

Úvodník – str. 2 • Zadání úloh 4. série – str. 2 a 29

Téma 6: Měnová reforma – str. 4

Doc.^{MM} Š. Šimsa: Měnová reforma s využitím počítače – str. 5

Řešení úloh 2. série – str. 10 • Seznámení s programem Gnuplot – str. 22

Výsledková listina – str. 31

Milí kamarádi,

chtěli bychom se vám omluvit za mírné zpoždění ve vydání tohoto čísla. Velkou měrou se na něm podepsalo probíhající zkouškové období a také drobné technické problémy, které se nám již našťěstí podařilo vyřešit.

Tímto číslem jsme se přehoupli do druhé poloviny letošního ročníku. Co to pro vás znamená? Hlavně to, že se začíná blížit termín jarního soustředění, které proběhne ve dnech 10. až 18. dubna v krásném prostředí Moravskoslezských Beskyd nedaleko Rožnova pod Radhoštěm. Vzhledem k tomu, že bychom vám chtěli poslat pozvánky na soustředění s dostatečným předstihem, je uzávěrka bodů 7. března. Do tohoto data můžete ještě posílat řešení témat a úloh ze 4. série.

Novinkou je, že webová verze našeho časopisu bude již brzy archivována Národní knihovnou. Z toho důvodu jsme nuceni upřesnit licenční pravidla našeho časopisu. Nově bude náš časopis, spolu s celým obsahem našeho webu, volně šířen pod licencí *Creativ Commons*. V případě, že nám zašlete článek k otisknutí, předpokládáme, že s touto licencí souhlasíte. Odměnou vám za to bude, že vaše texty budou uchovány pro budoucí generace. Podrobnosti o licenci naleznete na našem webu.

Pěkný zbytek zimy přejí

organizátoři 

Zadání úloh

Termín odeslání čtvrté série: 29. 3. 2010
(7. 3. 2010 pro účast na soustředění)

Úloha 4.1 – Alarm (5b)

Agent R se blížil k budově. Hbitě se schoval za strom a počkal, než projde hlídka. Následovalo pár rychlých kroků a už byl u vchodových dveří. Odklopil kryt bezpečnostního zařízení a chtěl zadat přístupové heslo. Ale tu se zarazil.

„Jak že byl ten přístupový kód?“ zamyslel se a vytáhl z kapsy papírek, na kterém měl napsáno: Kód jsou všechna řešení rovnice¹

$$a! + b! + c! = a! \cdot b! .$$

„Je jasné, že čísla a , b a c jsou přirozená, jiná se nedají zadat“, uvědomil si a ještě chvíli řešil tuto rovnici. Pak ho to přestalo bavit a raději našel kámen, kterým rozbil nejbližší okno a vniknul do budovy.

Uměli byste si poradit méně nápadně než agent R, tedy vyřešit tuto rovnici v oboru přirozených čísel?

¹ Číslo $n!$ znamená n faktoriál a vyjadřuje součin $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

Úloha 4.2 – Výtah (5b)

Agent R proběhl prázdnou místností a vešel do prostorné haly. Představa, že by měl jít až do nejvyššího n -tého patra pěšky, ho moc nenadchla. Proto si přivolal výtah.

Když čekal, než přijede, rozezněla se kolem něj siréna. Hlídka si konečně všimla rozbitého okna a vyhlásila poplach. Výtah zrovna přijel, a tak do něj agent nastoupil. Začínal být čím dál více nervozní, neboť se věci začínaly komplikovat.

Výtah pomalu stoupal nahoru a agent přemýšlel, jak ho zrychlit. Jedna z možností by bylo vylézt na střechu kabiny, přerušit lano a nechat se vynést protizávažím výtahu nahoru. Ale kolik takové protizávaží může vážit?

Předpokládejte, že kabina má hmotnost M a nosnost m . Výtah si můžete představit jako kabinu a protizávaží na opačných koncích lana, které přechází přes buben motoru. Jakou hmotnost byste zvolili pro ono protizávaží, aby byl provoz výtahu dlouhodobě co nejekonomičtější? Samozřejmě předem nevíte, jak zatížený bude výtah jezdit, tak si nějak poradte.

