravděpodobností  $P$  leží na bočních stranách obdélníku o podstavě  $2h$  a výšce  $n\sigma$ . Směrnice jejich spojnice bude nanejvýš  $n\sigma/(2h)$ , chyba tedy bude poloviční.

Obecně dle výše uvedeného by mělo platit, že čím je metoda vyššího řádu, tím lépe bude fungovat. Tomu navíc pomůže to, že metody vyššího řádu pracují s funkčními hodnotami ve více než dvou bodech, efektivně tedy v sobě obsahují obdobu váženého průměru, který by měl napomoci vyrušení rozptýlu hodnot. Všimněme si také, že rozptýl původních hodnot polohy je velmi malý, odpovídá měření s relativní odchylkou 0,02 %, což je velmi přesné měření. Přesto se tato odchylka aplikací derivace nesmírně zvětší. To je způsobeno tím, že numerická derivace není dobře podmíněná úloha, použité metody tedy nemohou být numericky stabilní.

Pokud bychom chtěli náš výsledek zpřesnit, mohli bychom toho docílit různými způsoby. Dle výše uvedeného by mělo pomoci zvětšení kroku přesakováním některých hodnot, či použití metody vyššího řádu, přičemž druhá možnost by měla fungovat lépe. Další možností je na data před derivací nejprve aplikovat tzv. klouzavý průměr<sup>14</sup>, čímž bychom zredukovali rozptýl hodnot. Jak již ale bylo zmíněno, metody vyšších řádů už v sobě jistý způsob průměrování efektivně obsahují, jednoduchým neváženým klouzavým průměrem bychom tedy nejspíš lepších výsledků nedosáhli. Ve chvíli, kdy známe teoretickou závislost, podle

<sup>13</sup>Stejný efekt by měla i dopředná diference s dvojnásobným krokem.

<sup>14</sup>[https://en.wikipedia.org/wiki/Moving\\_average](https://en.wikipedia.org/wiki/Moving_average)



Obr. 65: Závislost  $y$ ové složky zrychlení na čase získaná dopřednou a centrovanou diferencí.

kteří se data mají chovat, je asi nejlepším řešením data touto závislostí nafitovat a poté analyticky derivovat teoretickou závislost s koeficienty určenými z fitu.

- c) Zadaný integrál lze snadno spočítat například za pomoci goniometrického vzta-  
hu  $\sin^2 x = (1 - \cos(2x))/2$  a substituce  $y = 2x$ . Dostaneme

$$I = \int_0^{\pi} \sin^2 x dx = \int_0^{\pi} \frac{1}{2} dx - \int_0^{2\pi} \frac{1}{4} \cos y dy = \frac{\pi}{2}.$$

Zadání nám však ukládá postupovat bez znalosti integrálního počtu, pojďme se o to tedy pokusit.

- Pro kartézské souřadnice  $a$ ,  $b$  bodu na jednotkové kružnici, představující délky odvěsen pravouhlého trojúhelníku, platí Pythagorova věta ve tvaru  $a^2 + b^2 = 1$ . Neboli, vyjádřeno pomocí úhlu při středu kružnice mezi přepnou a jednou odvěsnou  $x$ ,  $\sin^2 x + \cos^2 x = 1$ . Dále víme, že určitý integrál  $I$  vyjadřuje obsah útvaru vymezeného funkcí  $\sin^2 x$  a osou  $x$  na intervalu  $\langle 0, \pi \rangle$ , tj. obsah jednoho „kopečku“ funkce  $\sin^2 x$ .

Když k funkci  $\sin^2 x$  všude připočteme funkci  $\cos^2 x$ , dostaneme díky Pythagorově větě místo kopečku obdélník o stranách  $\pi$  a 1, tedy o obsahu  $\pi$ . Nyní si už jen zbývá uvědomit, že funkce sinus a kosinus jsou identické až na posun o  $\pi/2$  v ose  $x$  (a tedy i jejich kvadráty). Jeden kopeček kosinu na druhou má tedy stejný obsah jako kopeček sinu na druhou. Jeden kopeček má tedy obsah přesně dvakrát menší než obdélník, a tedy  $I = \pi/2 \doteq 1,5708$ , což je v souladu s výsledkem výše.



2. K numerickému výpočtu použijeme kód na výpočet Monte Carlo integrace ze seriálu.<sup>15</sup> Generujeme dvojice pseudonáhodných čísel  $\langle x, y \rangle$  na intervalech  $\langle 0, \pi \rangle$  a  $\langle 0, 1 \rangle$ , kritériem pro zásah je  $y < \sin^2 x$ . Ve výpočtu směrodatné odchylky je nyní  $V = \pi$  a  $O = \pi/2$ , relativní směrodatnou odchylku pak vyjádříme jako

$$\frac{\sigma}{O} = \frac{1}{\sqrt{N}} \sqrt{(V/O) - 1} = 0,001.$$

Pro jeden konkrétní běh programu s  $N = 10^6$  jsme dostali výsledek 1,569 s relativní odchylkou od průměru 0,0012.

Při řešení bonusové úlohy, odhad  $\pi$  pomocí házení jehly, si také vystačíme s mírnou úpravou kódu ze seriálu. Budeme volit  $a = b = 1$ , tzn. stejnou šířku pásu jako délku jehly.<sup>16</sup> Čím menší jehlu bychom volili, tím méně častěji by prořala hranici pásů a tím déle by musela simulace běžet, abychom dosáhli dané přesnosti. Dále víme, že stačí generovat dvojice pseudonáhodných čísel  $\langle \vartheta, x \rangle$  z intervalů  $\langle -\pi/2, \pi/2 \rangle$  a  $\langle 0, 1/2 \rangle$ , neboť tak pokryjeme všechny možné horizontální polohy středu jehly a veškerá možná její natočení. Není potřeba uvažovat další pásy, protože se periodicky opakují. Stejně tak nepotřebujeme rozlišovat, zda jehla leží napravo nebo nalevo o hranice, kterou prořala. Kritérium protnutí vyjádříme tak, že horizontální souřadnice levého konce jehly  $x - (\cos \vartheta)/2$  je menší než nula.

Ze seriálu víme, že pro  $a = b = 1$  je pravděpodobnost protnutí  $2/\pi \doteq 0,6366$ . Pro jeden běh programu s  $N = 10^6$  jsme dostali výsledek 0,6368 s relativní směrodatnou odchylkou (teoretickou) 0,0008 a relativní odchylkou od průměru 0,0004.

- d) Objem koule v libovolné dimenzi musí záviset pouze na jejím poloměru  $R$ , neboť žádnou jinou informaci o kouli nemáme. Z jednotkových důvodů musí být poloměr umocněn na číslo udávající dimenzionalitu, v našem případě máme  $R^6$ . Tento mocninný výraz bude vynásoben konstantním faktorem, který obecně může být různý pro každou dimenzi. Volbou  $R = 1$  se naše úloha omezuje na hledání tohoto faktoru.

Objem  $n$ -dimenzionální koule o poloměru  $R$  lze vyjádřit vzorcem

$$V_n = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} R^n,$$

kde  $\Gamma$  je gama funkce. Nebudeme se zde zabývat její definicí, pouze si řekněme, že pro přirozené číslo  $n$  lze psát  $\Gamma(n + 1) = n!$ . Nás zajímá dimenze 6, argumentem gama funkce tedy bude přirozené číslo a můžeme psát

$$V_6 = \frac{\pi^3}{3!} R^6.$$

<sup>15</sup> Veškeré kódy použité v řešení této seriálové úlohy naleznete na našem webu.

<sup>16</sup> V seriálu jsme psali podmínku  $a < b$ , ale vzhledem k omezené přesnosti reprezentace čísel v počítači toto nemá smysl rozlišovat.

Hledaný faktor je tedy roven  $\pi^3/3! \doteq 5,1677$ .

S uvedeným faktorem nyní srovnáme výsledek získaný metodou Monte Carlo. Použili jsme kód ze seriálu pro výpočet obsahu čtvrtkruhu s mírnými úpravami. Počet bodů jsme zvýšili na  $10^7$  a pomocí funkce random nyní nevytváříme dvojice, ale šestice náhodných čísel. V boxu o hraně 1 (šestidimenzionální krychle) nyní není uzavřena čtvrtina kruhu, ale  $1/2^6$  z objemu koule. Kritériem zásahu dovnitř objemu koule je výraz

$$\sum_{i=1}^6 x_i^2 < 1,$$

kde  $x_i$ ,  $i \in \{1, 2, 3, 4, 5, 6\}$  jsou kartézské souřadnice náhodných bodů. Na základě znalosti přesného výsledku můžeme vypočíst směrodatnou odchylku

$$\sigma = \frac{1}{\sqrt{N}} \sqrt{O(V - O)},$$

kde  $V = 1^6 = 1$  a  $O = \pi^3 \cdot 2^{-6} \cdot 3!$ , číselně tedy získáme relativní odchylku  $\sigma/O \doteq 0.0011$ . Zde vidíme, že poměr  $O/V$  je již celkem nevýhodný a museli jsme proto volit o řád větší počet bodů (v porovnání s 2D případem), abychom dosáhli směrodatné odchylky  $\sim 10^{-3}$ . Výsledkem jednoho běhu programu byla průměrná hodnota hledaného faktoru 5,172 (relativní odchylka od průměru je 0,0008).

### Úloha III.S ... na procházce s integrály

- Vymyslete tři odlišné příklady markovovského procesu, z toho alespoň jeden fyzikální. Je procházka bez návratu markovovská? A co procházka bez křížení?
- Mějme 2D náhodnou procházku bez návratu na čtvercové síti s počátkem v bodě  $(x, y) = (0, 0)$ , která je omezena absorpčními bariérami  $b_1: y = -5$ ,  $b_2: y = 10$ . Nalezněte pravděpodobnost, že v bariéře  $b_1$  skončíme dříve než v  $b_2$ .
- Provedte simulaci pohybu brownovské částice ve 2D a vykreslete graf závislosti střední vzdálenosti od počátku na čase. Uvažujeme diskrétní čas a konstantní délku kroku (jeden krok simulace trvá  $\Delta t = \text{konst}$ , délka kroku je  $\Delta l = \text{konst}$ ) a umožňujeme pohyb do libovolného směru, tj. každý krok je specifikován délkou a úhlem  $\vartheta \in \langle 0, 2\pi \rangle$ , přičemž všechny směry jsou stejně pravděpodobné. Zajímá nás především asymptotické chování, tedy vývoj střední vzdálenosti pro  $t \gg \Delta t$ .
- Chybová funkce je definována vztahem

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Tabelujte tuto funkci, tedy vypočtěte integrál pro mnoho různých  $x$ . Do řešení nevkládejte tabulku hodnot, ale graf funkce. Zkuste tuto funkci opět numericky zderivovat. Co dostanete?

- e) Najděte si definici hustoty pravděpodobnosti Maxwellova-Boltzmannova rozdělení  $f(v)$ , tedy rozdělení rychlostí molekul ideálního plynu. Spočítejte pak pomocí MC integrace střední hodnotu rychlosti definovanou

$$\langle v \rangle = \int_0^{\infty} v f(v) dv,$$

přičemž pro vzorkování použijte náhodná čísla dle Maxwellova-Boltzmannova rozdělení získaná Metropolisovým-Hastingsovým algoritmem. Hodnotu pro konkrétní zvolené parametry srovnajte s hodnotou z literatury.

- a) Připomeňme, že Markovův proces jsme definovali jako náhodný proces, který nám umožňuje určit pravděpodobnost realizace určitého stavu systému (stav je dán hodnotou náhodné veličiny) v čase  $t_{i+1}$  zcela na základě znalosti stavu v čase  $t_i$ . Uvedme tři příklady Markovova procesu: matematický, fyzikální a „z běžného života“.

- (a) Mějme klobouk  $A$  obsahující 10 černých koulí a klobouk  $B$ , v němž je 10 bílých koulí. V každém kroku náhodně vybereme jednu kouli z klobouku  $A$  a jednu z klobouku  $B$  a vyměníme je. Sekvence náhodných proměnných představujících poměr černých ku bílým koulím v klobouku  $A$  je náhodný proces. Zároveň se jedná i o Markovův proces/řetězec, neboť znalost aktuálního (v čase  $t_i$ ) poměru koulí v klobouku  $A$  jednoznačně určává i poměr koulí v klobouku  $B$  a na základě těchto znalostí dokážeme určit pravděpodobnost, s jakou v klobouku  $A$  vzroste v dalším časovém kroku počet koulí jedné barvy o jedna, klesne o jedna, nebo se nezmění.
- (b) Fyzika je aplikací matematických modelů na dění kolem nás. Proto potřebujeme najít přírodní děj, pro jehož popis se používají náhodné procesy. Příkladem může být například vývoj počtu částic radionuklidu. Pokud známe počet částic v čase  $t$ , dokážeme určit pravděpodobnostní rozdělení v čase  $t + dt$  (zde bychom správně měli rozšířit naši definici na spojité plynutí času, můžeme ale uvažovat i model s konečnými časovými kroky  $\Delta t$  a konečnou pravděpodobností rozpadu každé částice v jednom kroku).
- (c) Označíme-li každý den jako slunečný, nebo deštivý, a víme, s jakou pravděpodobností přejde slunečný den v deštivý a naopak (a nebo nedojde ke změně), pak na základě dnešního stavu počasí dokážeme říct, s jakou pravděpodobností bude zítra slunečno či deštivo.

Třetí příklad je dobrou ukázkou toho, že každý jev lze modelovat jako markovovský. Totiž, abychom získali pravděpodobnost přechodu slunečného dne v deštivý, museli jsme provést statistiku na velkém počtu předchozích dní. Lze tedy namítnout, že na určení pravděpodobností realizace určitého stavu v příštím časovém kroku jsme potřebovali znát všechny předchozí. Podobně lze argumentovat, že téměř žádný fyzikální proces není markovovský, protože pro výpočet zrychlení z pohybové rovnice potřebujeme znát nejen polohy, ale i rychlosti, a ty získáme měřením polohy ve více časových bodech. Proto, když se mluví o (ne)markovosti určitého fyzikálního jevu, je vždy potřeba zmínit model, kterým jev popisujeme.

Co se týče zbylých dvou otázek na markovovskost náhodných procházek, ani jedna z uvedených nemá Markovovu vlastnost. Náhodná procházka bez návratu si kromě aktuální pozice vždy musí pamatovat i tu předešlou a náhodná procházka bez křížení musí mít v paměti celou trajektorie, tj. všechny pozice za celou dobu vývoje. Za markovovské bychom mohli tyto procházky považovat použitím „podvodu“ popsaného výše, kdy bychom za aktuální stav procházky prohlásili např. celou její trajektorii.

- b) K nalezení pravděpodobnosti, že procházka skončí dříve v dolní než horní bariéře, použijeme kód z textu seriálu. Do něj přidáme podmínky zastavení výpočtu a proměnnou, která obsahuje počet procházek  $L$ , které skončily v dolní bariéře. Také musíme přidat identifikátor, který si zapamatuje, jakým směrem byl proveden poslední krok, a nepovolí krok opačným směrem jako následující – tak zajistíme, že půjde skutečně o procházku bez okamžitého návratu. Hledaným řešením je poměr  $L/N$ , kde  $N$  je celkový počet procházek. Délku procházky přitom neomezujeme, protože bariéry jsou poměrně blízko počátku, a tedy procházka skončí po rozumném počtu kroků.

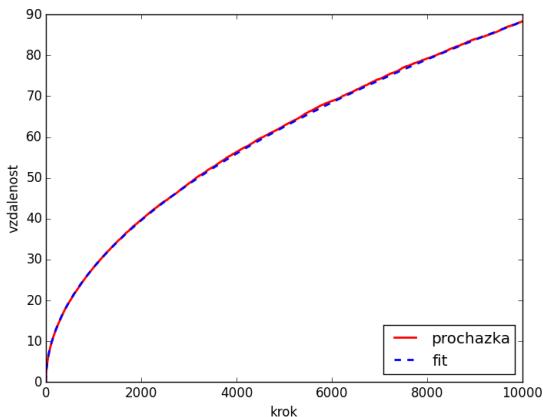
Na souboru  $N = 10^7$  procházek jsme dostali pravděpodobnost 0.6562 skončení procházky v bariéře  $b_1$ , tedy o něco méně než  $2/3$ , přičemž první tři platné číslice jsou s vysokou pravděpodobností přesné. Pro obyčejnou 2D procházku bychom dostali pravděpodobnost přesně  $2/3$  (ne simulací, výpočtem), protože pohyb ve směru osy  $x$  můžeme zcela vypustit a jedná se pak o 1D problém, který jsme již prozkoumali v prvním dílu seriálu. V případě procházky bez návratu neexistuje dobrá 1D analogie, taková procházka vždy probíhá jedním směrem a pravděpodobnost nárazu do jedné bariéry je stejná jako do druhé. Pokud bychom kromě zadané úlohy chtěli také nalézt průměrnou délku procházky, která se zastaví na nějaké bariéře, dostali bychom výsledek 57.27. U jednoduché procházky bychom dostali výrazně větší hodnotu než u procházky bez návratu, ta totiž zamezuje stavům, kdy se trajektorie „točí“ kolem jednoho místa a délka narůstá, aniž bychom se přiblížili k bariéře.

- c) K simulaci opět využijeme kód ze seriálu, nyní však budeme generovat náhodné reálné číslo  $\vartheta$  v rozsahu od 0 do  $2\pi$  a polohu v čase  $t + \Delta t$  určíme podle vztahů

$$\begin{aligned}x(t + \Delta t) &= x(t) + \Delta l \cos \vartheta, \\y(t + \Delta t) &= y(t) + \Delta l \sin \vartheta.\end{aligned}$$

Pro jednoduchost volíme<sup>17</sup>  $\Delta t = 1$ ,  $\Delta l = 1$ . Pro srovnání s pravoúhlou procházkou uvedenou v seriálu budeme generovat 10 000 procházek o délce 10 000. Graf střední vzdálenosti v závislosti na čase (na počtu kroků) je na obrázku 66. Zlogaritmujeme data a fitováním posledních 1 000 bodů lineární funkcí  $f(x) = ax + b$  dostaneme hodnotu směrnice  $a = 0,498 \pm 0,001$ . Vzdálenost procházky od počátku je tedy opět úměrná druhé odmocnině počtu kroků. Stejný výsledek dostaneme pro každou 2D procházku, která má umožněn pohyb symetricky ve všech osách.