Úloha 4.3 – Zatoulaná součástka (4b)

Po otevření dveří agent R bleskurychle zneškodnil strážce a konečně se dostal do padouchovy kanceláře. Rutinně prohrabal pár stolů, než našel, co hledal. Kufrík plný malých integrovaných obvodů.

Agent sebral kufrík a chystal se k odchodu. Tu se však stala veliká nepříjemnost. Kufrík se otevřel a součástky se rozkutálely po celé místnosti. Agent je sesbíral, ale zjistil, že z původních n součástek našel pouze k . Ví, které obvody jsou důležité, a které ne, a proto by potřeboval vědět, které součástky mu scházejí.

Integrované obvody jsou na pouzdrech očíslovány přirozenými čísly 1 až n . Agent R je už od cesty výtahem velmi nervózní a tak po něm nechtějte, aby si pamatoval nějaká čísla. Má s sebou speciální agentské hodinky, do kterých si může zapsat omezeně velké číslo. Součet čísel od 1 do n se do hodinek vejde, ale součin nikoliv.

Jak má agent R postupovat, aby zjistil, kterých $n - k$ obvodů chybí? Jak by měl postupovat, kdyby chyběl jenom jeden obvod? A kdyby chyběly dva?

Úloha 4.4 – Rogalo (2b)

Po překontrolování agent R zjistil, že mu chybí ten nejdůležitější integrovaný obvod, a tak ho začal hledat. Během hledání jej překvapil padouch, který se vrátil do své pracovny. Při velmi vzrušujícím boji proti přesile přešel agent R na venkovní římsu, a začal po ní utíkat pronásledovatelům. Ti mu však nadběhli z druhé strany a obklíčili ho.

Agentova situace se zdála beznadějná. Když tu si všiml velkého padouchova loga, které bylo připevněno na zdi budovy. Logo vypadá jako osmiúhelník vepsaný do kružnice. Délky stran tohoto osmiúhelníku jsou po řadě 2, 2, 2, 2, 3, 3, 3, 3 dm.

Agent R logo strhl a velmi rafinovaně jej přichytil ke své kombinéze. Pak se otočil, zamával svým pronásledovatelům a skočil dolů.

Co následovalo? To záleží na ploše rogala. Jaká je plocha popsaného osmiúhelníku?

Bonusová otázka: je plocha rogala dostatečná a agent R bezpečně dosedne na zem a splní úkol, nebo je plocha rogala malá a agent misi nesplní?

Řešení témat

Téma 6 – Měnová reforma

K tématu přišel jediný článek od Doc.^{MM} Štěpána Šimsy, který si můžete přečíst níže².

Štěpán pomocí počítače našel optimální řešení pro 3 až 6 hodnot mincí, ovšem pouze za zjednodušujícího předpokladu, že při placení použijeme vždy největší minci, která nepřevyšuje částku, kterou je ještě potřeba zaplatit – tento způsob byl popsán už v druhém čísle a pro zjednodušení jej nadále budeme označovat jako *standardní*.

Jak Štěpán sám zpozoroval, pro některé hodnoty mincí, které navrhl, není standardní způsob placení optimální. Otázkou tedy zůstává to, jaké jsou nejlepší hodnoty mincí pro n různých typů mincí, pokud požadujeme, aby standardní způsob placení byl ten nejlepší možný, případně zda takové hodnoty vůbec mohou existovat.

Druhou otázkou je, jaké jsou pro n různých typů mincí nejlepší hodnoty, pokud povolíme nestandardní způsoby placení. Jsou to ty, které našel Štěpán?

Dále se můžete zabývat teoretičtějšími aspekty problému: Štěpán ve svém článku ukazuje, jak lze odhadnout medián počtu mincí potřebných k zaplacení dané částky. Dokážete odhadnout také průměrný nebo maximální počet mincí? Pokud povolíme více než 6 typů mincí nebo budeme chtít platit i větší částky (třeba od 1 do 1 000 000), počítačové řešení, které zkouší všechny možnosti, začne být nepoužitelně pomalé. Dokázali byste najít nějaké obecné vlastnosti, které budou mít nejlepší (nebo alespoň „dobré“) hodnoty mincí – mám na mysli vlastnosti typu „hodnoty“ všech mincí budou vzájemně nesoudělné apod.

² Pozn. red.: Redakčně upraveno a zkráceno.

Měnová reforma s využitím počítače

Doc.^{MM} Štěpán Šimsa

Problematika článku

Rozhodl jsem se, že jednodušší bude, když za mě všechno vypočítá počítač, a tak jsem napsal program, o němž budu hovořit. Samozřejmě zde bude i část teorie. Pro lepší rozhodování, která rozložení mincí jsou lepší, jsem vytvořil jistá kritéria, ke kterým budu při rozhodování přihlížet.

Zabýval jsem se částkami 1–99 Kč, přičemž jsem předpokládal, že každá částka se platí stejně často. Dále jsem předpokládal, že každá částka je třeba alespoň jednou zaplatit, proto má vždy jedna z mincí hodnotu 1 Kč. Navíc jsem počítal s tím, že mají všechny mince různou hodnotu, neboť kdyby měly dvě mince stejnou hodnotu, bylo by to stejné, jako by bylo použito o jednu minci méně.

Požadované vlastnosti rozložení

Jak bylo zmíněno v zadání tématka, chceme od mincí, aby bylo nejvýhodnější platit nejdříve nejvyšší mincí menší, než je částka, kterou chceme zaplatit, poté nejvyšší mincí menší, než je zbývající částka, a takto pokračujeme, dokud nezaplatíme celou částku. Já tento požadavek trochu pozměním. Budu jednoduše počítat, jak by se které rozložení mincí vyplatilo, kdybychom platili výhradně zmíněným postupem.

Poté budeme samozřejmě požadovat, aby byl co nejmenší průměrný počet mincí použitých na částky 1–99 Kč. Jinak řečeno, kolik bychom museli mít v peněžence minimálně mincí, kdybychom platili výše zmíněným způsobem, abychom zaplatili postupně částky 1–99 Kč (to bude číslo 99-krát větší a zbavíme se tím nepěkných zlomků). Na tento aspekt budu klást největší důraz.

Algoritmus jednoduše vyzkouší pro každé rozložení, kolik mincí je potřeba, což udělá tak, že si pro každou z částek 1–99 zjistí, kolik je na ni potřeba mincí, což udělá algoritmem zmíněným v zadání tématu. Respektive, nevyzkouší si všechny rozložení, vypustí ty, u kterých je nepravděpodobné, že mají šanci na dobrý výsledek.

Další ze zkoumaných vlastností je, jaký je maximální počet mincí na zaplacení libovolné částky 1–99 Kč (samozřejmě opět při placení výše zmíněným způsobem) a na kolik částek je tolik mincí třeba „pro rozhodování při remíze“. Jak bylo zmíněno v zadání, pro 6 mincí a rozložení 1, 2, 5, 10, 20, 50 potřebujeme na zaplacení například částky 88 Kč 6 mincí a tolik mincí je třeba pro čtyři částky (88, 89, 98, 99).

Dále nás zajímá medián minimálního počtu použitých mincí na zaplacení jednotlivých částek, samozřejmě výše zmíněným způsobem. Ten nám říká, že alespoň polovinu částek lze zaplatit tolika mincemi.

A nakonec nás bude zajímat, kolik částek se dá zaplatit 1–3 mincemi.