<sup>17</sup>I kdybychom měli zadánu např. střední rychlost částice, stále bychom volili tyto kroky, protože simulace je vhodné provádět v normalizovaných jednotkách. Obzvláště v kvantové fyzice by volba  $\hbar \doteq 1,05 \cdot 10^{-34}$  J-s namísto  $\hbar = 1$  způsobila spoustu zbytečných numerických problémů.



Obr. 66: Vzdálenost 2D procházky od počátku v závislosti na počtu kroků (čas). Graf vznikl průměrováním 10 000 procházek o délce 10 000 kroků. Mocinný fit byl proveden na posledních 1 000 bodech. Fit obecně nemusí dobře sedět i na prvních 9 000 bodech, ale může, jak je vidět a tomto příkladu.

Pro srovnání uvádíme i trajektorie procházek o délce 10 000 kroků a 200 000 kroků, obrázky 67 a 68. Zatímco v prvním případě odpovídá „všesměrová“ procházka lépe naší představě o pohybu Brownovy částice než pravoúhlá procházka, při velkém počtu kroků už se rozdíl mezi diskretní a spojitou volbou směru stírají.

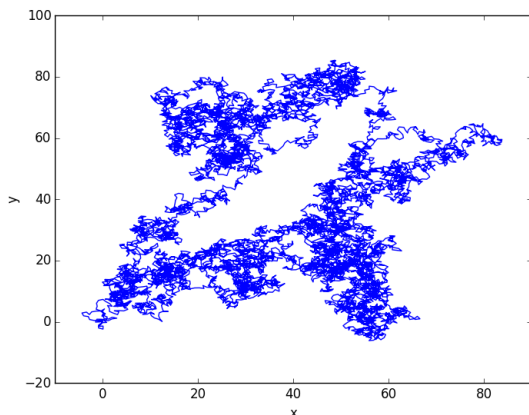
- d) Pro výpočet hodnot chybové funkce použijeme složené Simpsonovo pravidlo, neboť jde o neefektivnější nám známou<sup>18</sup> metodu výpočtu jednorozměrných integrálů. Funkci přitom budeme tabelovat pro hodnoty  $x \in \langle -5, 5 \rangle$ , protože chybová funkce se chová zajímavě v okolí nuly, jinde je skoro konstantní. Integrál byl samplován  $N = 1000$  body. Při této volbě je chyba řádu  $10^{-15}$ , což je skoro strojová přesnost. Tato hodnota  $N$  je tedy rozumným kompromisem mezi dobou výpočtu a přesností. Samotnou funkci jsme tabelovali v  $k = 10000$  bodech, důvodem je vhodný krok při opětovné derivaci. Tabelovaná chybová funkce je vynesena v grafu 69.

Pokud chybovou funkci zderivujeme, analyticky dostaneme

$$(\operatorname{erf}(x))' = \frac{2}{\sqrt{\pi}} \frac{d \left( \int_0^x e^{-t^2} dt \right)}{dx} = \frac{2}{\sqrt{\pi}} e^{-x^2}.$$

Pro numerickou derivaci použijeme metodu 4. řádu, která byla odvozena v minulém díle seriálu. Optimální krok této metody je řádu  $10^{-4}$ , což skoro odpovídá

<sup>18</sup>Samozřejmě existují daleko efektivnější metody, jejich implementace v Pythonu najdete např. v balíku `scipy.integrate`.



Obr. 67: Trajektorie 2D procházky o délce 10 000 kroků. V tomto měřítku je stále vidět, že se nejedná o spojitý proces, jednotlivé diskrétní kroky jsou ještě patrné.

rozestupu dvou hodnot naší tabelované funkce. Provedením derivace skutečně dostaneme kýžený výsledek, jak je vidět v grafu 70.

- e) Maxwellovo-Boltzmannovo rozdělení, popisující rozdělení rychlostí částic ideálního plynu, má tvar<sup>19</sup>

$$f(v) = \begin{cases} \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} 4\pi v^2 e^{-\frac{mv^2}{2kT}} & , v \geq 0 \\ 0 & , v < 0 \end{cases},$$

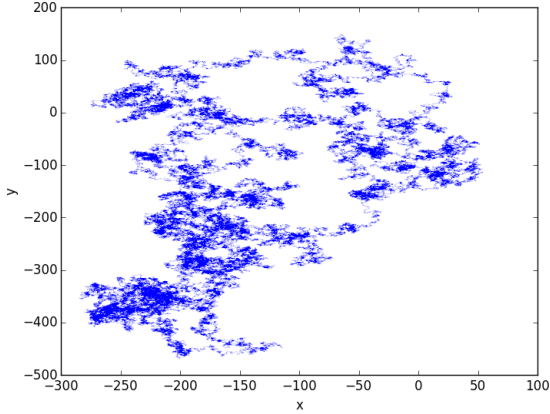
kde  $v$  je velikost rychlosti částice,  $m$  její hmotnost,  $k$  Boltzmannova konstanta a  $T$  termodynamická teplota. Střední hodnota rychlosti je pak z definice rovna

$$\begin{aligned} \langle v \rangle &= \int_0^{\infty} v f(v) dv = \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} 4\pi \int_0^{\infty} v^3 e^{-\frac{mv^2}{2kT}} dv \quad \left/ t = \frac{mv^2}{2kT}, dt = \frac{mv}{kT} dv \right. \\ &= \sqrt{\frac{8kT}{\pi m}} \int_0^{\infty} t e^{-t} dt \quad \text{/per partes} \\ &= \sqrt{\frac{8kT}{\pi m}} \left( [-te^{-t}]_0^{\infty} + \int_0^{\infty} e^{-t} dt \right) = \sqrt{\frac{8kT}{\pi m}}. \end{aligned}$$

Ti z vás, kteří neumějí integrovat, mohou použít program Wolfram Alpha<sup>20</sup>, najít výsledný vzorec v literatuře, či v tabulkách rovnou najít číselnou střední hodnotu rychlosti pro zvolené parametry.

<sup>19</sup>[https://en.wikipedia.org/wiki/Maxwell-Boltzmann\\_distribution](https://en.wikipedia.org/wiki/Maxwell-Boltzmann_distribution)

<sup>20</sup><http://www.wolframalpha.com>



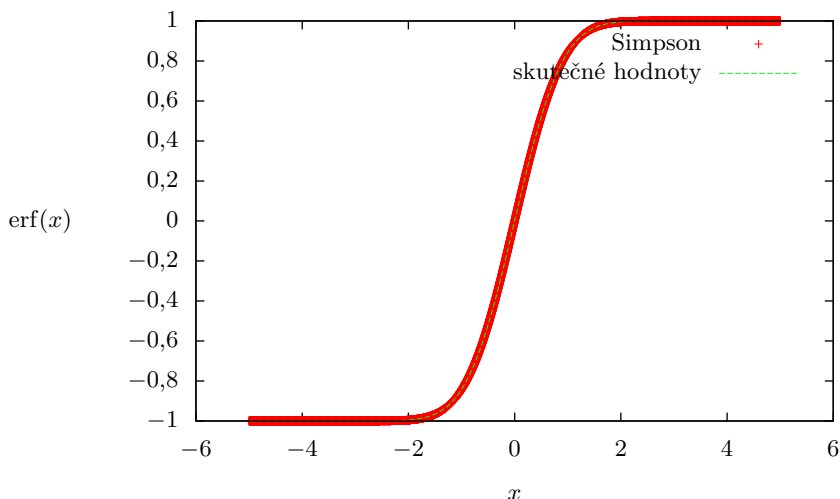
Obr. 68: Trajektorie 2D procházky o délce 200 000 kroků. V tomto měřítku již není poznat, zda byl proces v čase a prostoru diskrétní či spojité.

Nyní, když máme připraveny podklady pro ověření výsledku, se zabýváme samotným numerickým výpočtem. Střední hodnotu rychlosti budeme určovat pro dvouatomový plyný vodík ( $m = 2m_p$ , kde  $m_p \doteq 1,67 \cdot 10^{-27}$  kg je hmotnost protonu) při teplotě  $T = 280$  K, což je rozumná fyzikální volba. Samozřejmě ideálním plynem lze modelovat mnoho jiných plynů za různých teplot, nejde tedy o jedinou možnou volbu. Pokud si vykreslíme Maxwellovo-Boltzmannovo rozdělení pro tyto hodnoty parametrů, zjistíme, že jej tvoří převážně jeden „kopeček“ s maximem kolem hodnoty  $v \approx 1\,500 \text{ m}\cdot\text{s}^{-1}$  a šířkou řádově  $2\,000 \text{ m}\cdot\text{s}^{-1}$  (bráno v polovině výšky „kopečku“, tedy jako tzv. FWHM). Tyto informace nám poslouží pro správné nastavení Metropolisova-Hastingsova algoritmu. Počáteční hodnotu procházky zvolíme do okolí maxima,<sup>21</sup> jako generátor návrhu přeskočení (v textu seriálu označen jako funkce  $g$ ; můžete si ověřit, že skutečně má všechny požadované vlastnosti) si zvolíme rovnoměrné rozdělení se středem v aktuální hodnotě a šířkou  $1\,000 \text{ m}\cdot\text{s}^{-1}$ , tedy polovinou šířky „kopečku“.

Získali jsme posloupnost  $N$  náhodných čísel  $v_i$ , s kterou nyní již můžeme přímočaře použít Monte Carlo integraci. Nesmíme ale zapomenout na to, že generujeme náhodná čísla ne dle rovnoměrného, ale dle Maxwellova-Boltzmannova rozdělení, musíme tedy použít variantu MC integrace s vahami  $w_i = 1/f(v_i)$ , jak bylo uvedeno v textu seriálu. Protože integrujeme funkci  $vf(v)$ , můžeme ukázat, že pro hodnotu integrálu bude platit vztah

$$I = \frac{1}{N} \sum_{i=1}^N w_i v_i f(v_i) = \frac{1}{N} \sum_{i=1}^N \frac{v_i f(v_i)}{f(v_i)} = \frac{1}{N} \sum_{i=1}^N v_i,$$

<sup>21</sup>Není nutno zcela přesně, hodnota by ale měla ležet v blízkém okolí „kopečku“.



Obr. 69: Porovnání námi vypočtené a skutečné chybové funkce.

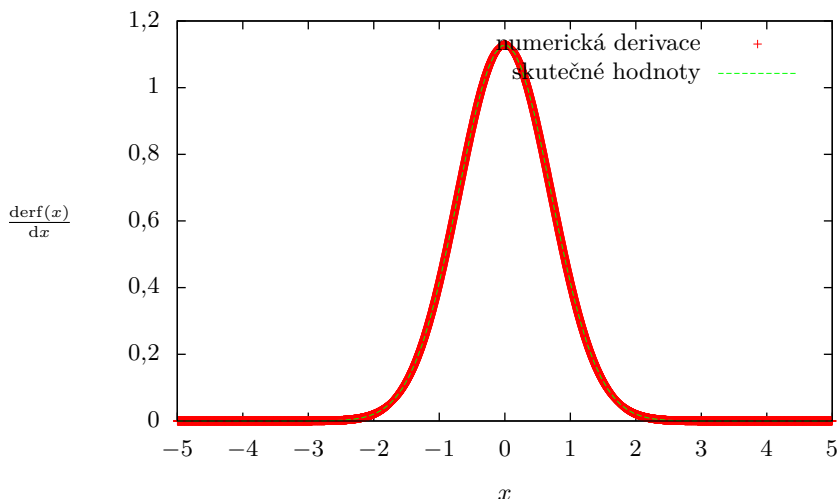
kde  $v_i$  jsou náhodná čísla generovaná Maxwellovým-Boltzmannovým rozdělením  $f(v)$ . Na tento finální vztah vlastně můžeme nahlédnout i tak, že sčítáme  $N$  náhodných proměnných se střednou hodnotou  $\langle v \rangle$ .

Pozastavme se ještě nad jednou věcí. Počítáme určitý integrál v mezích 0 a  $+\infty$ , tyto meze ale v MC integraci nikde nespecifikujeme. Kde tedy dáme programu najevo, že chceme použít právě tyto meze? Vzpomeňme si na MC integraci s rovnoměrným rozdělením náhodných čísel. Tam jsme meze specifikovali šířkou rovnoměrného rozdělení. Zde je to stejné, meze, resp. oblast integrace je určena nosičem<sup>22</sup> hustoty pravděpodobnosti, kterou používáme pro generování náhodných čísel. V našem případě je Maxwellovo-Boltzmannovo rozdělení nenulové v intervalu  $(0, +\infty)$ , vše je tedy v pořádku.

Integrací pro námi zvolené parametry získáme, že pro hodnotu střední rychlosti platí  $\langle v \rangle \doteq (1\,714 \pm 1) \text{ m} \cdot \text{s}^{-1}$ , což odpovídá teoretické hodnotě  $\langle v \rangle \doteq 1\,715 \text{ m} \cdot \text{s}^{-1}$ . Integraci jsme opakovali stokrát, z těchto hodnot jsme pak vypočetli průměr a směrodatnou odchylku, což bylo použito jako zmíněná výsledná hodnota a její chyba. V každém běhu byla integrovaná funkce samplována ve 100 000 bodech. Zdrojový kód použitého programu naleznete jako přílohu řešení na našich webových stránkách.

<sup>22</sup>Nosičem funkce  $f$  tady myslíme množinu  $x$  takových, že  $f(x) \neq 0$ . Přesněji je definován jako tzv. uzávěr této množiny.





Obr. 70: Porovnání námi vypočtené a skutečné derivace chybové funkce.

#### Komentáře k došlým řešením

Ve vašich řešeních problémů zaměřených na markovovské procesy a náhodné procházky (NP) se objevilo několik chyb, které stojí za to si zde rozebrat. Často docházelo k nepochopení pojmu bezpaměťovost. Jestliže každý prvek náhodného řetězce/procesu závisí nejvýše na předchozím prvku, pak je řetězec bezpaměťový neboli Markovův. Pokud při konstrukci trajektorie NP rozhodneme v čase  $t_i$  o následující poloze (čas  $t_{i+1}$ ) na základě předchozí polohy (čas  $t_{i-1}$ ), nejedná se bezpaměťový proces, protože stav v  $t_{i+1}$  je dán stavem v  $t_{i-1}$ , o dva kroky zpět. To se přesně děje u NP bez návratu, proto není markovovská. Někteří z vás ve svém řešení definovali jako stav NP bez návratu směr kroku, pak ji lze považovat za markovovskou, ale z matematického hlediska už jde o jiný proces, při kterém neumíme bezpaměťově určit aktuální polohu. Dále mnoho z vás předpokládalo, že když v případě obyčejné NP pro výpočet narázu do bariéry můžeme zanedbat pohyb v ose  $x$ , lze to udělat i v případě NP bez návratu. To nelze, jednak musíme pohyb v  $x$  alespoň povolit, byť ho nebudeme zaznamenávat, a navíc se i ve 2D projeví tendence NP bez návratu „pokračovat ve směru, kterým vyrazila“.

Dále je při studiu vzdálenosti Brownovy částice od počátečního bodu potřeba zprůměrovat velké množství NP, abychom zjistili, jak se typicky (statisticky) vzdálenost vyvíjí po delším čase. Z jedné realizace procházky žádnou statistiku nevyčteme. Několik řešitelů se také dopustilo chyby, kdy místo počítání aktuální vzdálenosti sčítalo vzdálenosti ze všech předchozích kroků, což je samozřejmě jiná veličina, než nás zajímá. Nakonec je potřeba uvědomit si, že předpokládaná odmocnná závislost vzdálenosti na čase je asymptotická, chování okolo počátku je

jiné. Po delším čase se vzdálenost každé realizace od počátku mezi sebou výrazně liší a je potřeba velký statistický soubor, jinak bude každý bod na průměrovaném grafu zatížen velkou chybou a i směrnice fitu bude nepřesná. (Použité funkce pro fitování v Pythonu neuvážují chyby jednotlivých bodů, proto vám přesná hodnota 0,5 nemusela padnout ani do několikerého násobku směrodatné odchylky fitu.)

## Úloha IV.S . . . kořeni a automati

- a) Nalezňte všechny (tři) reálné kořeny funkce  $\exp(x) - 5x^2$ . Výběr metody je na vás. Nezapomeňte okomentovat, jak a proč jste zvolili daný postup.
- b) Newtonova metoda tak, jak jsme si ji představili funguje i pro funkce komplexní proměnné. Vaším úkolem je vykreslit tzv. Newtonovy fraktály, tedy oblasti v komplexní rovině takové, že když v nich zvolíme počáteční odhad kořenu pro Newtonovu metodu, tak dokonvergujeme k určitému kořenu. Fraktál vykreslete pro funkce  $z^3 - 1$  a  $z^6 + z^3 - 1$ , kde  $z$  je komplexní číslo. Derivace těchto funkcí jsou  $3z^2$ , resp.  $6z^5 + 3z^2$ . Pro výpočet a vykreslení můžete použít Pythonní kód přiložený k zadání.

Poznámka Komplexní derivaci, pokud existuje, lze technicky spočítat stejně, jako reálnou derivaci, tedy pro ni platí stejné vzorce pro derivaci součtu, součinu a složené funkce.

Bonus Nalezňte co nejzajímavější nebo nejhezčí Newtonův fraktál.

- c) Simulujte na počítači (nebo napočítejte ručně) elementární buněčný automat s pravidlem 54 na mřížce délky 20 s periodickými podmínkami alespoň na 10 časových kroků (více určitě neuškodí). Na počátku má jedna libovolná buňka hodnotu 1 a zbylé 0. Výsledek zobrazte v časoprostorovém diagramu (jako na obr. 1 z textu seriálu).
- d) Simulujte hrubnutí 1D povrchu pomocí modelu náhodné depozice popsáném v seriálu. Povrch má rozměr  $L = 100$ , na počátku je zcela hladký. Nakreslete graf závislosti hrubosti  $W$  na čase pro alespoň  $10^8$  kroků (jeden krok = jedna nová částice), výsledek diskutujte.