Počet částek, které lze zaplatit 1–3 mincemi a medián

Nyní se zmíním, kolik by to mohlo být ideálně. V ideálním případě by šlo právě třemi mincemi zaplatit až $\binom{n+2}{3}$ částek, kde n je počet mincí. Proč? No zajímají nás vlastně všechny součty tří mincí, přičemž je můžeme použít i vícekrát. To jsou kombinace s opakováním, o nichž víme, že jich je $\binom{n+k-1}{k}$. Pokud nás tedy zajímá, kolik částek by šlo v ideálním případě zaplatit $1-k$ mincemi, bude to:

$$\binom{n}{1} + \binom{n+1}{2} + \binom{n+2}{3} + \dots + \binom{n+k-1}{k} = \binom{n+k}{k} - 1.$$

Proč platí rovnost výše, je celkem zřejmé. Pokud si k levé straně přičteme číslo $1 = \binom{n}{0}$, jasně vidíme, že podle základních pravidel kombinačních čísel se sečte s kombinačním číslem $\binom{n}{1}$, a dostaneme místo nich číslo $\binom{n+1}{1}$. To se sečte zase s dalším kombinačním číslem v naší sumě a takto nakonec dostaneme kombinační číslo $\binom{n+k}{k}$ od kterého ještě musíme odečíst číslo jedna, abychom dostali stejnou hodnotu, jako má původní suma. Je zjevné, že pro větší n , tedy pro větší počet mincí, dostaneme větší číslo. Když si nyní dosadíme, zjistíme ideální hodnoty podle mincí. Pro tři mince až 19 částek, pro čtyři mince až 34 částek, pro pět mincí až 55 částek a pro šest mincí až 83 částek. Jak ale zanedlouho zjistíme z výsledku práce, nejsou tyto hodnoty realistické pro 5 a 6 mincí pokud se zajímáme pouze o částky 1–99.

Podle předchozího vztahu si můžeme velice jednoduše zjistit, že pro tři mince nemůže být medián čtyři, ale pět už být může. To proto, že $\binom{3+4}{4} - 1 < 50$, ale $\binom{3+5}{5} - 1 > 50$. Pro čtyři mince nemůže být medián tři a méně, ale může být čtyři a více. Pro pět a šest mincí může být medián tři a více, ale méně být nemůže.

Výsledky

Zde se zmíním o výsledcích, které jsem pomocí programu zjistil, a pokusím se vybrat pro každý počet mincí některé rozložení, které prohlásím za „výherce“ pro tento počet mincí. Přiložím tabulky 10 nejlepších výsledků.

- **3 mince:** Dvě nejlepší rozložení, 1, 5, 22 a 1, 5, 23, mají naprosto stejné vlastnosti. Zbývajících osm vypsanych rozložení nemá žádnou vlastnost lepší než tyto dvě, jsou tedy jasnými vítězi. Kdybych nyní vybíral mezi těmito dvěma, záleží jestli vezmu v potaz, že menší částky se platí častěji než vyšší. Pak by se více vyplatilo první ze zmíněných rozložení. Jelikož ale celou dobu předpokládám, že se každá částka platí stejně často, budu v tom pokračovat, a tak pro mě bude důležitější aby poměr počtu použitých mincí na částky 50–99 ku počtu použitých mincí na částky 1–49 byl co nejmenší. Tomu zcela zřejmě lépe vyhovuje rozložení 1, 5, 23. Na druhém místě bude rozložení 1, 5, 22.
- **4 mince:** Využiji opět stejnou funkci programu, ale tentokrát pro čtyři mince. Dostáváme stejnou situaci jako v předchozím případě. Stejně „nejlepší“ vlastnosti mají rozložení 1, 3, 11, 37 a 1, 3, 11, 38. Z nich bych opět vybral druhou možnost, ze stejného důvodu jako předtím.