- a) Naším úkolem je nalézt všechny kořeny funkce  $\exp(x) - 5x^2$ , ale všechny metody, se kterými jsme se seznámili, dokáží samy o sobě najít vždy jen jeden kořen. Prvním krokem tedy bude najít přibližné hodnoty kořenů. Na každý z nich poté zvlášť spustíme některý z algoritmů pro upřesnění hodnoty. Konkrétně použijeme metodu regula-falsi, protože je zpravidla rychlejší, než bisekce. Metodu sečen ani Newtonovu metodu nepoužijeme, neboť obecně nemáme jistotu, že dokonvergujeme, a pokud ano, že dokonvergujeme ke kořenu, který je nejbližší od počátečního odhadu (viz Newtonovy fraktály).

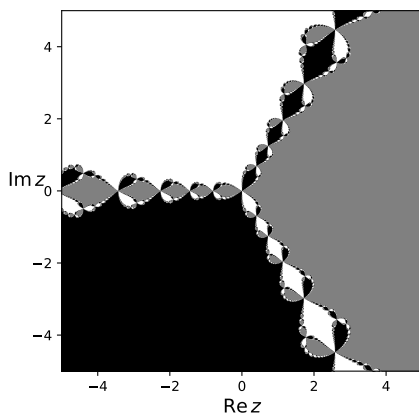
Těchto nevýhod metody sečen a tečen se lze zbavit tak, že je zkombinujeme třeba s metodou bisekce. Základní myšlenka těchto metod je taková, že dostaneme počáteční interval, odněkud z něj vystartujeme Newtonovou metodou a hlídáme si, aby odhad kořene nevylezl z intervalu. Pokud z něj vyleze, přepneme na bisekci.

Abychom mohli použít metodu regula-falsi, musíme nejprve najít tři intervaly takové, aby v každém z nich ležel právě jeden kořen. Nejjednodušeji toho docílíme vykreslením grafu funkce. Takto „od oka“ zjistíme, že kořeny leží přibližně v bodech  $-0,4$ ,  $0,6$  a  $4,7$ . Jako počáteční intervaly nám tedy poslouží například  $(-1, 0)$ ,  $(0, 1)$  a  $(4, 5)$  (první dva intervaly můžeme uhadnout i podle znamének hodnot funkce v bodech  $x = -1$ ,  $x = 0$  a  $x = 1$ , třetí kořen pak musí ležet v  $(1, \infty)$ , protože v nekonečnu je exponenciála větší než kvadratická

funkce). Přímočarou aplikací metody regula-falsi pak dostaneme kořeny přibližně  $-0,37142$ ,  $0,60527$  a  $4,70794$ . Zdrojový kód řešení naleznete v příloze na webu.

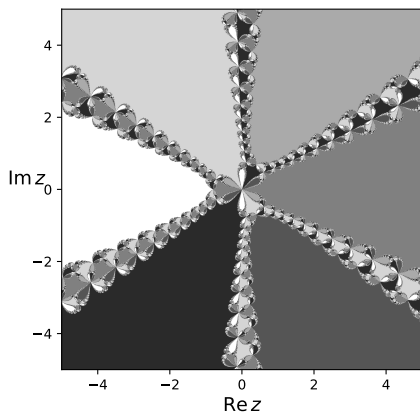
- b) Dosazením funkcí ze zadání do šablony přímočaře získáme Newtonovy fraktály na obrázcích 71 a 72. Barva přitom označuje kořen, do kterého dokonvergujeme z daného počátečního místa. Je vidět, že tyto oblasti tvoří poměrně komplikovaný geometrický útvar – fraktál. Nebudeme zde uvádět složitou matematickou definici fraktálu založenou na nerovnosti různých definic dimenze, všimněme si ale jedné časté vlastnosti fraktálů, soběpodobnosti. Pokud se totiž podíváme na námi vykreslené fraktály, uvidíme, že určité výřezy vypadají stejně jako celý obrázek, jen jsou zmenšeny. Pokud bychom si tyto oblasti přiblížili a vykreslili s dostatečným rozlišením, zjistili bychom, že v zoomování můžeme pokračovat dále, prakticky do nekonečna, a vzniklý výřez by byl stále podobný celému fraktálu.

Prakticky tedy vidíme, že při použití Newtonovy metody si musíme dát pozor na to, že nemusíme vždy dokonvergovat k nejbližšímu kořenu a že v určitých oblastech stačí nepatrně změnit počáteční odhad a dokonvergujeme k jinému kořenu, než předtím. A v neposlední řadě, fraktály jsou prostě hezké a fascinující.



Obr. 71: Newtonův fraktál pro funkci  $z^3 - 1$ .

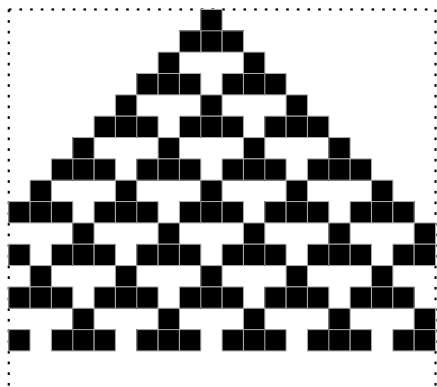
- c) Připomeňme si, jak se elementární buněčné automaty klasifikují. Budeme-li možné stavy buňky označovat 1 a 0, musí okolí jedné buňky (včetně ní) vypadat jako jedna z následujících trojic: 111, 110, 101, 100, 011, 010, 001, 000. Pro každou z těchto osmic (v tomto pořadí) si zapíšeme, do jakého stavu se vyvine prostřední buňka a tato série nul a jedniček nám dá binární číslo, které představuje hledané pravidlo. V této úloze máme pravidlo zadané, 54 se zapíše ve dvojkové soustavě jako 00110110 (úmyslně nevynecháváme nuly na začát-



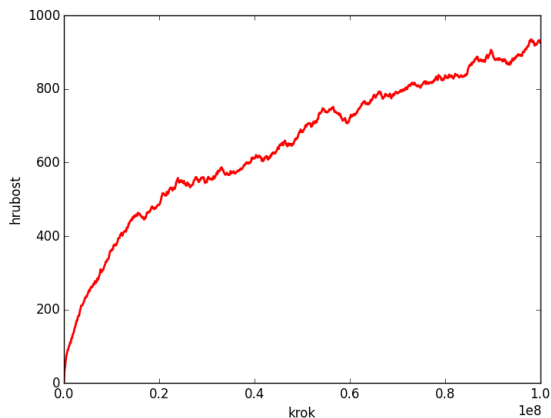
Obr. 72: Newtonův fraktál pro funkci  $z^6 + z^3 - 1$ .

ku). Nyní tedy víme, že pokud je buňka ve stavu 1 a zleva i zprava sousedí s buňkami také ve stavu 1, změní svůj stav na 0 atd. Když se na pravidlo dobře podíváme, vidíme, že se dá popsat i slovně – buňka změní barvu vždy, když sousedí s černou.

Když si zkusíme nakreslit prvních pár kroků na čtverečkový papír, zjistíme, že se jedná o velice jednoduché pravidlo, kde se po dvou krocích pokaždé zreplicuje tvar připomínající tetrisový dílek. Jediný potenciální problém představují okrajové podmínky, ale jak vidíme na obrázku 73, kromě rozseknutí okrajových dílků se nic nezmění.



Obr. 73: Vývoj elementárního CA s pravidlem 54 na mřížce délky 20 s periodickými podmínkami.



Obr. 74: Vývoj hrubosti na povrchu délky  $L = 100$  s náhodnou depozicí po dobu  $10^8$  kroků.

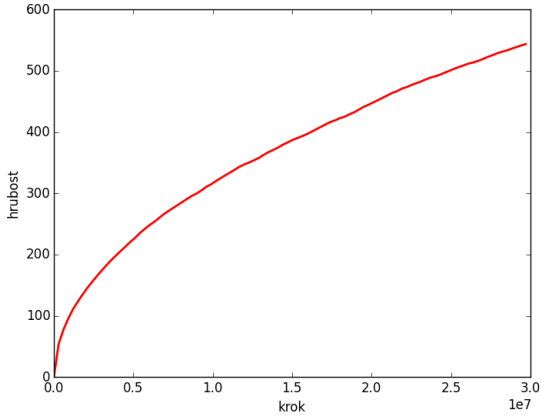
Obrázek 73 byl nakreslen v grafickém editoru, pokud bychom chtěli přeci jen získat výsledek pomocí počítače, museli bychom použít kód, který využívá některé pokročilejší funkce jazyka Python. Na našem webu naleznete příklad takového kódu: <http://fykos.cz/rocnik31/ulohy/serie4>.

- d) Tento úkol již nelze vyřešit pomocí tužky a papíru, ale nebude potřeba konstruovat dvourozměrné pole. Sice opět studujeme vývoj 1D automatu, avšak nemusíme si pamatovat stav každé buňky v každém časovém kroku, ale pouze jednu hodnotu – hrubost. Také je dobré si uvědomit, že na grafu nezobrazíme všech zadaných  $10^8$  kroků, stačí tedy hrubost spočítat vždy jen po intervalu délky  $10^4$  nebo větším.

Výsledek simulace vidíte na grafu 74. Všimněte si, že zatímco na začátku roste hrubost strmě, postupně se rychlost růstu snižuje a zvyšuje se šum. Kód použitý k simulaci náhodné depozice je velmi jednoduchý, naleznete ho taktéž na našem webu.

Podívejme se ještě, jaké je statistické chování vývoje hrubosti – data získáme mírnou úpravou kódu, kdy cyklus uzavřeme do dalšího cyklu a při každém opakování přičteme výsledné pole hrubostí k nově definovanému poli, jehož každý prvek po proběhnutí vnějšího cyklu vydělíme počtem opakování. Po delší simulaci (100 realizací pro  $3 \cdot 10^7$  kroků, může trvat i několik hodin) zjistíme, že charakter vývoje hrubosti je mocninný, konkrétně jde o druhou odmocninu. Stejný výsledek jsme dostali minule při zkoumání střední vzdálenosti 2D náhodné procházky od počátku. Jelikož k této úloze nebyl poskytnut kód, nevyžadujeme fitování, lze si však zkopírováním kódu z minulého dílu<sup>23</sup> ověřit, že

<sup>23</sup>[http://fykos.cz/\\_media/rocnik31/ulohy/pdf/serial31\\_3.pdf](http://fykos.cz/_media/rocnik31/ulohy/pdf/serial31_3.pdf), str. 7.



Obr. 75: Vývoj hrubosti na povrchu délky  $L = 100$  po dobu  $3 \cdot 10^7$  kroků, zprůměrováno přes 100 realizací.

se střední hrubost v asymptotě velkých  $t$  chová podle vztahu  $(t/L)^{1/2}$ , kde  $t$  je počet kroků. Důležitým poznatkem je, že střední hrubost neustále narůstá, což pro růstové modely obecně neplatí.

## Úloha V.S ... rostou nám diferenciální rovnice

- Řešte problém dvou těles pomocí Verletovy a Rungovy-Kuttovy metody 4. řádu přes několik (mnoho) period. Krok přitom volte tak velký, aby se projevily numerické chyby, a pozorujte, jakým způsobem se chyby v obou případech projevují na tvaru trajektorie.
- Řešte pohyb tlumeného lineárního harmonického oscilátoru, který je dán rovnicí  $\ddot{x} + 2\delta\omega\dot{x} + \omega^2x = 0$ , kde  $\omega$  je úhlová frekvence a  $\delta$  tlumící člen. Parametry měňte a sledujte změny v chování oscilátoru. Pro jaké hodnoty parametrů se oscilátor utlumí nejrychleji?
- Modelujte růst povrchu metodou balistické depozice a studujte statistické chování hrubosti povrchu. Nalezněte mocniny  $\alpha$  a  $\beta$  popisující růst před saturací a po saturaci (viz seriál). Vyjděte z kódu v seriálu. Volte takový počet kroků, abyste byli schopni dobře studovat oba režimy hrubnutí. Lineární rozměr povrchu volte alespoň  $L = 256$ . (Upozornění: simulace mohou trvat i několik hodin.)
- Simulujte na čtvercové mřížce šíření zhoubného nádoru pomocí Edenova modelu. Uvažujte přitom následující obměnu: s pravděpodobností  $p_1$  dojde k nákaze zdravé buňky v kontaktu s nádorovou a s pravděpodobností  $p_2$  dojde k uzdravení nakažené. Volte nejprve  $p_1 \gg p_2$ , pak  $p_1 > p_2$  a nakonec  $p_1 < p_2$ . Na

počátku necht' je nakaženo pět buněk do tvaru kříže. Kvalitativně popište, co pozorujete.

- e) Přepište kód ze seriálu pro růst fraktálního krystalu (DLA model) na hexagonální mřížce na růst na čtvercové mřížce a spočtete dimenzi výsledného fraktálu. Poznámka Využití kódy přiložené k seriálu není nutné, ale doporučeno.

- a) Nejprve musíme odvodit správné pohybové rovnice, přičemž pro jednoduchost budeme celou dobu pracovat v kartézských souřadnicích. Gravitační síla, kterou první bod o hmotnosti  $m_1$  působí na druhý hmotný bod o hmotnosti  $m_2$ , je

$$\mathbf{F}_g = -\frac{Gm_1m_2}{r_{12}^3}\mathbf{r}_{12},$$

kde  $\mathbf{r}_{12}$  je polohový vektor druhého bodu vůči prvnímu bodu a  $r_{12}$  je jeho velikost. Silou o stejné velikosti, ale s opačným znaménkem, pak působí bod 2 na bod 1. Zrychlení bodu pak obdržíme vydělením síly hmotností daného bodu. Protože druhý Newtonův zákon má (ve verzi s konstantní hmotností) tvar  $\ddot{\mathbf{x}} = \mathbf{a}$ , máme sestaveny pohybové rovnice, jen je potřeba je upravit do tvaru vhodného pro danou metodu. U Verletovy metody máme splněno, neboť je očekávána diferenciální rovnice právě v tomto tvaru. Pro Rungovu-Kuttovu metodu musíme rovnici upravit na soustavu rovnic prvního řádu. To docílíme substitucí  $\dot{\mathbf{x}} = \mathbf{v}$ , díky čemuž automaticky obdržíme i okamžité rychlosti bodů, které budeme potřebovat při výpočtu energie. Pro Verletovu metodu musíme rychlosti získat oklikou, například aplikací numerické derivace dle času na funkci polohy.<sup>24</sup> Výsledná soustava rovnic tedy je

$$\begin{aligned}\ddot{x}_1 &= \frac{Gm_2(x_2 - x_1)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, & \ddot{y}_1 &= \frac{Gm_2(y_2 - y_1)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, \\ \ddot{x}_2 &= \frac{Gm_1(x_1 - x_2)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, & \ddot{y}_2 &= \frac{Gm_1(y_1 - y_2)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}\end{aligned}$$

pro Verletovu metodu a

$$\begin{aligned}\dot{x}_1 &= \frac{Gm_2(x_2 - x_1)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, & \dot{y}_1 &= \frac{Gm_2(y_2 - y_1)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, \\ \dot{x}_2 &= \frac{Gm_1(x_1 - x_2)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, & \dot{y}_2 &= \frac{Gm_1(y_1 - y_2)}{|(x_2 - x_1)^2 + (y_2 - y_1)^2|^{\frac{3}{2}}}, \\ \dot{x}_1 &= v_{x1}, & \dot{y}_1 &= v_{y1}, \\ \dot{x}_2 &= v_{x2}, & \dot{y}_2 &= v_{y2}\end{aligned}$$

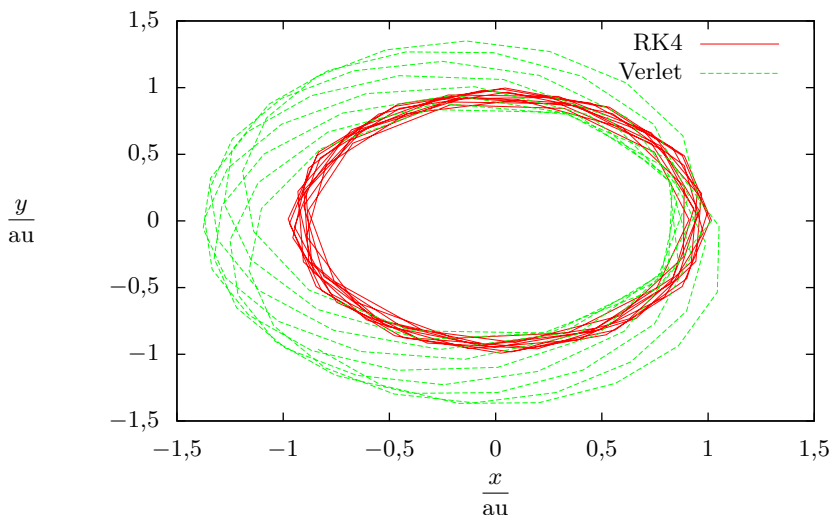
pro Rungovu-Kuttovu metodu. Tyto rovnice platí v inerciální soustavě spojené s těžištěm. Bylo by samozřejmě možné simulaci provádět i v jiné soustavě, bylo

<sup>24</sup>V případě, že potřebujeme znát rychlosti, lze také použít metody, které jsou Verletovi ekvivalentní, jako leap-frog, či rychlostní Verlet. O těchto a dalších metodách jsme se v seriálu nezmiňovali, můžete si je ale dohledat.



by ale potřeba příslušně transformovat polohy a rychlosti a v případě neinerciálních soustav započítat vliv setrvačných sil. Příkladem vhodné transformace je umístění jednoho z těles do počátku (nerotující) souřadné soustavy, pak totiž stačí simulovat pohyb pouze jednoho tělesa vůči druhému, což ve výsledku vede k menším numerickým chybám.