- **5 mincí:** Zde dokonce dostáváme pro šest prvních rozložení stejné vlastnosti: 1, 3, 7, 16, 40; 1, 3, 7, 16, 41; 1, 3, 7, 18, 44; 1, 3, 7, 18, 45; 1, 3, 8, 20, 44; 1, 3, 8, 20, 45. Všimneme si ovšem jedné zajímavosti. My jsme součty dělali tak, že jsme částky platili standardním způsobem. Nezabránili jsme ale, aby nebylo výhodnější platit jiným způsobem. U posledního rozložení například můžeme částku 60 zaplatit třemi dvacetikorunami, místo abychom používali pět mincí (45, 8, 3, 3, 1). Proto bych v tomto případě použil toto rozložení.
- **6 mincí:** První čtyři rozložení mají na první pohled stejné vlastnosti 1, 2, 5, 11, 25, 62; 1, 2, 5, 11, 25, 63; 1, 2, 5, 13, 29, 64; 1, 2, 5, 13, 29, 65. Ze stejného důvodu jako v předchozím případě bych zvolil poslední z těchto rozložení, kde můžeme částku 87 zaplatit třemi 29 korunami, místo abychom ji platili pěti mincemi (65, 13, 5, 2, 2).

Tabulky

Použité značení:

- (Sum) ... Suma počtu mincí pro zaplacení částek 1–99.
- (Max) ... Maximální počet mincí na zaplacení jedné částky (kolik částek se platí tolika mincemi).
- (Med) ... Medián počtu použitých mincí na zaplacení jednotlivých částek 1–99.
- (Poč) ... Počet částek, které lze zaplatit 1–3 mincemi.

Pořadí	Mince	Sum	Max	Med	Poč
1	1, 5, 22	526	10(1)	5	19
2	1, 5, 23	526	10(1)	5	19
3	1, 4, 22	534	10(1)	6	19
4	1, 5, 21	534	10(2)	5	19
5	1, 5, 24	534	10(2)	5	19
6	1, 5, 27	534	10(1)	6	19
7	1, 5, 28	534	10(1)	6	19
8	1, 4, 18	536	10(1)	6	19
9	1, 6, 28	536	10(1)	6	19
10	1, 4, 19	537	10(2)	6	19

Tabulka t6.2.1: Nejlepší rozložení pro tři mince

Naše rozložení

A jak je to s naším rozložením? Zatímco u našeho výherce pro 6 mincí jsme dosáhli součtu 313, u našeho rozložení dostáváme součet 340, což je o dost

Pořadí	Mince	Sum	Max	Med	Poč
1	1, 3, 11, 37	410	7(4)	4	32
2	1, 3, 11, 38	410	7(4)	4	32
3	1, 3, 8, 29	412	7(4)	4	32
4	1, 3, 10, 32	412	8(1)	4	33
5	1, 3, 10, 37	412	7(4)	4	32
6	1, 3, 10, 38	412	7(4)	4	32
7	1, 3, 11, 29	412	7(5)	4	32
8	1, 3, 11, 40	412	7(4)	4	32
9	1, 3, 11, 41	412	7(4)	4	32
10	1, 4, 11, 38	412	7(5)	4	32

Tabulka t6.2.2: Nejlepší rozložení pro čtyři mince

Pořadí	Mince	Sum	Max	Med	Poč
1	1, 3, 7, 16, 40	346	6(3)	4	48
2	1, 3, 7, 16, 41	346	6(3)	4	48
3	1, 3, 7, 18, 44	346	6(3)	4	48
4	1, 3, 7, 18, 45	346	6(3)	4	48
5	1, 3, 8, 20, 44	346	6(3)	4	48
6	1, 3, 8, 20, 45	346	6(3)	4	48
7	1, 3, 7, 18, 40	348	6(4)	4	48
8	1, 3, 7, 18, 46	348	6(4)	4	48
9	1, 3, 7, 19, 46	348	6(4)	4	48
10	1, 3, 8, 18, 40	348	6(4)	4	48