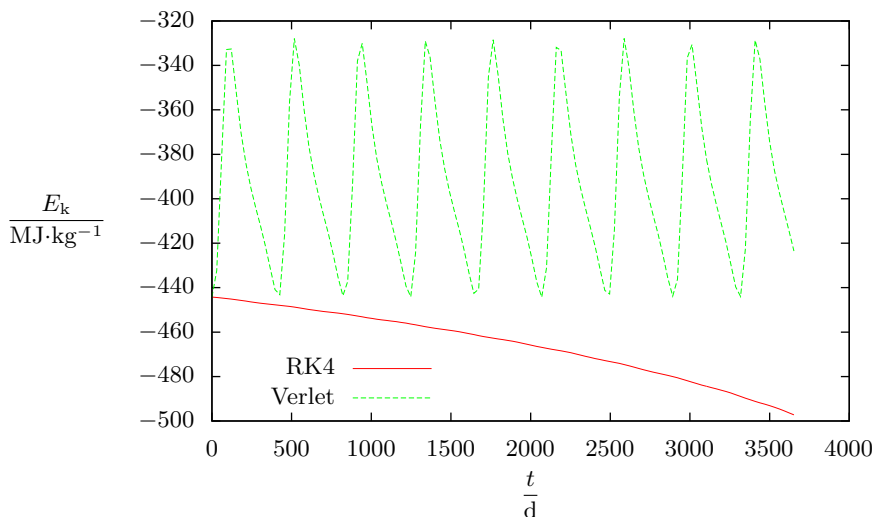
Pro porovnání vlastností metod si nyní problém zjednodušíme a budeme uvažovat, že platí  $m_1 \gg m_2$ , tedy například pohyb planety okolo Slunce. Slunce pak můžeme napevno posadit do středu a simulovat pouze pohyb planety.



Obr. 76: Simulace oběhu Země kolem Slunce pomocí Verletovy a Rungovy-Kuttovy metody 4. řádu s krokem 30 dní.

V grafu 76 vidíme výsledek takové simulace po dobu 10 oběhů s poměrně velkým krokem. Na první pohled je vidět nesrovnalost ve velikosti velké poloosy mezi oběma metodami. Ta je způsobena tím, že Verletovu algoritmu je nutno dodat počáteční podmínky ve formě dvou poloh místo polohy a rychlosti, jako v případě Rungovy-Kuttovy metody. Pro výpočet tohoto prvního kroku byl (z důvodu lenosti) použit Eulerův algoritmus, který s tímto velkým krokem způsobil značnou chybu. Pokud bychom byli při volbě počátečních podmínek pečlivější, tato chyba by nevznikla, dále ji tedy budeme ignorovat.

Druhá věc, které si všimneme, je stáčení dráhy vypočtené Verletovou metodou, zatímco dráha vypočtená Rungovou-Kuttovou metodou je víceméně stále stejná. To je způsobeno tím, že Verletova metoda je metodou druhého řádu, zatímco použitá Rungova-Kuttova metoda je řádu čtvrtého, pro daný krok má tedy menší chybu. Všimněme si ale, že dráha je touto chybou pouze stáčena, nemění se její rozměry. To je právě důsledkem časové reverzibility a následného



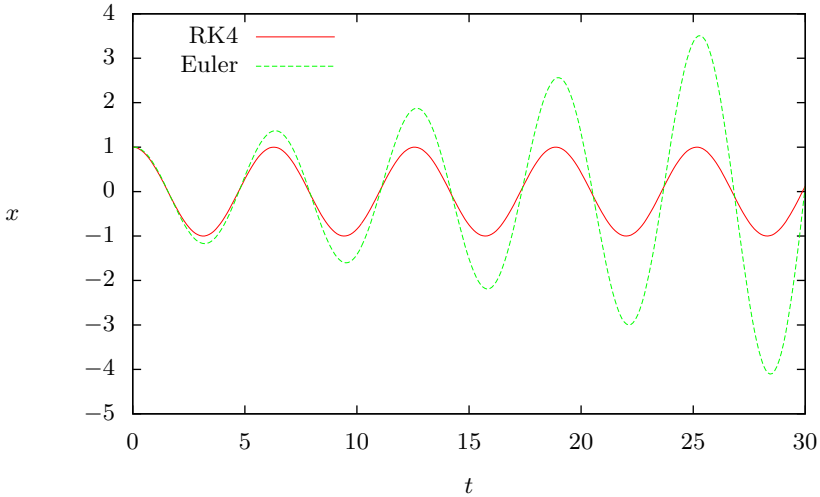
Obr. 77: Simulace oběhu Země kolem Slunce pomocí Verletovy a Rungovy-Kuttovy metody 4. řádu s krokem 30 dní – vývoj celkové energie.

zachování energie u Verletovy metody. Pokud se podíváme na časovou závislost energie planety (vztažené na jednotkovou hmotnost planety) v grafu 76, vidíme, že v případě Verletovy metody energie osciluje s periodou jednoho oběhu. Není tedy vždy na zcela správné hodnotě, v průběhu času ale nikam nedriftuje, i s takto hrubým krokem se tedy bude takto držet, i kdybychom simulovali třeba 10 000 oběhů. Energie v případě Rungovy-Kuttovy metody oproti tomu v tomto úseku simulace drží lépe díky vyššímu řádu metody, postupně se ale mění (a čím dál rychleji), po mnoha periodách by tedy vůbec neodpovídala skutečnosti.

- b) Po substituci  $\dot{x} = v$  dostáváme diferenciální rovnici  $\dot{v} = -2\delta\omega v - \omega^2 x$ , čímž jsme obdrželi systém dvou rovnic prvního řádu. Na jeho řešení použijeme Rungovu-Kuttovu metodu 4. řádu, protože je na rozdíl od Adamsových-Bashforthových-Moultonových metod jednokroková, není ji tedy potřeba startovat, navíc je Rungova-Kuttova metoda 4. řádu dokonce vhodnější z hlediska stability.<sup>25</sup> Naopak explicitní Eulerovu metodu zde použít nemůžeme, neboť pro netlumený oscilátor  $\delta = 0$  je řešení nestabilní pro libovolně malý krok. Tvrzení opět souvisí s oblastí stability – tento pojem se pokusíme objasnit v příštím díle seriálu. Nicméně i bez těchto teoretických znalostí lze z chování řešení od pohledu usuzovat na jeho (ne)stabilitu. Nestabilita totiž znamená, že se chyby neúměrně zvětšují, řešení tedy roste k nekonečnu (či osciluje mezi kladným a záporným

<sup>25</sup>Má větší tzv. oblast stability, můžeme tedy volit o něco větší krok pro dané parametry při zachování stability.

nekonečnem)<sup>26</sup> Jak takové nestabilní chování vypadá, vidíme na obrázku 78, kde je vykresleno nestabilní řešení rovnice  $\ddot{x} + \omega^2 x = 0$  explicitní Eulerovou metodou a pro porovnání stabilní řešení Rungovou-Kuttovou metodou.

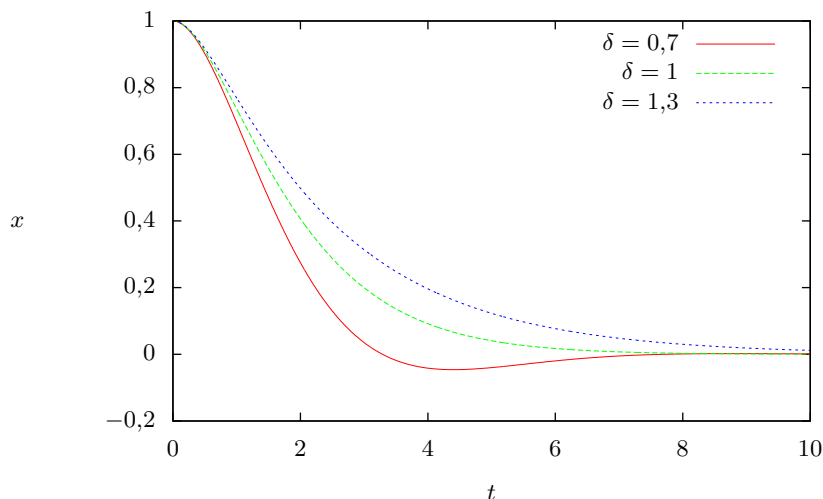


Obr. 78: Porovnání stabilního a nestabilního řešení rovnice  $\ddot{x} + \omega^2 x = 0$  se stejnými počátečními podmínkami, s krokem 0,1 a  $\omega = 1$ .

Kromě počátečních podmínek, které mají jasný význam počáteční rychlosti a polohy, můžeme měnit další dva parametry, úhlovou rychlost  $\omega$  a tlumící člen  $\delta$ . S trochou experimentování zjistíme, že mají význam odpovídající jejich názvu, parametr  $\omega$  ovlivňuje periodu kmitů, zatímco  $\delta$  řídí rychlost exponenciálního útlumu.<sup>27</sup> Jak jsme již zjistili, hodnota  $\delta = 0$  odpovídá kmitům bez tlumení. Čím větší pak faktor je, tím je tlumení silnější. Je ale zajímavé, že silnější tlumení nutně neznamená, že výchylka rychleji klesne k nule, jak můžeme vidět na obrázku 79, kde jsou vyneseny časové závislosti výchylky pro hodnotu parametru  $\delta = 0,7$ ,  $\delta = 1$  a  $\delta = 1,3$ . Je patrné, že nejrychleji se utlumí oscilátor při  $\delta = 1$ . Nejde jen o náš tip na základě experimentování s hodnotou parametru, ale jde o obecný výsledek. Říkáme, že takový oscilátor koná *mezní aperiodický pohyb*. Pokud je  $\delta < 1$ , oscilátor není zcela utlumen během jednoho kmitu, tlumení k nulové výchylce tedy trvá déle. Oscilátor koná *periodický pohyb*. Naopak pokud  $\delta > 1$ , oscilátor je přetlumený tak, že „tlumí i tlumení výchylky k nule“, výchylka tedy klesá pomalu, ale nepřekmitne a u nuly se ustálí, pohyb je *aperiodický*.

<sup>26</sup> Je samozřejmě třeba dát pozor, jestli divergence není očekávaným správným řešením dané rovnice.

<sup>27</sup> Ve skutečnosti  $\delta$  také ovlivňuje periodu kmitů a  $\omega$  rychlost útlumu, jak si rozebereme dále.



Obr. 79: Časová závislost výchylky lineárního harmonického oscilátoru pro tři různé hodnoty parametru  $\delta$  při  $\omega = 1$  a stejných počátečních podmínkách.

Doplňme, že pro tuto diferenciální rovnici lze nalézt analytické řešení ve tvaru

$$x(t) = Ae^{\lambda_1 t} + Be^{\lambda_2 t},$$

kde  $\lambda_{1,2} = -\delta\omega \pm \omega\sqrt{\delta^2 - 1}$  a  $A$  a  $B$  jsou integrační konstanty (odvoditelné z počátečních podmínek). Pro  $\delta \in (0, 1)$  je výraz pod odmocninou záporný, odmocnina má tedy komplexní hodnotu. Pak platí  $\lambda_{1,2} = -\delta\omega \pm i\omega\sqrt{1 - \delta^2}$ , řešení tedy pomocí úpravy přes Eulerův vzorec pro komplexní exponenciálu a triku využívajícího fakt, že součet dvou různých řešení je také řešení, můžeme přepsat do tvaru

$$x(t) = e^{-\delta\omega t} \left( C \cos(t\omega\sqrt{1 - \delta^2}) + D \sin(t\omega\sqrt{1 - \delta^2}) \right), \quad (85)$$

kde  $C$  a  $D$  jsou reálné integrační konstanty. Z tohoto výsledku vidíme, že tlumení působí s faktorem  $\delta\omega$  a oscilace mají úhlovou frekvenci  $\omega\sqrt{1 - \delta^2}$ . Pro mezní případ  $\delta = 1$  pak z rovnice (85) oscilace vymizí a zůstane jen tlumící exponenciála. Zároveň ale<sup>28</sup> bude řešením rovnice i  $x(t) = Kte^{-\omega t}$ , kde  $K$  je konstanta. A protože součet řešení homogenní diferenciální rovnice je také řešením, je celkové řešení pro případ  $\delta = 1$  rovno

$$x(t) = (F + Gt)e^{-\omega t},$$

<sup>28</sup>Protože jde o případ s vícenásobnými kořeny tzv. charakteristického polynomu dané ODR. Detaily teoretického řešení dif. rovnic si můžete dohledat ve vhodné literatuře.

kde  $F$  a  $G$  jsou integrační konstanty. Můžete si derivováním ověřit, že toto řešení klesá k nule monotónně a (pro stejné počáteční podmínky) rychleji než periodické řešení nebo součet dvou exponenciál z aperiodického řešení.

Celou dobu jsme měli představu mechanického oscilátoru, například závaží na pružině. Uvědomme si ale, že stejná rovnice popisuje i jiné, neméně významné systémy, například RLC obvod, kde výchylka odpovídá elektrickému náboji (který je úměrný napětí) na kondenzátoru a rychlost proudu protékajícímu obvodem.

- c) V této úloze je zásadní dodržet limit na minimální velikost mřížky, nebo lépe použít ještě větší rozměry<sup>29</sup>  $L = 512$  či  $L = 1024$ . Lze si snadno rozmyslet, že pro  $L = 2$  a podobně malé mřížky bude balistická depozice vykazovat jiné odlišné chování od velkých mřížek. Chceme-li tedy získat správné hodnoty škálovacích parametrů, musíme volit tak velkou mřížku, aby se neprojevovala významně její konečnost.

Podívejme se nejprve na kritický škálovací koeficient  $\alpha$  a prozradme si dopředu, že exaktní analytické řešení diferenciální rovnice příslušné balistické depozice vede na  $\alpha = 0.5$ . S využitím kódu v seriálu nyní vypočteme vývoj hrubosti pro dvě mřížky  $L_1 = 256$  a  $L_2 = 512$ . Jelikož v asymptotě velkých časů platí  $W(L) \sim L^\alpha$ , srovnáním dvou simulací dostaneme

$$\frac{W(L_2)}{W(L_1)} = \left(\frac{L_2}{L_1}\right)^\alpha,$$

neboli

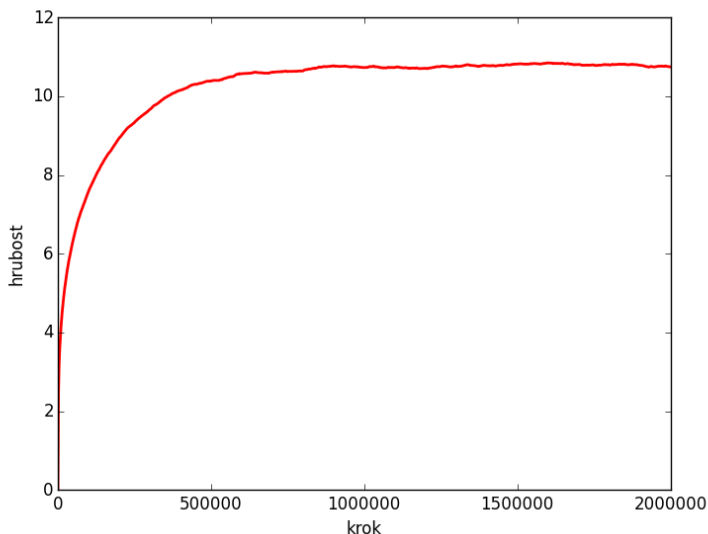
$$\alpha = \log_2 \frac{W(L_2)}{W(L_1)}. \quad (86)$$

Abychom splnili podmínku  $t \gg t_x$ , tj. že se pohybujeme v časech výrazně větších, než je charakteristický čas přechodu od růstu k saturaci, vykreslíme si graf a odhadneme, kdy už je hrubost saturována. Jako příklad uvádíme obr. 80, kde jsme vykreslili pro mřížku  $L_2$  hrubost zprůměrovanou přes  $5 \cdot 10^3$  běhů na  $2 \cdot 10^6$  krocích. Vidíme, že po  $10^6$  krocích je hrubost již bezpečně saturována. Také si můžeme všimnout, že i přes průměrování není graf zcela hladký. Jelikož není v našich časových možnostech provést tolik běhů, budeme za saturovanou hrubost považovat průměr hrubostí na intervalu  $10^6 < t < 2 \cdot 10^6$ , který je v našem případě roven  $\overline{W}(L_2) = 10,76$ . Simulaci provedeme též pro  $L_1$  a podobným zpracováním získáme  $\overline{W}(L_1) = 7,82$ . Ze vztahu (86) pak vypočteme<sup>30</sup>  $\alpha = 0,46$ . To není úplně špatný výsledek, ale pořád se od exaktního výsledku nezanedbatelně odchyľujeme. Pro  $L_3 = 1024$  dostaneme porovnáním s  $L_2$  koeficient  $\alpha = 0,48$ .

Pro nalezení škálovacího parametru  $\beta$  nepotřebujeme porovnávat více mřížek, stačí pouze fitovat evoluce hrubosti v oblasti  $t \ll t_x$  (jak v Pythonu fitovat

<sup>29</sup>Že se jedná o mocniny dvojky není příliš podstatné, leda že bychom chtěli využít způsob ukládání čísel do paměti.

<sup>30</sup>Pravděpodobně jste si všimli, že jsme v seriálu již před delší dobou upustili od počítáním směrodatných odchylek. Je to proto, že nás obvykle více trápí nejistoty způsobené nastavením naší simulace než nejistoty způsobená statistikou.

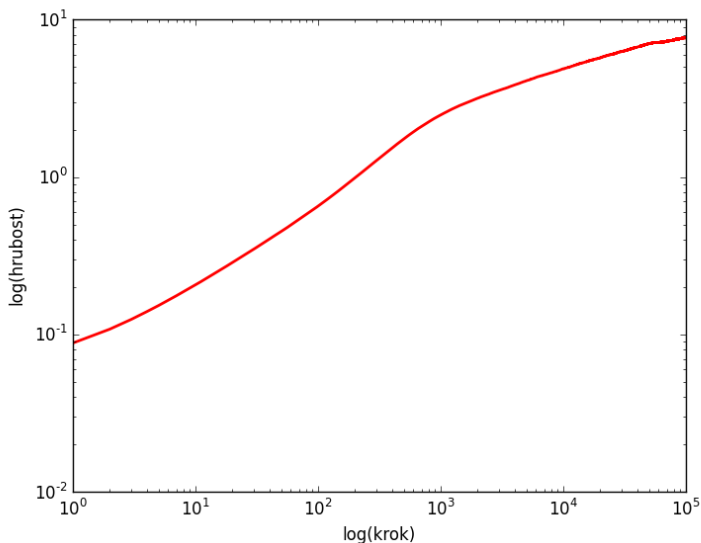


Obr. 80: Vývoj hrubosti povrchu spočtený na základě modelu balistické depozice na mřížce délky  $L = 512$ , průměrováno přes 5 000 běhů.

jsme se učili v třetím díle seriálu, který se věnoval náhodným procházkám). Z lineárního grafu však jen špatně odhadneme, kdy se ve vztahu  $W(t, L) \sim t^\beta$  začíná exponent měnit. Proto si nakreslíme log-log graf (obě osy logaritmické), viz obrázek 81. Krok indexujeme od 1, čímž se taky vyhneme logaritmování nuly. Jelikož děláme simulaci pouze pro  $10^5$  kroků na mřížce  $L_1 = 256$ , mohli jsme si dovolit nefiltrovat body v grafu. Průměrovali jsme přes 1000 běhů.

Vidíme, že hrubost v naší simulaci vykazuje na počátku komplikovanější chování. Během prvních několika desítek kroků vidíme lineární chování a v oblasti několika tisíců až desetitisíců kroků také, ale v obou případech s jiným sklonem. Dále již graf plynule přechází k saturaci (není zde vyobrazeno, ale víme z obr. 80). Fitováním prvního lineárního úseku na intervalu 20–50 kroků získáme koeficient  $\beta_1 \doteq 0,50$ , na druhém úseku získáme na intervalu 8 000–20 000 kroků  $\beta_2 \doteq 0,26$ . Koeficient  $\beta_1$  není ten, který hledáme. Pro takto malé počty kroků je totiž malá pravděpodobnost, že se nová částice trefí do blízkosti nějaké předešlé, a chování je proto téměř nezávislé na modelu.<sup>31</sup> Pro čtyřnásobně velkou mřížku se intervaly linearity posunou v čase doprava,  $\beta_1$  se nezmění, ale dostaneme  $\beta_2 = 0,28$ . Pokud půjdeme ještě dál a provedeme výpočet na mřížce velikosti 4 096, dostaneme koeficient  $\beta_2 = 0,29$ , pro mřížku velikosti 65 536

<sup>31</sup>Podle tzv. narozeninového paradoxu víme, že tohle chování bychom měli pozorovat pro řádově prvních  $\sqrt{L}$  kroků.



Obr. 81: Vývoj hrubosti povrchu spočtený na základě modelu balistické depozice na mřížce délky  $L = 256$ , v logaritmickém grafu. Průměrováno přes 1 000 běhů.

pak  $\beta_2 = 0,32$ . I v tomto případě existuje analytické řešení, které dává  $\beta = 1/3$ . Vidíme, že konvergence je velice pomalá,<sup>32</sup> navíc pro menší mřížky je obtížné najít lineární interval pro fitování koeficientu  $\beta$ .

Všimněme si také chování křivky mezi našimi dvěma lineárními oblastmi. Její sklon nejdříve z 0,5 roste a pak prudce poklesne na  $\approx 1/3$ , přičemž v grafu udělá (na obr. 81 kolem tisíce kroků) kopeček. Toto chování je celkem atypické a nevíme, jak ho teoretický model, ze kterého plyne  $\alpha$  a  $\beta$ , vysvětlí. Nejde pouze o chybu způsobenou malým  $L$  – pro větší  $L$  je sice křivka „rovnější“, ale kopeček mezi dvěma oblastmi linearity je stejně výrazný.

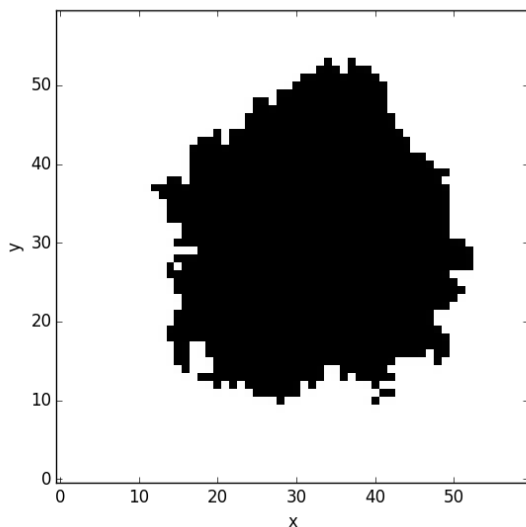
Alternativním numerickým přístupem, kterým taky vyřešíme těžce pozorovatelnou vlnitost křivky, může být výpočet numerické derivace naší log-log křivky a hledání intervalu, ve kterém je tato derivace konstantní. U balistické depozice je ale problém v tom, že numerická derivace výrazně osciluje a musíme hrubost průměrovat přes příliš mnoho běhů (nebo počítat derivaci přes hodně sousedních bodů) na to, aby byla pro výpočet koeficientu  $\beta$  použitelná.

Obecně můžeme říct, že jsme u této úlohy naráželi na problém s výpočetním časem. Pro seriózní simulační výpočty se proto nepoužívají interpretované jazyky jako Python, ale C, C++ či Fortran.

d) Uzdravování nakažených buněk implementujeme do modelu jednoduše: před

<sup>32</sup>Viz např. <http://www.thp.uni-koeln.de/krug/teaching-Dateien/SS2012/Farnudi2011.pdf>

řádky kódu provádějící nakažení jedné ze sousedních buněk necháme vygenerovat náhodné číslo od 0 do 1 a pokud je menší než pravděpodobnost uzdravení, právě vybraná buňka se uzdraví a vybíráme jinou buňku. Také již nebudeme označovat nemocné buňky obklopené nemocnými indexem 2. Navíc předpokládáme  $p_1 + p_2 = 1$ ; nemá smysl uvažovat kroky, kdy nedojde k nakažení žádné buňky, protože v simulaci nesledujeme skutečný čas (tj. s klidem vyřazujeme kroky, kdy se nic neděje).



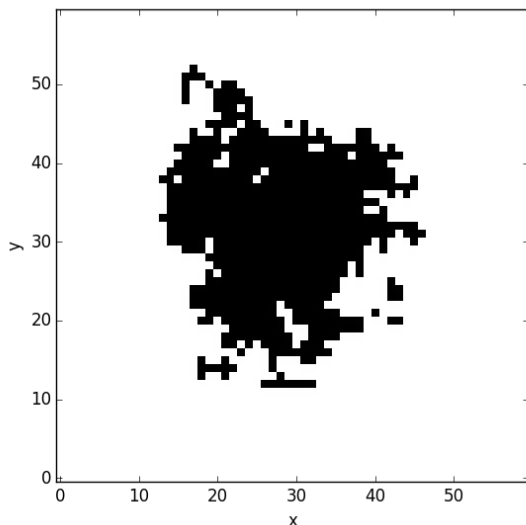
Obr. 82: Stav nakažených buněk po 1500 krocích Edenova modelu s pravděpodobností uzdravení 0,1.

Pokud zvolíme  $p_1 < p_2$ , tedy že pravděpodobnost nakažení je menší než pravděpodobnost uzdravení, dojde v několika málo krocích (čím větší  $p_2$ , tím méně kroků) k uzdravení všech buněk a simulace končí. Pokud zvolíme  $p_1 \gg p_2$ , například  $p_1 = 0,9$ , nebude se výsledek příliš lišit od případu  $p_1 = 1$ , pouze mírně naroste porozita a rozeklanost povrchu, viz obrázek 82. Zajímavý je případ  $p_1 \gtrsim p_2$ . Pro volbu  $p_1 = 0,55$  jsme dostali obrázek 83, přičemž simulace měla pětkrát více kroků než v případě  $p_1 = 0,9$ . Porozita je mnohem vyšší, dokonce vznikají oddělené ostrůvky nakažených buněk. Celý vývoj nákazy si můžete prohlédnout v animaci na webu.<sup>33</sup>

- e) Model pro simulaci DLA obsahuje dva parametry, jejichž hodnoty se výrazně projeví na výsledku. Jsou to počáteční vzdálenost každé difundující částice

<sup>33</sup><http://fykos.cz/rocnik31/ulohy/serie5>



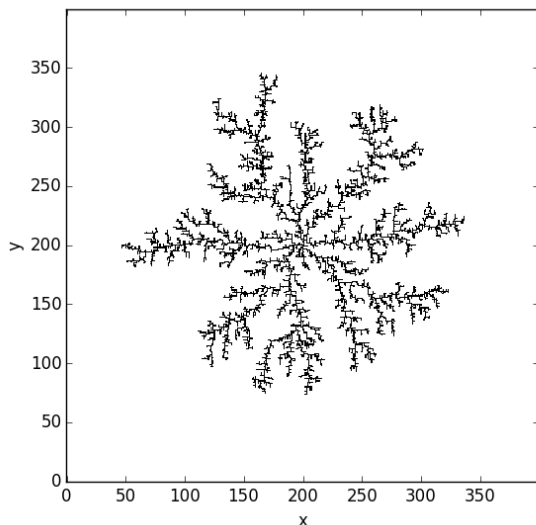


Obr. 83: Stav nakažených buněk po 7 500 krocích Edenova modelu s pravděpodobností uzdravení 0,45.

a velikost oblasti, mimo kterou nesmí částice utéct.<sup>34</sup> Také vyvstává otázka, jak měřit vzdálenost částice od počátku. Na šestiúhelníkové mřížce jsme radiální vzdálenost interpretovali jako „šestiúhelníkovou slupku“ – ta má svým tvarem již poměrně blízko ke kružnici. Měli bychom počáteční vzdálenost na čtvercové mřížce interpretovat jako poloměr kružnice, nebo rozměr čtverce?

Položená otázka nemá jednoznačnou odpověď, oba přístupy mohou přinést zajímavý výsledek. Jelikož je ale cílem naší simulace napodobit přírodní jev růstu krystalů z roztoku, necháme startovat částice na kružnici a podobně jako u úloh s náhodnými procházkami budeme předpokládat, že pro simulace velkých rozměrů přestane hrát roli, že jsme skutečný všesměrový pohyb omezili na mřížku. Co se týče dvou zmíněných parametrů, jedná se o trade-off mezi rychlostí a přesností simulace. Čím dále bude částice startovat a čím méně omezíme její pohyb, tím přesnější výsledek dostaneme – na druhou stranu, náročnost simulace bude prudce růst. Zde jsme zvolili počáteční vzdálenost jako  $1,2r_{\max}$  a šíření jsme zastavili, když částice utekla za  $1,5r_{\max}$ . Zde použitý kód naleznete jako přílohu u řešení na webu, na obrázku 84 je pak výsledný krystal složený z 8 000 částic.

<sup>34</sup>Existují samozřejmě alternativní postupy, například místo omezení pohybu na určitou oblast můžeme zavést limit na délku trajektorie difundující částice.



Obr. 84: Výsledek simulace růstu podle DLA modelu na čtvercové mřížce. Počet částic v krystalu je 8 000.

Na základě vzorečku z seriálu<sup>35</sup>

$$D_{\text{DLA}} = \frac{\log N}{\log r_{\text{max}}}, \quad (87)$$

jsme určili fraktální dimenzi  $D = 1,79$ ; opakováním simulace si můžeme ověřit, že se první tři platné číslice nemění. Řešení diferenciálních rovnic definujících DLA, kde se částice pohybují Brownovým pohybem (tj. nejsou vázány na mřížku), vede na hodnotu dimenze  $D = 1,71$ . Všimněte si, že na čtvercové mříži jsme dostali vyšší hodnotu dimenze než na šestiúhelníkové v seriálu (1,75). Mohli bychom se tedy domnívat, že povolení pohybu do více směrů přibližuje náš odhad výsledku pro skutečnou, na mřížku nevázanou difúzi. Nesmíme však zapomínat, že u obou simulací jsme výrazně omezili prostor, ve kterém se částice smí pohybovat, nemáme tedy dostatečně dobrý podklad pro vyvozování podobných závěrů.

<sup>35</sup>V seriálu bylo napsáno, že zatímco v případě Kochovy vločky měříme velikost strany trojúhelníku, u DLA měříme rozměr krystalu. Nebylo tam však již zmíněno, že tato záměna má být provedena podle jako  $1/\varepsilon \rightarrow r_{\text{max}}$ . Pokud prostě dosadíme místo  $\varepsilon$  hodnotu  $r_{\text{max}}$ , dostaneme výsledek s opačným znaménkem. Proto za řešení uvádějící chybné znaménko nebyly strhávány body. Matematická definice dimenze ovšem může být libovolná, jde pouze o to, jak interpretujeme výsledky – klidně bychom dimenzi mohli zadefinovat tak, aby byla pro fraktál jako Kochova vločka, který neroste, ale „houstne“, záporná.

Ještě uvedme, že existují různé modifikace DLA. Například můžeme zavést pravděpodobnost navázání částice na krystal. Pokud je výrazně menší než 1, může se částice dostat hlouběji do keříčků krystalu. Fraktál pak začíná být geometricky hustý, tzn. jeho dimenze se blíží hodnotě 2 (dimenze prostoru, ve kterém se pohybuje).

## Úloha VI.S . . . matice a populace

- a) Na základě Lotkova-Volterrova modelu simulujte vývoj populace predátora a kořisti (např. sluněčka sedmitečného a mšice makové) pro následující hodnoty parametrů:  $r_m = 0,8$ ,  $D_m = 1,0$ ,  $r_s = 0,75$ ,  $D_s = 1,5$ . Počáteční populace volte po dvojicích jako  $m = 0,5$  a  $s = 2,0$ ;  $m = 1,5$  a  $s = 0,5$ ;  $m = 1,95$  a  $s = 0,75$ . Výsledek zanešte do grafu závislosti populace predátora na populaci kořisti. Výsledky diskutujte.  
 Bonus Nalezněte tvar křivek v grafu pomocí analytických metod (integrací diferenciální rovnice).
- b) Použitím kompetitivního Lotkova-Volterrova modelu simulujte vývoj dvou soupeřících populací s omezenou populační kapacitou (např. káně lesní a poštolka obecná) pro tyto hodnoty parametrů:  $r_k = 0,8$ ,  $I_{kp} = 0,2$ ,  $k_k = 2,0$ ,  $r_p = 0,6$ ,  $I_{pk} = 0,3$ ,  $k_p = 1,0$ . Počáteční populace volte jako  $k = 0,01$ ,  $p = 1,0$ . Poté změňte interakční koeficienty na  $I_{kp} = 1,5$  a  $I_{pk} = 0,6$ , zbytek ponechejte. Výsledky zanešte do jednoho grafu závislosti velikosti populací na čase, diskutujte.
- c) Ověřte důležitost pivotizace. Vyřešte soustavu

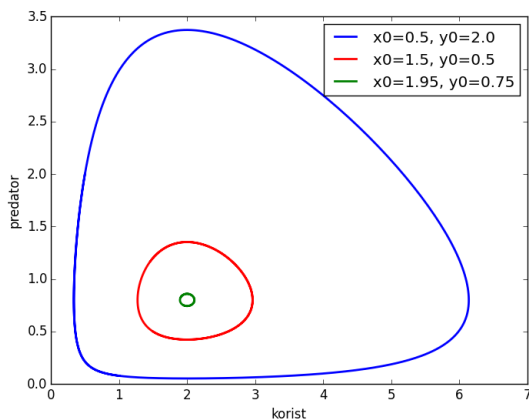
$$\begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

nejprve přesně (na papíře), poté s využitím LU dekompozice s (částečnou) pivotizací (využijte nějakou knihovni funkci, např. `scipy.linalg.lu()`), a nakonec pomocí LU dekompozice bez pivotizace (to si budete muset sami naprogramovat). Porovnejte výsledky  $\mathbf{x}$  z jednotlivých metod a výsledky zpětného vynásobení matic  $L^{-1} \cdot U$  (resp.  $P \cdot L^{-1} \cdot U$  v případě s pivotizací).

- d) Mějme nekonečný deskový kondenzátor se vzdáleností desek  $L = 10$  cm a napětím mezi deskami  $U = 5$  V. Do kondenzátoru vložíme uzemněnou elektrodu ve tvaru nekonečně dlouhého hranolu s čtvercovou podstavou o hraně  $a = 2$  cm, jejíž střed leží  $l = 6,5$  cm od uzemněné desky původního kondenzátoru (tak, že leží mezi deskami). Hranol je orientován tak, že jedna z jeho kratších hran je kolmá k deskám kondenzátoru. Nalezněte průběh elektrického potenciálu v kondenzátoru. Protože je problém symetrický vůči posunu v ose rovnoběžné s nekonečnou hranou hranolu, stačí jej řešit v řezu kolmém k této ose, jde tedy o 2D problém. V této rovině pak získaný průběh potenciálu také vykreslete. K řešení můžete využít program přiložený k zadání.

Bonus Vypočtěte a vykreslete také průběh velikosti intenzity el. pole  $\mathbf{E}$ .

- a) S využitím kódu z šestého dílu seriálu jsme spočetli časový vývoj populací kořisti a predátora pro zadané parametry a počáteční hodnoty. Populaci predátora jsme pak vykreslili jako proměnnou závislou na populaci kořisti. Výsledný graf je na obrázku 85.



Obr. 85: Vývoj populace predátora a kořisti pro různé počáteční podmínky.

Jelikož jsou řešení Lotkova–Volterrova modelu predátor–kořist periodická, dostáváme v rovině  $xy$  uzavřené křivky. Vidíme, že případ s vysokou počáteční populací predátora předvídá, že v důsledku nedostatku kořisti klesne během vývoje populace predátora do velmi nízkých hodnot. Také si všimneme nepravidelného tvaru křivky. Pro druhou volbu počátečních hodnot je křivka již pravidelnější a pro třetí volbu je téměř eliptická. Dalším postřehem je, že se křivky nekříží – kdyby tomu tak nebylo, neměla by soustava diferenciálních rovnic jednoznačné řešení.

Protože se křivky nekříží a jsou uzavřené, postupnou změnou počátečních populací můžeme dokonkovat do jednoho bodu. Jedná se o stacionární bod

$$(x, y) = (D_s/r_s, r_m/D_m) = (2, 0; 0, 8).$$

Pokud tedy nastartujeme simulaci poblíž tohoto bodu, dostaneme téměř eliptické křivky, neboť pro malé změny populací můžeme Lotkovy–Volterrovy rovnice nahradit harmonickým oscilátorem, jehož řešením ve fázovém prostoru jsou právě elipsy. Časový vývoj probíhá proti směru hodinových ručiček.