Tabulka t6.2.3: Nejlepší rozložení pro pět mincí

horší. U našeho rozložení se dokonce čtyři částky platí šesti mincemi, zatímco u toho, co vyhrálo v našem programu, na každou částku stačí jen 5 mincí. No a v našem rozložení se jen 51 částek zaplatí nejvýše třemi mincemi, zatímco u námi zjištěného rozložení je to celých 60 částek. Tak proč tedy používáme takové rozložení, jaké používáme? Za prvé se s čísly 1, 2, 5 a násobky desíti lépe počítá. Navíc jsme zanedbávali, že menší částky se platí častěji než ty větší. No a naše rozložení není jen pro částky do stovky, ale i pro vyšší částky, což jsme

Pořadí	Mince	Sum	Max	Med	Poč
1	1, 2, 5, 11, 25, 62	313	5(8)	3	60
2	1, 2, 5, 11, 25, 63	313	5(8)	3	60
3	1, 2, 5, 13, 29, 64	313	5(8)	3	60
4	1, 2, 5, 13, 29, 65	313	5(8)	3	60
5	1, 2, 5, 11, 24, 62	314	5(8)	3	59
6	1, 2, 5, 11, 25, 58	314	5(8)	3	59
7	1, 2, 5, 11, 25, 61	314	5(8)	3	59
8	1, 2, 5, 11, 25, 64	314	5(8)	3	59
9	1, 2, 5, 11, 25, 67	314	5(8)	3	59
10	1, 2, 5, 11, 26, 60	314	5(8)	3	59

Tabulka t6.2.4: Nejlepší rozložení pro šest mincí

také zanedbávali. Proto jsme toho zatím nezjistili dost na to, abychom mohli skutečně rozhodnout, jestli by se nevyplatilo jiné rozložení než to naše.

Výběr počtu mincí

Ještě zbývá rozhodnout, které z vítězných rozložení bychom vybrali, samozřejmě stále s našimi omezeními. Tři mince mají opravdu příliš špatné výsledky. Hledat 10 mincí, abych zaplatil jednu poměrně levnou věc bych opravdu nechtěl. A tak těžkou peněženku taky ne. Čtyři mince na tom jsou sice o dost lépe, ale pořád dost špatně oproti pěti a šesti mincím. Naproti tomu u pěti a šesti mincí už nejsou tak rapidní rozdíly, zde by to bylo na pováženou.

Závěr

Jak jsem se už zmínil, dost jsem toho zanedbával. Nebylo by na škodu zjistit, jak by to bylo pro sedm mincí, jak by to bylo, kdybychom počítali i s vyššími částkami (do tisíce, desetitisíc, ...). Navíc je také vhodné přihlédnout k tomu, že nižší částky se platí častěji, ideálně podle nějakého reálného schématu získaného statistikami z několika obchodů. A nakonec ještě přihlédnout například k pohodlnosti, čímž myslím jednoduchost počítání, která mi u našeho stávajícího rozložení opravdu vyhovuje.

Řešení úloh

Úloha 2.1 – Slavnostní přípitek

(5b)

Zadání:

Jistě znáte, jak malý princ putoval po asteroidech 325, 326, 327, 328, 329 a 330 a pak se dostal na Zemi. Nebylo to tak úplně přesně. Než malý princ na Zemi potkal lišku a zažil další dobrodružství, navštívil ještě několik planet. Po tom, co malý princ opustil zeměpisce, se dostal na malinkatou planetku, kde žil ceremoniář.

„Buď pozdraven vzácný hoste!“ vítal malého prince.

„Dobrý den,“ odpověděl malý princ.

„Špatně! Máš odpovědět: také buď pozdraven slovně ceremoniáři. A vůbec, neruš mě, mám práci,“ odbyl ceremoniář malého prince a sklonil se zpět ke svému papíru.

„Co děláte?“ neudržel svou zvědavost malý princ.

„Připravuji velmi důležitou recepci. Ale nemůžu přijít na to, jak to vhodně zrealizovat. Recepce bude začínat přípitkem a v tom je ten problém. Všechny n hostů bude sedět kolem kulatého stolu. Samosebou si každý musí při přípitku přituknout s každým. Ale jak to udělat? V jednom kole ťukání si může přituknout i více dvojic, ale nikdy se nesmí křížit rukama. To je ve vyšší společnosti nepřijatelné, neboť někteří lidé věří, že to nosí smůlu. Zároveň ale přípitek nesmí být moc dlouhý, protože by omezil další program. Tak vymyslím, jak to udělat, aby se to zvládlo na co nejméně kol,“ řekl ceremoniář a už byl opět skloněný nad svým papírem.