Řešení bonusové úlohy začíná vydělením diferenciální rovnice pro predátora

rovnici pro kořist. Diferenciály času se zkrátí<sup>36</sup> a dostaneme jednu rovnici

$$\frac{dy}{dx} = \frac{y(r_y x - D_y)}{x(r_x - D_x y)}.$$

Na pravé straně jsme rovnou provedli vytknutí, aby bylo zřejmé, že rovnici lze řešit separací proměnných. Integrujeme<sup>37</sup>

$$\int dy \frac{r_x - D_x y}{y} = \int dx \frac{r_y x - D_y}{x}.$$

Obě strany mají podobný tvar, podívejme se proto jen nalevo. Integrál z  $D_x$  je triviálně  $yD_x$ , integrace  $r_x/y$  vede na  $r_x \ln y$ . Nezapomeneme na konstantu a dostaneme

$$r_x \ln y - D_x y = r_y x - D_y \ln x + C.$$

V tomto tvaru již rovnici necháme, neboť nemá analytické řešení. Kdybychom bývali chtěli, mohli jsme numerické řešení Lotkových-Volterrových rovnic vynechat a použít rovnou toto exaktní řešení, pouze bychom museli určit konstantu  $C$  z počátečních podmínek.

Zde i v seriálu jsme zmínili, že poblíž stacionárního bodu se řešení chová stejně jako harmonický oscilátor. Padlo u toho slůvko „linearizace“. Linearizací se myslí rozvoj řešení okolo stacionárního bodu do prvního (lineárního) řádu. Abyste si nemysleli, že se jedná o nějakou magii, ukážeme si zde odvození. Píšeme  $x(t) = x_0 + x_1(t)$ , kde  $x_0$  je souřadnice stacionárního bodu, na čase samozřejmě nezávislá, a  $x_1(t)$  je na čase závislá odchylka o stacionární polohy. Podobně pro  $y(t)$ . Po dosazení do Lotkových-Volterrových rovnic máme

$$\dot{x}_1 = r_x(x_0 + x_1) - D_x(x_0 + x_1)(y_0 + y_1).$$

Použili jsme zde značení časové derivace tečkou. Jelikož odchylky od stacionárního bodu jsou malé, můžeme zanedbat člen  $x_1 y_1$  a po dosazení známých hodnot  $x_0$ ,  $y_0$  nám zbude

$$\begin{aligned} \dot{x}_1 &\approx r_x x_1 - D_x x_0 y_1 - D_x x_1 y_0 \\ &= r_x x_1 - D_x \frac{D_y}{r_y} y_1 - D_x x_1 \frac{r_x}{D_x} \\ &= -\frac{D_x D_y}{r_y} y_1, \\ \dot{y}_1 &\approx \frac{r_x r_y}{D_x} x_1. \end{aligned}$$

První rovnici zderivujeme podle času a napravo dosadíme z druhé rovnice. Podobnou operaci uděláme s druhou rovnicí a dostaneme rovnice harmonického oscilátoru

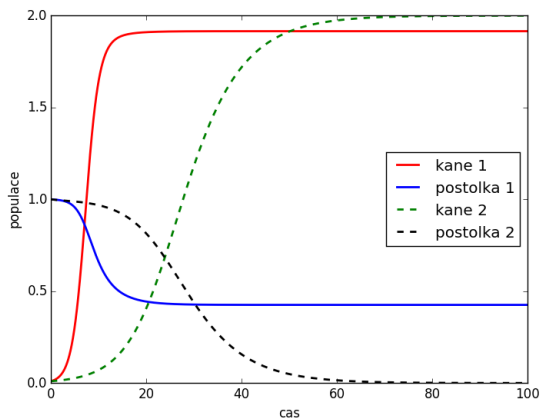
$$\begin{aligned} \ddot{x}_1 + r_x D_y x_1 &= 0, \\ \ddot{y}_1 + r_x D_y y_1 &= 0. \end{aligned}$$

<sup>36</sup>Možná už jste někde zaslechli, že za výroky typu „ $dx$  se nám vykrátí“ se vyhazuje z Matfyzu, zvlášť pokud studujete nějaký matematický obor. Je pravda, že diferenciály jakožto samostatné matematické objekty dělit nelze (operace není definována), ale v Leibnizově notaci můžeme dát dělení diferenciálů dobrý význam na základě řetězového pravidla  $\frac{dy(x(t))}{dt} = \frac{dy}{dx} \frac{dx}{dt}$ .

<sup>37</sup>„A tady už jako můžeme pracovat s diferenciály jako se zlomkem?“ Ne. Jedná se opět o řetězové pravidlo. Jak zde bylo použito, vám necháme na rozmyšlení.

Všimněte si, že frekvence  $\sqrt{r_x D_y}$  závisí pouze na parametrech, které nevystupují v interakčních členech.

- b) Kompetitivní model vyžaduje pouze přidání jednoho koeficientu, úpravu předpisů diferenciálních rovnic a výstupu z funkce, jinak se simulační kód nijak neliší od předchozího případu. Pro dané koeficienty jsme dostali graf na obrázku 86.



Obr. 86: Časový vývoj dvou soupeřících populací s různými interakčními parametry.

Pro malé hodnoty interakčních parametrů se velice rychle ustanoví rovnováha. Přesné velikosti populací, ke kterým řešení konverguje, získáme vypočtením polohy stacionárního bodu

$$(x, y) = \left( \frac{k_x - I_{xy}k_y}{1 - I_{xy}I_{yx}}, \frac{k_y - I_{yx}k_x}{1 - I_{xy}I_{yx}} \right) \doteq (1,915; 0,426).$$

Pro druhou volbu parametrů dostaneme stacionární bod  $(x, y) = (5, 0; -2, 0)$ . Vidíme, že zde již nemůže dojít ke konvergenci ke stacionárnímu bodu, protože v něm vystupuje záporná hodnota. Druhá populace proto poměrně rychle (exponenciálně) konverguje k nule a první populace ke svojí nosné kapacitě. Kdybychom volili v druhém případě  $I_{pk} = 0,5$  místo 0,6, dostali bychom speciálně stacionární hodnotu  $y = 0$  a druhá populace by se v limitě  $t \rightarrow \infty$  chovala jako  $1/t$ , což je pomalejší pokles než ten, který pozorujeme s původními parametry.

Za povšimnutí také stojí, že pokud jsou organismy schopny koexistovat, jejich populace může být pro určité hodnoty parametrů (např. i v této úloze) v součtu větší, než když jedna z populací vyhyne.

- c) Označme matici soustavy se zadání jako  $A$  a vektor pravých stran  $\mathbf{b}$ . Nejprve provedeme LU dekompozici na papíře, budeme přitom postupovat dle algoritmu

představeného v seriálu. Protože máme matici  $2 \times 2$ , bude  $L = L_1$ . Jediný netriviální prvek je  $l_{21} = a_{21}/a_{11} = 10^{20}$ . Matice  $L$  tedy je

$$L = \begin{pmatrix} 1 & 0 \\ -10^{20} & 1 \end{pmatrix}$$

a hledaná matice  $L^{-1}$  je

$$L^{-1} = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix}.$$

Matici  $U$  bychom dokázali spočítat součinem  $U = LA$ , maticové násobení je ale náročná operace (alespoň pro matice vyššího řádu, ale i pro  $2 \times 2$  matice bychom potřebovali 8 násobení), k problému tedy přistoupíme jinak. Z popisu algoritmu je jasné, že první řádek bude mít matice  $U$  shodný s maticí  $A$ . Protože naším cílem bylo vynulovat první sloupec pod diagonálou, bude  $u_{21} = 0$  (matice značíme velkým písmem, jejich prvky malým). Zbývající prvek vypočteme pravidlem pro maticové násobení, protože má ale každá z matic  $L_k$  na každém řádku jen dva nenulové prvky, z nichž jeden je jednička na diagonále, redukuje se maticové násobení na  $a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik}a_{kj}^{(k-1)} \forall i, j > k$ . V našem případě tedy platí  $u_{22} = a_{22} - l_{21}a_{12}$  a matice  $U$  je

$$U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix}.$$

Nyní vypočteme vektor  $\mathbf{x}$ , tedy vyřešíme rovnice  $L^{-1}\mathbf{y} = \mathbf{b}$  a  $U\mathbf{x} = \mathbf{y}$ . Protože matice jsou v trojúhelníkovém tvaru, můžeme použít metodu dopředné a zpětné substituce, jak jsme si ji představili v seriálu. Platí tedy rovnice

$$\begin{aligned} y_1 &= \frac{b_1}{l_{11}} = 1, \\ y_2 &= \frac{b_2 - y_1 l_{21}}{l_{22}} = -10^{20}, \\ x_2 &= \frac{y_2}{u_{22}} = \frac{10^{20}}{10^{20} - 1}, \\ x_1 &= \frac{y_1 - x_2 u_{12}}{u_{11}} = \frac{10^{20}}{1 - 10^{20}}, \end{aligned}$$

výsledek tedy je

$$\mathbf{x} = \begin{pmatrix} \frac{10^{20}}{1 - 10^{20}} \\ \frac{10^{20}}{10^{20} - 1} \end{pmatrix} \approx \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Lehce si pak také ověříme, že  $L^{-1}U = A$ .

Nyní v Pythonu provedeme LU rozklad s částečnou pivotizací. Použijeme k tomu metodu `scipy.linalg.lu()`. Z její dokumentace zjistíme, že má jeden povinný a tři nepovinné parametry. Povinným parametrem je matice  $A$ , kterou chceme rozložit, nepovinné parametry měnit nebudeme. Dále se z dokumentace

dozvíme, že metoda provede rozklad  $A = PLU$ , kde  $P$  je permutační matice (reprezentace pivotizace),  $L$  je dolní trojúhelníková matice (to, co zde značíme  $L^{-1}$ ) a  $U$  horní trojúhelníková matice. Tyto tři matice metoda vrací v tomto pořadí. Při použití LU rozkladu s částečnou pivotizací dostaneme stejně<sup>38</sup> (tedy správné) výsledky, jen kvůli pivotizaci nyní máme rozklad  $PL^{-1}U\mathbf{x} = \mathbf{b}$ , při hledání  $\mathbf{x}$  tedy musíme nejprve vypočítat  $\tilde{\mathbf{b}} = P^{-1}\mathbf{b} = P^T\mathbf{b}$ , kde  $P^T$  je transponovaná permutační matice. Tím soustavu převedeme do tvaru  $L^{-1}U\mathbf{x} = \tilde{\mathbf{b}}$ , který už umíme řešit.

Nakonec provedeme LU dekompozici bez pivotizace. V podstatě jde o stejný postup, jaký jsme před chvílí prováděli na papíře, tentokrát jej ale chceme počítat se strojovou přesností a ne přesně. Jeho provedením zjistíme, že matice  $L$  je stejná, jako v přesném výpočtu, protože při jejím výpočtu pouze dělíme, což je z hlediska numerických chyb bezpečná operace. Pokud bychom ale počítali větší matici než  $2 \times 2$ , pak by výpočet  $L_k$  obsahoval i odčítání, chyby by se tedy objevily i zde. Matice  $U$  ale již nyní vypadá

$$U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{pmatrix}.$$

Na první pohled jde o velmi malou relativní chybu jen v jednom z prvků matice, jak ale uvidíme, tato chyba způsobí velkou chybu výsledku. Zpětným vynásobením totiž dostaneme

$$L^{-1}U = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 0 \end{pmatrix},$$

což není původní matice  $A$ . Ještě výraznější chyba je v  $\mathbf{x}$ , který nyní je

$$\mathbf{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Jasně tedy vidíme, že LU dekompozice bez pivotizace je numericky nestabilní.

- d) Skript příložený k zadání řeší velmi podobnou úlohu, pouze místo elektrody ve tvaru hranolu má elektrodu ve tvaru nekonečně vysokého válce, skript tedy potřebujeme lehce modifikovat. Nejprve si všimneme, že potenciál na horní desce je 1 V místo 5 V, což lehce upravíme v definici pravé strany zobecněné Dirichletovy podmínky v oblasti horní desky, tedy na řádku 39. Dále potřebujeme upravit rozměry simulované oblasti. Toho lze docílit úpravou hodnot  $h$ ,  $X0$  a  $Y0$ , v našem případě je ale jednodušší prostě v popisku grafu zaměnit metry za centimetry.<sup>39</sup> Hodnota  $h$ , vyjadřující vzdálenost dvou sousedních bodů mřížky, nyní nebude v metrech, ale v centimetrech. Nakonec nahradíme válec hranolem, tedy změním oblast, kde aplikujeme Dirichletovu okrajovou podmínku. Toho docílíme změnou podmínky na řádku 44, kde výraz  $(j - Ny/4)**2 + (i - Nx/2)**2 <= Ny/5$  (podmínka na to, že bod mřížky  $(i, j)$  leží v kruhu o poloměru  $\sqrt{Ny/5}$  a středu  $(Nx/2, Ny/4)$ ) nahradíme výrazem

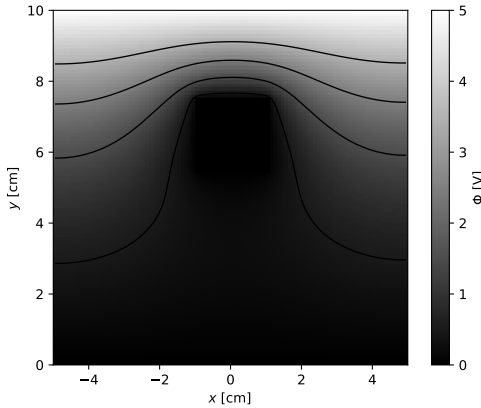
<sup>38</sup>Matice  $L^{-1}$  a  $U$  jsou kvůli permutaci samozřejmě zcela jiné.

<sup>39</sup>V případě obecné Poissonovy rovnice bychom si museli dát pozor i na správné přeškálování hodnot na pravé straně.



$\text{abs}(j-3.5/h) \leq 1.0/h$  and  $\text{abs}(i-Nx/2) \leq 1.0/h$ , podstava hranolu tedy bude mít stranu délky 2 cm a její střed bude vzdálen 3,5 cm od horní desky (tedy 6,5 cm od spodní, uzemněné desky). Protože je kondenzátor nekonečný, můžeme hranol v  $x$ -ovém směru umístit kamkoliv, nejlepší je ale umístit jej doprostřed simulované oblasti, jednak abychom vystihli symetrii úlohy, jednak abychom byli co nejdále od krajů, kde používáme von Neumannovu podmínku, tak, aby tato podmínka byla fyzikálně co nejvíce opodstatněná.

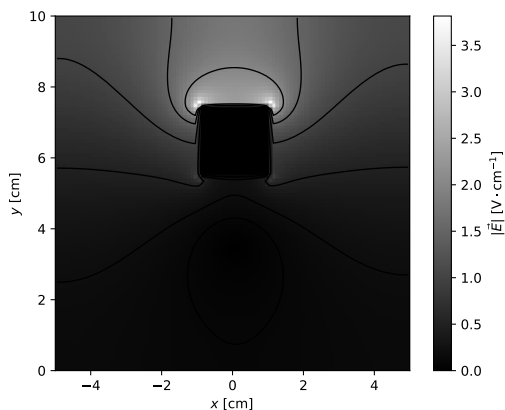
Zbývají dva volné parametry simulace  $Nx$  a  $Ny$ , tedy počet bodů mřížky, na které simulujeme. Pokud obě hodnoty necháme na výchozí hodnotě 100, což, jak jistě uznáte, není nijak zázračné rozlišení, bude matice soustavy řádu 10 000, protože pro každý bod mřížky budeme mít jednu rovnici. Matice tedy bude mít 100 000 000 prvků, pokud každý z nich bude typu `double`, tedy zabere 8 B paměti, celá matice zabere cca 800 MB. I s takto malým rozlišením (navíc pouze ve 2D) jsme na hranici možností běžného počítače (typická velikost paměti RAM je několik GB). Naštěstí je tato matice velmi řídká, čehož lze využít a ušetřit tak paměť, je pak ale třeba použít speciální (převážně iterativní) metody pro řešení dané soustavy. Pokud máme program správně nastaven, zbývá jej pouze spustit. Výslednou mapu potenciálu vidíme na obrázku 87.



Obr. 87: Rozložení elektrického potenciálu v řezu kondenzátorem.

Elektrickou intenzitu pak vypočítáme ze vztahu  $\mathbf{E} = -\text{grad } \varphi$ , musíme tedy zjistit první derivace v obou směrech. Nejjednodušší je využít dopředné diference, např. pro  $x$ -ovou složku můžeme psát  $E_x = -(\text{phi}[:, 1:] - \text{phi}[:, :-1]) / h$ , kde  $\text{phi} = \mathbf{x}.\text{reshape}(Ny, Nx)$ . Výraz výše říká, že od pole s uřízlým prvním sloupcem odečteme pole s uřízlým posledním sloupcem (prvek po prvku), v každém prvku tedy od sebe odečteme hodnoty sousedních buněk v řádku, což odpovídá dopředné diferenci. Pole  $E_x$  tedy bude mít o jeden sloupec méně,

než pole  $\phi$ . Obdobně provedeme dopřednou diferenci v  $y$ -ovém směru, ta zase bude mít o jeden řádek méně. Abychom získali velikost, potřebujeme (prvek po prvku) provést  $E\_size=np.sqrt(Ex*Ex+Ey*Ey)$ . To se nám ale nepodaří, protože  $Ex$  a  $Ey$  mají jiný tvar. V obou polích tedy musíme dodatečně umazat jeden (poslední) sloupec, resp. řádek. Výsledné příkazy tedy jsou  $Ex=-(\phi[:-1,1:]-\phi[:-1,-1])/h$  a  $Ey=-(\phi[1:,-1]-\phi[:-1,-1])/h$ . Další možností je nahlédnout do dokumentace NumPy a zjistit, že nabízí metodu `numpy.gradient()`, která přímo vrací gradient dodaného pole. Počítá jej přitom centrovanou diferencí tam, kde to lze a dopřednou nebo zpětnou diferencí u krajů. Výsledek je tedy přesnější, než co jsme spočítali my, navíc má stejný počet řádků a sloupců, jako původní pole  $\phi$ .



Obr. 88: Rozložení velikosti intenzity el. pole v řezu kondenzátorem.

Výslednou velikost intenzity el. pole vidíme na obrázku 88. Všimněme si, že intenzita je nejvyšší u rohů elektrody, pozorujeme známý efekt „sršení“.<sup>40</sup> Kromě těchto oblastí je nejvyšší intenzita v oblasti, kde je mezi uzemněnou a neuzemněnou elektrodou nejmenší vzdálenost. Jde o očekávaný výsledek, neboť se zde potenciál musí změnit na krátkém úseku, jeho derivace tedy bude velká. Tyto dva jevy hezky ilustrují, proč blesk při bouřce často udeří do vysokých a (relativně) špičatých objektů, jako jsou třeba komíny a osamělé stromy.

<sup>40</sup>V případě přesného řešení je přímo na rohu dokonce nekonečná, nicméně my zde počítáme pouze aproximaci na mřížce o konečném počtu bodů, proto dostáváme konečné hodnoty. Pokud zvýšíme počet bodů mřížky, intenzita v bodech nejbližší k hranám skutečně roste do nekonečna. V reálné fyzice dokonalé hrany ovšem neexistují, ale nabitý „hrot jehly“ se skutečně využívá na dosažení vysokých intenzit.



## Akce FYKOSu

### Podzimní soustředění v Zelené Lhotě

Podzimní soustředění proběhlo v Zelené Lhotě ve dnech 23. 9.–1. 10. 2016.

#### Organizátoři

Filip Ayazi, Jáchym Bártík, Markéta Calábková, Michal Červeňák – kápo, Daniel Dupkala, Jakub Dvořák, Erik Hendrych, Jakub Jambrich, Karel Kolář, Mikuláš Matoušek, Matěj Mezera, Michal Nožička, Daniela Pittnerová, Štěpán Stenclák, Jakub Šafin, Lukáš Timko

#### Účastníci

Karel Balej, Dominik Beňo, Vít Beran, Katarína Častulíková, Lubor Čech, Martina Daňková, Tomáš Dulava, Robert Gemrot, Marie Grunová, Ivan Hudák, Kateřina Charvátová, Marek Jankola, Patrik Kašpárek, Radka Křížová, Lucie Kundratová, Karolína Letochová, Viktor Materna, Josef Minařík, Zuzana Richterová, Kateřina Rosická, Pavla Rudolfová, Jakub Růžička, Hana Slámová, Jakub Smolka, Marco Souza de Joode, Martin Vavřík, Eva Vochozková, Jiří Zelenka



Experiment s difrakcí světla.



Účastníci zajali jednoho z povstalců.

### *Legenda*

Jaké je vaše nejtajnější, nejniternější přání? Co kdybychom vám řekli, že v okolí bývalé jaderné elektrárny, ve středu tzv. Zóny, žije entita, která je dokáže plnit? Je ale potřeba se k ní nejprve dostat, což bude obtížný úkol, nikdo živý ji totiž nikdy neviděl. Přesně o to se pokusila parta 28 dobrodruhů. Museli se vypořádat s každodenními strastmi života v Zóně, jako vůbec získat povolení k pobytu nebo opatřit si svého mutanta jako domácího mazlíčka. Při hledání středu Zóny málem naletěli na léčku vědců, kteří je pouze chtěli využít ke svým vlastním plánům, jednou je setkání s frakcí povstalců dokonce stálo život. Ti nejvytrvalejší se ale přeci jen nakonec do středu Zóny dostali. A co si přáli? To ví jen oni a Zóna...

### *Jarní soustředění v Zásece*

Podzimní soustředění proběhlo v Zásece u Radostína nad Oslavou ve dnech 3. 4.–10. 4. 2018.

### *Organizátoři*

Filip Ayazi, Jáchym Bártík, Markéta Calábková, Michal Červeňák, Jakub Dolejší, Daniel Dupkala, Jakub Dvořák, Jakub Jambrich, Karel Kolář, Mikuláš Matoušek, Matěj Mezera, Kristína Nešporová, Filip Pastierovič, Daniela Pittnerová, Štěpán Stenclák, Lukáš Timko

## Účastníci

Jáchym Bareš, Jan Benda, Marko Bermell, Katarína Častulíková, Lubor Čech, Tomáš Červeň, Martina Daňková, Robert Gemrot, Marie Grunová, Soňa Husáková, Kateřina Charvátová, Marek Jankola, Jindřich Jelínek, Patrik Kašpárek, Vojtěch Klimeš, Radka Křížová, Lucie Kundratová, Karolína Letochová, Viktor Materna, Šimon Pajger, Kateřina Rosická, Pavla Rudolfová, Hana Slámová, Marco Souza de Joode, Šárka Štěpánková, Martin Vavřík, Eva Vochozková, Jiří Zelenka



Záchrana skartovaných dokumentů.



Účastníci při hře.

### Legenda

K výcviku ve středisku Zásoka bylo pozváno 28 nových rekrutů, aby se z nich stali agenti tajné služby Agentura FYKOS 7. V průběhu výcviku se například naučili plížit v noci střeženou oblastí, či znovu skládat již skartovaný dokument. Výcvik byl ale nečekaně přerušen zprávou, že na ústředí AF7 byl spáchán útok a oni jsou nyní jediní přeživší členové Agentury. Záhy se jim podařilo vypátrat, že byl útok veden z okolí Ria de Janeiro, kde shodou okolností probíhala vědecká konference. Kufr s vědeckými výsledky ale odletěl do jiné destinace, rekruti tedy museli v rámci experimentálního odpoledne v improvizovaných podmínkách získat výsledky nové. Na konferenci narazili na stopu, jež je přes Marshallovy ostrovy a kasino v Macau zavedla až k hledaným teroristům. Ti ale rekruty dostali do léčky a postavili je před volbu – buď se všichni rekruti obětují, nebo zamoří celé město biologickou zbraní. Poté, co všichni rekruti rozkousli svoji kyanidovou kapsli, vešel do místnosti šéf výcviku a rekrutům pogrataloval ke zdárně složené finální zkoušce. AF7 se tak úspěšně rozrostla o nové agenty.

### FYKOSí Fyziklání



Soutěžící před začátkem hry.

12. ročník FYKOSího Fyziklání proběhl v pátek 16. 2. 2018. Kategorie B a C soutěžili v budovách Matematicko-fyzikální fakulty Univerzity Karlovy na Karlově a kategorie A v prostorách Hotelu Duo, kde byla zároveň ubytovaná většina soutěžících. Soutěže se zúčastnil rekordní počet 135 týmů. Letos se historicky poprvé účastnily i týmy mimo ČR a Slovenska, kteří soutěžili v angličtině. Byli to *Retamar Physics Team* ze Španělska, *RKG Hello-IT* z Lotyšska a *Patkins* z Makedonie.

Všechny tři zahraniční týmy soutěžily v kategorii A. České a slovenské týmy byly rozřazeny do kategorií podle ročníku studia jednotlivých členů. V kategorii A (tedy kategorii nejstarších) zvítězil tým *Pozitrony* ve složení Matúš Kopunec, Mikuláš Mikula, Marek Kadlečík, Robert Jurčo a Martin Babača (všichni Gymnázium Ludovíta Štúra, Trenčín). V kategorii B vyhrál tým *Hawkingovo záření* z Gymnázia Brno, tř. Kpt. Jaroše, v kategorii C se z vítězství těšil tým *Dašák* z Gymnázia Dašická, Pardubice.

### Pravidla

Soutěže se účastní družstva s nejvýše pěti členy. Na začátku soutěže dostane každé družstvo sedm příkladů. Za úspěšně vyřešený příklad si družstvo přičítá počet bodů, který závisí na počtu pokusů potřebných k jeho vyřešení. Dále si od organizátorů vyzvedne nový příklad. Samotná soutěž probíhá 3 hodiny a jejím cílem je samozřejmě získat co největší počet bodů. Přesná pravidla jsou k dispozici na webových stránkách semináře.

### Výsledky

#### **Stredoškólači A**

- |    |   |        |
|----|---|--------|
| 1. | Gymnázium Gymnázium Ludovíta Štúra, Trenčín | 189 b. |
| 2. | Gymnázium Grösslingova, Bratislava          | 177 b. |
| 3. | Gymnázium Olomouc-Hejčín                    | 168 b. |

#### **Stredoškólači B**

- |    |                                      |        |
|----|--------------------------------------|--------|
| 1. | Gymnázium Brno, tř. Kpt. Jaroše      | 163 b. |
| 2. | Gymnázium Poštová, Košice            | 142 b. |
| 3. | Gymnázium Christiana Dopplera, Praha | 137 b. |

#### **Stredoškólači C**

- |    |  |        |
|----|--|--------|
| 1. | Gymnázium Dašická, Pardubice                                 | 112 b. |
| 2. | Gymnázium Legionářů, Příbram                                 | 106 b. |
| 3. | Gymnázium Brno, tř. Kpt. Jaroše, Gymnázium Nad Štolou, Praha | 101 b. |

*Ve výsledkové listině jsou pouze nejlepší týmy. Kompletní výsledková listina včetně bodování jednotlivých úloh je na našich webových stránkách.*



Vítězné týmy přebírají ceny.

## Fyziklání online

Letos proběhl již sedmý ročník internetové soutěže Fyziklání online. Konal se ve středu 29. 11. 2017. Do soutěže se aktivně zapojilo 129 českých a slovenských středoškolských týmů, 91 zahraničních středoškolských týmů a 39 týmů v kategorii open, celkem tedy 259 týmů sestávajících z 1157 účastníků. Tím se překonal loňský rekord v počtu zúčastněných o celých 9 týmů.

Celkovým vítězem se stal tým *FtáKopySk -fks.sk* z kategorie open se ziskem 267 bodů. Na druhé příčce se umístil tým *Lundehund has six toes* z kategorie open s 226 body a třetí pozici obsadil tým *SuperPentaPhosphate* se 188 body, který současně vyhrál kategorii zahraničních středních škol. Vítězem kategorie A českých a slovenských středoškoláků se stal tým *Vypínači Hejčín* se 141 body, čímž se umístil na celkovém sedmáctém místě a prvním místě mezi středoškolskými týmy. Ziskem 124 bodů se tým *Hawkingovo záření* dostal na první místo v kategorii B a celkové 24. místo. Z týmů kategorie C na první příčku dosáhl tým *#To je na vás. :/»*, který získal 94 bodů a celkově tedy skončil na 43. místě.

## Pravidla

Na začátku soutěže každý, maximálně pětičlenný, tým obdržel sedm úloh s jednoznačným číselným výsledkem. Po zadání správného výsledku do internetového systému získal tým zadání následující úlohy. Soutěž trvala 3 hodiny, přičemž v průběhu soutěže proběhla také půlhodinová hurry up část, v níž byly úlohy rozděleny do tří fyzikálních témat a vyřešení každé úlohy bylo hodnoceno bonusovými body. Jelikož se soutěž konala přes internet, byly všechny pomůcky povoleny. Zakázána však byla komunikace s lidmi mimo soutěžní tým.



## Výsledky

**Středoškoláci A**

- |                      |        |
|----------------------|--------|
| 1. Vypínači Hejčín   | 141 b. |
| 2. Achtung Wurmloch! | 129 b. |
| 3. Fanatická pětka   | 128 b. |

**Středoškoláci B**

- |                           |        |
|---------------------------|--------|
| 1. Hawkingovo záření      | 124 b. |
| 2. つ●_● つ DIRETIDE つ●_● つ | 95 b.  |
| 3. Šamani                 | 92 b.  |

**Středoškoláci C**

- |                       |       |
|-----------------------|-------|
| 1. #To je na vás. :]» | 94 b. |
| 2. Gymšpitácká elita  | 88 b. |
| 3. Holub pitomý       | 82 b. |

**Zahraniční středoškoláci**

- |                                 |        |
|---------------------------------|--------|
| 1. SuperPentaPhosphate          | 188 b. |
| 2. SCI-TECH                     | 185 b. |
| 3. Carmel CornSUPERSTARs 玉米超級明星 | 181 b. |

**Open**

- |                           |        |
|---------------------------|--------|
| 1. FtáKopySk -fks.sk      | 267 b. |
| 2. Lundehund has six toes | 226 b. |
| 3. ὄπερ ἔδει δεῖξαι       | 182 b. |

*Ve výsledkové listině jsou uvedeny pouze nejlepší týmy. Kompletní výsledková listina včetně bodování jednotlivých úloh je k nalezení na webových stránkách soutěže <https://online.fyziklani.cz/cs/stats/>.*

**Fyzikální Náboj**

V tomto školním roce FYKOS organizoval ve spolupráci se slovenským FKS už potřetí soutěž Fyzikální Náboj. Letos se konal v pátek 20. října. Náboj je soutěž, která probíhá podobně jako FYKOSí Fyziklání – 5členné týmy během dvou hodin řeší fyzikální úlohy, přičemž se snaží získat co nejvíce bodů za správná řešení. Kromě kratšího času na řešení úloh se Náboj od Fyziklání liší i počtem bodů za správný výsledek – za ten soutěžící získají vždy jen jeden bod a za špatnou odpověď nedostávají žádnou penalizaci. Soutěž probíhá současně na více místech, tento rok se konala v Praze, Ostravě, Bratislavě, Košicích, maďarské Budapešti a polském Gdaňsku. V Praze jsme i tento rok soutěž organizovali na Gymnáziu Christiana Dopplera. Organizovali jsme také soutěž v Ostravě, a to na Gymnáziu Ostrava-Zábřeh.

Za Českou republiku soutěžilo celkem 63 týmů. Na základě věku byly rozděleny do dvou kategorií – Junioři a Senioři.

### **Vítězné týmy, kat. Senioři**

1. (3.) Gymnázium Christiana Dopplera, Zborovská 45, Praha 5
2. (7.) Gymnázium, Tomkova 45, Olomouc - Hejčín
3. (9.) PORG - gymnázium a základní škola, o.p.s., Lindnerova 3, Praha 8

### **Vítězné týmy, kat. Junioři**

1. (1.) Gymnázium, třída Kapitána Jaroše 14, Brno
2. (2.) Gymnázium Jihlava, Jana Masaryka 1, Jihlava
3. (5.) Gymnázium, Jírovcova 8, České Budějovice

*V závorkách je uvedeno pořadí v rámci mezinárodního žebříčku. Kompletní výsledkovou listinu najdete na webu soutěže (<https://physics.naboj.org>).*

## **Týden s aplikovanou fyzikou**

Po roční pauze jsme opět uspořádali Týden s aplikovanou fyzikou, a to tentokrát po České republice díky financím získaným v rámci Podpory nadaných žáků základních a středních škol. Hlavní pointa TSAFu je umožnit účastníkům kontakt se současnými špičkovými vědeckými pracovišti a současně s místy, kde se využívají různé aplikované formy fyziky. Akce se konala 19. až 25. listopadu 2017. Účastníci byli ubytováni v hostelu v centru Prahy, ze kterého každý den jezdili na různá pracoviště zabývající se fyzikou. Akce se zúčastnilo 20 středoškoláků. Akci podpořila skupina ČEZ.

### **Organizátoři**

Karel Kolář (vedoucí akce), Markéta Calábková, Michal Červeňák, Daniel Dupkala, Erik Hendrych, Daniela Pittnerová, Štěpán Stenclák, Lukáš Timko.

### **Účastníci**

Karel Balej, Vít Beran, Lubor Čech, Marie Grunová, Adéla Hanková, Marek Jančola, Patrik Kašpárek, Lucie Kundratová, Karolína Letochová, Tereza Pavlišová, Jana Pekařová, Daniel Pitoňák, Kateřina Rosická, Pavla Rudolfová, Jakub Růžička, Dominik Starý, Šárka Štěpánková, Ladislav Trnka, Martin Vavřík, Eva Vochozková.



Exkurze v reaktoru Vrabec.

### *Jak akce proběhla?*

První den jsme se sešli na ubytování a přejeli na Matfyz do Troji na úvodní aktivity. Po byrokratických záležitostech následovaly seznamovací hry, aby se poznali jak účastníci mezi sebou, tak se seznámili i s organizátory. Také byl věnován čas různým deskovým, společenským a karetním hrám, pro které byl pak prostor i další večery na hostelu. Účastníci se tak například snažili nalézt léky na všechny choroby v rámci jedné hry, což se jim ne vždy podařilo.

Druhý den proběhla návštěva Astronomického ústavu AV ČR v Ondřejově. Ta byla zpestřená zejména nečekajícím autobusem na zpožděný vlak, což cestu do Ondřejova protáhlo o dvě hodiny. Nicméně jsme se vítězoslavně dostali na místo a mohli jsme si prohlédnout několik místních pracovišť včetně největšího dalekohledu v ČR.

V úterý byl na programu Den s experimentální fyzikou, o kterém si můžete přečíst v dalším článku.

Ve středu jsme jeli do Liberce. Hlavním cílem tohoto dne byla iQLANDIA, což je jedno z těch větších Science Center v Česku. Ráno jsme začali program workshopem na téma Jaderná energie. Po obědě bylo dost času na to si projít alespoň část expozice iQLANDIE a na závěr jsme shlédli pořad v planetáriu. Tímto děkujeme vedení iQLANDIE, které nám návštěvu poskytlo zdarma. Den jsme zakončili v přilehlém akvaparku a v podvečer jsme se vrátili na ubytování.

Ve čtvrtek byl dopolední a odpolední program zajištěn v rámci Dne otevřených dveří Matematicko-fyzikální fakulty Univerzity Karlovy. Podvečerním programem byla fakultativní procházka historickým centrem města, kudy chodili a kde žili slavní fyzikové jako Albert Einstein či Ernst Mach.

Páteční program probíhal v rámci komplexu v Řeži – na ÚJV Řež, a. s. a na Ústavu jaderné fyziky AV ČR. Účastníci byli rozdělení do dvou skupin. Obě skupiny navštívily jeden z reaktorů – mají zde jak lehkododní reaktor „nulového výkonu“, tak reaktor s tepelným výkonem 15 MW. Další pracoviště pak byly například tandetron či cyklotron. Zejména tento den hodnotili účastníci velmi pozitivně, protože se dostupnou formou dozvěděli o mnoha zajímavostech spojených s jadernými reaktory a jaderným zařízením, se kterými se jinak nemohou jen tak setkat.

Akce pak byla zakončena závěrečnými aktivitami na Matfyzu v Troji.

### *Den s experimentální fyzikou*

Den s experimentální fyzikou (DSEF) je každoroční akce FYKOSu, která umožňuje středoškolákům nahlédnout do fyzikálních laboratoří. Letošní Den s experimentální fyzikou proběhl v úterý 21. 11. 2017 v Praze a to v průběhu týdne, kdy probíhala akce Týden s aplikovanou fyzikou. DSEFu se zúčastnilo 32 středoškoláků.