A co vy, věděli byste jak poradit ceremoniáři?

Řešení:

Nejdřív si rozmyslíme, jak bude situace vypadat pro malé hodnoty n (mlčky předpokládáme $n \geq 2$, jinak se neťuká). Pro $n = 2$ nám stačí jedno přituknutí, pro $n = 3$ už potřebujeme tři. Dále tedy mějme $n \geq 4$.

Pojmenujme si hosty po směru hodinových ručiček A_1 až A_n . A_1 si musí přituknout se všemi ostatními. Navíc si ale v některém kole musí ťuknout A_2 s A_n a v tomto kole si A_1 nemůže ťuknout s nikým, protože by „překřížil“ $A_2 - A_n$. A_1 si tedy $n - 1$ -krát ťuká a jednou čeká, takže určitě potřebujeme alespoň n kol.

Tím jsme zdárně vyřešili jednu polovinu úlohy. Zbývá ještě ukázat, že umíme slavnostní přípitek po n kolech skutečně ukončit k všestranné spokojenosti. Hurá do toho.

Podívejme se na stůl shora. Bez újmy na obecnosti („búno“) můžeme předpokládat, že hosté stojí ve vrcholech pravidelného n -úhelníku (rozmysli si). Pospojujeme každého hosta s každým (tj. dokreslíme strany a úhlopříčky n -úhelníku).

Nejdříve obarvíme červeně spojnicí $A_1 - A_2$ a všechny spojnice s ní rovnoběžné. Dále obarvíme oranžově $A_1 - A_3$ a všechny spojnice s ní rovnoběžné. Postupujeme podobně dále, až konečně obarvíme žlutě spojnicí $A_1 - A_n$ (a všechny spojnice s ní rovnoběžné). Navíc ještě obarvíme zeleně spojnicí $A_2 - A_n$ a všechny spojnice s touto rovnoběžné. Celkem jsme použili n barev (ověř si). Naším cílem bude ukázat, že se žádné dvě stejně barevné spojnice nekříží, a že každou spojnicí jsme obarvili. Pak totiž budeme moci vyhlásit první kolo „červené“, druhé kolo „oranžové“ atd. až poslední kolo „zelené“ a budeme hotovi.

Každé dvě stejně barevné spojnice jsou rovnoběžné, takže se nekříží. Teď pro spor předpokládejme, že existuje nějaká spojnice A_k-A_l , kterou jsme neobarvili. Jelikož n -úhelník je pravidelný, je určité A_k-A_l rovnoběžná s A_1-A_z , kde $z = k + l - 1$, popřípadě zbytek z tohoto čísla po dělení číslem n (jen jsme ošklivě napsali, že A_k , A_l , A_1 a A_z jsou v nějakém pořadí vrcholy rovnoramenného lichoběžníka). Jediný problém, který může nastat, je, že vrcholy A_1 a A_z splynou, ale pak je $k + l = n + 2$ a A_k-A_l je rovnoběžná s A_2-A_n . Tak jako tak jsme našli rovnoběžku, která je určité obarvená a to je spor.

Popsali jsme tedy, jak úspěšně zrealizovat přípitek na n kol (pro $n \geq 3$), a ukázali jsme, že méně kol nestačí, takže si můžeme poblahopřát a udělit si 5 bodů :-).

Pepa

Úloha 2.2 – Atlet (4b)

Zadání:

Další planetka už z dálky zářila svou červenou barvou. Vypadala, jako by z oka vypadla Marsu. Uprostřed jedné z planin stál atlet. Kolem něj ležely asi kilové koule. Stejnou držel atlet ve své ruce a usilovně se soustředil.