Akce začala úvodním slovem a přednáškou na téma nízkých teplot s ukázkami kapalného dusíku. V dopoledních hodinách studenti navštívili laboratoře v budovách MFF UK v Troji (V Holešovičkách 2). Mohli se tak dozvědět mnohé například o nukleární magnetické rezonanci, křemíkových detektorech částic v urychlovačích či měřicích přístrojích, které letěly do vesmíru.

Na odpolední program se část účastníků zúčastnila v Troji a navštívila školní reaktor VR-1, tzv. Vrabec, který je výukovým zařízením FJFI ČVUT v Praze. Ostatní účastníci se přesunuli k velkému laseru PALS (Prague Asterix Laser System; FzÚ AV ČR a Ústav fyziky plazmatu v.v.i.) a vodnímu plazmatronu AV ČR.



Účastníci na exkurzi.

## Víkend s aplikovanou fyzikou

Víkend s aplikovanou fyzikou je akce, kde se setkají studenti se zájmem o fyziku a společně stráví celý víkend. Tento rok zahájil pan prof. RNDr. Radomír Kužel, CSc. se svojí přednáškou *Krystalografie a rentgenová difrakce nejen ve fyzice*. Navštívili jsme Štefánikovu hvězdárnu a také jsme absolvovali prohlídku Národního technického muzea. V plánu jsme měli i večerní procházku po zákoutích Prahy od Petřína až po Karlův most, ale kvůli nepříznivému počasí se nekonala. Samozřejmě se našel čas i na zábavu a účastníci si mohli zahrát mnoho společenských her. Celou akci ukončila návštěva Interaktivní fyzikální laboratoře, kde jsme si mohli vyzkoušet mnoho zajímavých experimentů.

### Účastníci

Kateřina Rosická, Marco Souza de Joode, Robert Gemrot, Lubor Čech, Martina Daňková, Marie Grunová, Daniel Pitoňák, Jonáš Havelka, Jindřich Jelínek, Hana Slámová, Michal Holec, Jakub Pohly, Maroš Bratko, Tomáš Dulava.

## Cyklus přednášek pro středoškoláky

I letos se přednášky konaly v obou semestrech akademického roku a taktéž byly nahrávány na Youtube.<sup>1</sup> Proti plánovanému počtu devíti přednášek se jich nakonec konalo osm (po čtyřech v obou semestrech).

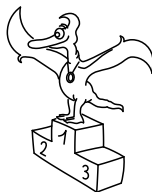
První čtveřici přednášek přinesli Daniel Slezák a Dominik Beck. V první o Eulerově metodě řezu se mohli posluchači seznámit s užitečným myšlenkovým postupem, vhodným pro řešení řady úloh z mechaniky. Na ni navazovala přednáška o obvodech střídavého proudu, která byla volným pokračováním přednášek z elektřiny a magnetismu, které doteď probíhaly každý semestr na nové téma. Nakonec následovala dvojice přednášek o termodynamice, která pokryla jak obecné principy termodynamických dějů, tak chování tepla.

V letním semestru jsme se mohli seznámit s novými přednášejícími a také novými tématy. O částicích přišel promluvit Michal Křištof, relativistické paradoxy potom konečně vysvětlil Filip Krátký, počítačové simulace rozebral a prakticky ukázal jeden z autorů letošního seriálu Lukáš Timko a nakonec Daniel Slezák v zastoupení Dominika Becka odpřednášel Geometrické metody ve fyzice.

Celkově také tento ročník zaznamenal pokles účasti na pořádaných přednáškách a i proto projekt dozná jisté změny do dalšího ročníku, které budou spočívat zejména v rozšíření našeho portfolia činností, o kterém se budete moci dočíst do konce léta na našem webu.<sup>2</sup> V prvním pololetí příštího roku se však pravidelné přednášky budou konat i nadále a již nyní vás na ně srdečně zveme. Archiv proběhlých přednášek najdete na adrese: <https://fykos.cz/akce/prednasky/archiv>.

<sup>1</sup><https://www.youtube.com/user/fykosak>

<sup>2</sup><https://prednasky.fykos.cz/>



## Pořadí řešitelů

### Kategorie prvních ročníků

jméno	škola	$\Sigma$
<i>Student Pilný</i>	MFF UK	<b>380</b>
1. <i>Robert Gemrot</i>	G Komenského, Havířov	<b>306</b>
2. <i>Martina Daňková</i>	Klasické a španělské G, Brno	<b>249</b>
3. <i>Marco Souza de Joode</i>	G Nad Štolou, Praha	<b>208</b>
4. <i>Lubor Čech</i>	G Brno, tř. Kpt. Jaroše	<b>206</b>
5. <i>Patrik Kašpárek</i>	Katolické gymnázium Třebíč	<b>180</b>
6. <i>Šárka Štěpánková</i>	G J. Ressela, Chrudim	<b>147</b>
7. <i>Víktor Fukala</i>	G Jana Keplera, Praha	<b>97</b>
8. <i>Hana Slámová</i>	G Brno, tř. Kpt. Jaroše	<b>96</b>
9. <i>Karolína Letochová</i>	G Šternberk	<b>90</b>
10. <i>Soňa Husáková</i>	G, Českolipská, Praha	<b>59</b>
11. <i>Adam Húštava</i>	European School Luxembourg II	<b>42</b>
12. <i>Jan Raja</i>	G, Nymburk	<b>35</b>
13. <i>Jiří Szotkowski</i>	G, Karviná	<b>29</b>
14. <i>Michaela Valková</i>	G Česká, Bratislava	<b>27</b>
15. <i>Lucia Gintnerová</i>	G Sv. Františka, Žilina	<b>25</b>
16. <i>Lucie Urbanová</i>	G Chotěboř	<b>23</b>
17. <i>Aleš Socha</i>	G a SOŠ, Frýdek-Místek	<b>22</b>
18. <i>Martin Polyácsko</i>	G Alejová, Košice	<b>10</b>
19. <i>Karolína Červeňáková</i>	Evanjelické G JAK, Košice	<b>6</b>

## Kategorie druhých ročníků

<b>jméno</b> <i>Student Pilný</i>	<b>škola</b> MFF UK	$\Sigma$ <b>380</b>
1. <i>Martin Schmiéd</i>	G Jihlava	269
2. <i>Jakub Jobus</i>	G PdC, Piešťany	263
3.–4. <i>Radka Křížová</i>	G J. Heyrovského, Praha	209
3.–4. <i>Viktor Materna</i>	G Brno, tř. Kpt. Jaroše	209
5. <i>Jan Benda</i>	G, Litoměřická, Praha	194
6. <i>Martin Vavřík</i>	G, Šumperk	184
7. <i>Jiří Zelenka</i>	G Z. Wintra, Rakovník	175
8. <i>Jonáš Havelka</i>	G Jírovcova, České Budějovice	148
9. <i>Pavla Rudolfová</i>	G, Vídeňská, Brno	128
10. <i>Lukáš Hronek</i>	G, Písek	85
11. <i>Eva Vochozková</i>	Biskupské G, Brno	81
12. <i>Adam Grunt</i>	G, Trutnov	70
13. <i>Jan Vondra</i>	G Týn nad Vltavou	69
14. <i>Jakub Pravda</i>	ŠpMNDaG, Bratislava	60
15. <i>Matěj Holubička</i>	G, SOŠ, SOU a VOŠ, Hořice	55
16. <i>Aneta Vacková</i>	Jiráskovo G, Náchod	53
17. <i>Milan Tichavský</i>	Slezské G, Opava	52
18. <i>Adam Křivka</i>	Cyrilomet. G a SOŠ pg., Brno	51
19. <i>Kateřina Barotová</i>	G, Olomouc-Hejčín	26
20. <i>Václav Svoboda</i>	G J. Š. Baara, Domažlice	22
21. <i>Jan Svoboda</i>	G J. Š. Baara, Domažlice	20
22.–23. <i>Soňa Curylová</i>	G F. Palackého, Val. Mez.	19
22.–23. <i>Filip Wagner</i>	G Tišnov	19
24. <i>Daniel Krátký</i>	G, Trutnov	18
25. <i>Marek Nestěra</i>	G K. Sladkovského, Praha	16
26. <i>Josef Polášek</i>	G Jana Keplera, Praha	15
27. <i>Marcel Zdeněk</i>	SPŠ strojnická a SOŠ profesora Š	14
28. <i>Lucie Ambrožová</i>	G, Svitavy	13
29.–30. <i>Marek Černoch</i>	G F. Palackého, Val. Mez.	10
29.–30. <i>Filip Novotný</i>	G Jihlava	10
31. <i>Lucia Krajčovičková</i>	G Jura Hronca, Bratislava	8
32. <i>MERT UNSAL</i>	Bahcesehir HS for Sc and Tech,TR	2

## Kategorie třetích ročníků

<b>jméno</b> <i>Student Pilný</i>	<b>škola</b> MFF UK	$\Sigma$ <b>344</b>
1. <i>Kateřina Rosická</i>	G J. Ortena, Kutná Hora	<b>329</b>
2. <i>Jindřich Dušek</i>	G Jana Keplera, Praha	<b>283</b>
3. <i>Ladislav Trnka</i>	G, Havlíčkův Brod	<b>257</b>
4. <i>Vojtěch Klimeš</i>	G, Třeboň	<b>247</b>
5. <i>Marek Jankola</i>	G M. Hattalu, Trstená	<b>214</b>
6. <i>Jindřich Jelínek</i>	G, Olomouc-Hejčín	<b>174</b>
7.–8. <i>Ivan Hudák</i>	ESŠ, Lip. Mikuláš	<b>171</b>
7.–8. <i>Lucie Kundratová</i>	G, nám. TGM, Zlín	<b>171</b>
9. <i>Josef Mínařík</i>	G Brno, tř. Kpt. Jaroše	<b>166</b>
10. <i>Tomáš Drobil</i>	G Dačice	<b>163</b>
11. <i>Jiří Blaha</i>	G, Uherské Hradiště	<b>141</b>
12.–13. <i>Marie Grunová</i>	G Moravský Krumlov	<b>114</b>
12.–13. <i>Petr Zahradník</i>	G dr. V. Šmejkal, Ústí n. L.	<b>114</b>
14. <i>Marko Bermell</i>	Slovanské G, Olomouc	<b>87</b>
15. <i>Filip Koňářík</i>	G F. Palackého, Val. Mez.	<b>77</b>
16. <i>Matej Moško</i>	G Grösslingova, Bratislava	<b>76</b>
17. <i>Tereza Prokopová</i>	G Jura Hronca, Bratislava	<b>74</b>
18. <i>Bibiána Hroncová</i>	G Poštová, Košice	<b>68</b>
19. <i>Karel Balej</i>	G a SOŠ, Rokycany	<b>56</b>
20. <i>Kateřina Charvátová</i>	G B. Němcové, HK	<b>55</b>
21. <i>Samuel Amrich</i>	G Poštová, Košice	<b>52</b>
22. <i>Jakub Růžička</i>	G, Nymburk	<b>47</b>
23. <i>Veronika Vohníková</i>	Nový PORG, Praha	<b>39</b>
24. <i>Mária Poláčková</i>	G Velká okružná, Žilina	<b>29</b>
25. <i>Daniel Staník</i>	G Uničov	<b>25</b>
26. <i>Daniel Pitoňák</i>	G a SOŠP, Čáslav	<b>20</b>
27. <i>On Tai Wu</i>	Li Po Chun UWC, Hong Kong	<b>17</b>
28.–29. <i>Tereza Pavlišová</i>	G, Olomouc-Hejčín	<b>16</b>
28.–29. <i>Jaromír Sladkovský</i>	PORG, Praha	<b>16</b>
30.–31. <i>Adéla Foglarová</i>	G, Špitálská, Praha	<b>14</b>
30.–31. <i>Jakub Šmolka</i>	Slezské G, Opava	<b>14</b>
32.–33. <i>Zuzana Fialková</i>	Sunny Can. International Sch.	<b>10</b>
32.–33. <i>Martin Škoudlil</i>	G T. G. Masaryka, Litvínov	<b>10</b>
34.–35. <i>Domínik Beňo</i>	G L. Svobodu, Humenné	<b>9</b>
34.–35. <i>Jana Pekařová</i>	G Volgogradská 6a, Ostrava	<b>9</b>
36. <i>Kristýna Kamenářová</i>	G, nám. TGM, Zlín	<b>6</b>
37. <i>Richard Veselý</i>	G, Budějovická, Praha	<b>5</b>
38. <i>Michal Jůza</i>	G, Benešov	<b>3</b>
39. <i>Jakub Zemek</i>	G, Uherské Hradiště	<b>1</b>



## Kategorie čtvrtých ročníků

<b>jméno</b> <i>Student Pilný</i>	<b>škola</b> MFF UK	$\Sigma$ <b>344</b>
1. <i>Šimon Pajger</i>	G Velká okružná, Žilina	<b>277</b>
2. <i>Viktor Rosman</i>	G, Pelhřimov	<b>243</b>
3. <i>Tomáš Červeň</i>	G V. P. Tótha, Martin	<b>198</b>
4. <i>Katarína Častulíková</i>	1. súkromné G v Bratislave	<b>149</b>
5. <i>Tomáš Dulava</i>	Matiční G, Ostrava	<b>134</b>
6. <i>David Němec</i>	G, Tanvald	<b>109</b>
7. <i>Domínik Starý</i>	G, Benešov	<b>66</b>
8. <i>Martin Repčík</i>	G, Olomouc-Hejčín	<b>64</b>
9. <i>Jáchym Bareš</i>	G, Olomouc-Hejčín	<b>55</b>
10. <i>Zuzana Řichterová</i>	G, Pelhřimov	<b>41</b>
11. <i>Ján Pavlech</i>	G sv. Jozefa Nové Mesto n. V.	<b>40</b>
12. <i>Miroslav Hrabal</i>	G, Olomouc-Hejčín	<b>33</b>
13. <i>Jan Kučera</i>	G, Písek	<b>31</b>
14. <i>Soňa Burešová</i>	G J. Heyrovského, Praha	<b>23</b>
15. <i>Filip Keller</i>	G P. de Coubertina, Tábor	<b>20</b>
16. <i>Ondřej Buček</i>	G Brno, tř. Kpt. Jaroše	<b>17</b>
17. <i>Adéla Hanková</i>	První české G, Karlovy Vary	<b>12</b>
18. <i>Domínik Ivan</i>	ŠG PinkHarmony, Zvolen	<b>9</b>
19. <i>Vít Beran</i>	Masarykovo G, Plzeň	<b>7</b>



DANIELA PITTNEROVÁ A KOLEKTIV  
**Fyzikální korespondenční seminář**  
**XXXI. ročník – 2017/18**

*Předmluva:* Daniela Pittnerová

*Náměty úloh:*

Karel Kolář (I.4, I.5, II.1, II.2, II.4, II.P, III.2, III.E, IV.3, IV.5, IV.P, V.4, V.5, V.P, VI.P), Miroslav Hanzelka (I.S, II.5, II.S, III.S, IV.S, V.S, VI.S), Matěj Mezera (III.4, IV.2, IV.4, V.E, VI.1, VI.4, VI.E), Lukáš Timko (I.S, II.S, III.S, IV.S, V.S, VI.S), Jáchym Bártík (III.5, IV.4, V.3, VI.3, VI.E), Michal Nožička (II.E, III.1, IV.1), Daniela Pittnerová (V.2, VI.2, VI.5), Michal Červeňák (I.E, IV.E), Filip Ayazi (II.3), Erik Hendrych (I.P), Michal Koutný (I.2), Jozef Lipták (V.1), Mikuláš Matoušek (I.3), Aleš Podolník (III.3), Jakub Sláma (III.P), Tereza Steinhartová (I.1), Štěpán Stenclák (VI.E),

*Autoři řešení úloh:*

Miroslav Hanzelka (I.1, I.S, II.S, III.S, IV.S, V.S, VI.S), Matěj Mezera (II.4, III.4, IV.2, IV.4, V.E, VI.1, VI.4), Jáchym Bártík (II.5, III.3, III.5, IV.5, V.5, VI.5), Lukáš Timko (I.S, II.S, III.S, IV.S, V.S, VI.S), Jozef Lipták (I.5, II.3, IV.P, V.1, VI.3), Jakub Jamblich (II.1, III.2, VI.P), Mikuláš Matoušek (I.3, III.P, VI.E), Václav Mikeska (I.4, IV.3, V.3), Daniela Pittnerová (III.1, V.2, VI.2), Kateřina Rosická (I.E, IV.E), Kateřina Smítalová (II.2, IV.1), Markéta Calábková (V.P), Karel Kolář (II.P), Michal Koutný (I.2), Jaromír Mielec (I.P), Filip Pastierovič (V.4), Jakub Šafin (II.E), Pavol Šimko (III.E),

*Numerické metody a počítačové simulace:* Miroslav Hanzelka, Lukáš Timko

*Legenda podzimního soustředění:* Michal Červeňák

*Legenda jarního soustředění:* Filip Ayazi

*Sazba:* Markéta Calábková

*Obrázky a grafy:* Miroslav Hanzelka, Mikuláš Matoušek

*Jazykové korektury:* Daniela Pittnerová, Kateřina Smítalová, Jozef Lipták

*Odborné korektury:* Jakub Dolejší, Jakub Šafin, Jáchym Bártík, Matěj Mezera



DANIELA PITTNEROVÁ A KOLEKTIV  
**Fyzikální korespondenční seminář**  
**XXXI. ročník – 2017/18**

Vydal MatfyzPress  
nakladatelství Matematicko-fyzikální fakulty  
Univerzity Karlovy  
Sokolovská 83, 186 75 Praha 8  
jako svou 564. publikaci

Výtiskl ReproStředisko MFF UK

Publikace neprošla recenzním ani lektorským řízením  
Nakladatelství neodpovídá za kvalitu a obsah textu  
Vydáno pro interní potřebu MFF UK  
Neprodejná publikace

302 stran, 88 obrázků  
Sazba písmem Latin Modern v programu X<sub>q</sub>L<sup>A</sup>T<sub>E</sub>X  
Vydání první  
Náklad 400 výtisků

Praha 2018

ISBN 978-80-7378-370-9