„Dobrý den, co tady děláte?“ zeptal se malý princ, když atleta uviděl.

„Trénuji. Chci totiž vyhrát sázku,“ odpověděl atlet. Ještě chvíli se soustředil. Pak zařval a odhodil kouli.

„A jakou sázku chcete vyhrát?“ vyzvídal malý princ.

„Vsadil jsem se s bratrem, že pokud hodím správnou silou a správným směrem, tak zasáhnu libovolné místo na této planetě. Některá místa jsem už trefil, ale na jiná jsem zatím nedohodil, tak trénuji,“ odpověděl atlet a už si vybíral novou kouli.

Tato sázka malého prince zaujala, a tak začal o problému přemýšlet. Povrch planety byl dostatečně rovný, aby se to atletovi povedlo. Atmosféra byla také zanedbatelná. Jak má atlet postupovat, aby vyhrál svou sázku?

Řešení:

Abychom tento problém vyřešili, není ani potřeba počítat, stačí mít nápad. Vyjdeme z prvního Keplerova zákona. Ten říká, že planety se pohybují po rovinných křivkách (kuželosečkách), kolem stálého středu. Teď již si jistě vzpomenete na celou kapitolu fyziky o raketách opouštějících Zemi (vrhy v centrálním poli), tohle je přesně ono. Začneme jednodušším případem, kdy atlet vrhá koule vodorovně.

Problém můžeme rozdělit do několika situací podle rychlosti koule. Jedno ohnisko křivky je vždy ve středu planety, druhé zvětšováním rychlosti posouváme po polopřímce atlet–střed planety. Řešení je už čistě geometrický problém průsečíku dvou křivek:

1. Pro malé rychlosti je trajektorie koule částí paraboly. Protože hážeme vodorovně, je počátek v jejím vrcholu. Koule dopadne celkem blízko (obrázek u2.2.1).
2. Pro větší rychlosti je trajektorií část elipsy. Její velikost a křivost závisejí na kombinaci síly vrhu a gravitačního zrychlení. Jedno z ohnisek elipsy je vždy ve středu planety. Druhé ohnisko s se s rostoucí rychlostí koule vzdaluje od atleta, dohodí dál a dál (obrázek u2.2.2).

3. Pak ale nastane situace, kdy obě ohniska splynou v jedno, a dostaneme kružnici. Atlet teď místo povrchu planety strefí sám sebe. Koule dosáhla kruhové rychlosti (obrázek u2.2.3).
4. Pokud se druhé ohnisko elipsy vzdaluje ještě víc od atleta, dostáváme elipsy, které zasahují dál do prostoru mimo planetu. Ale stále koule strefuje atleta do hlavy (obrázek u2.2.4).
5. Pro ještě větší síly vrhu, kdy koule překročí únikovou rychlost planety, letí po hyperbole, a odletí úplně pryč (obrázek u2.2.5).

Výsledek tedy zní: Atlet není schopen vodorovným vrhem dohodit nikam na protilehlou polokouli.

Pokud uvážíme, že atlet nebude vrhat vodorovně, ale šikmo nahoru, pak stále platí, že se koule musí pohybovat po elipse, která má jedno z ohnisek ve středu planety. To druhé však již neleží na polopřímce atlet–střed planety, ale obecně jinde (obrázek u2.2.6) Pokud si zvolíme, kde atlet stojí (jeden bod elipsy), kam má dohodit (druhý bod elipsy), a známe jedno ohnisko, máme stále nekonečně mnoho elips, po kterých může kámen letět. Volbou velikosti a úhlu rychlosti zvolíme takovou elipsu, že koule dopadne na požadované místo povrchu.

Takže pokud atlet háže šikmo nahoru, je schopen trefit jakékoli místo na planetě.

Úlohu sice lze řešit i početně, ale vzhledem k tomu, že se v zadání nepožadovalo zjištění počáteční rychlosti a úhlu na zasažení konkrétního místa na povrchu, bylo by to zbytečně složité, a nevyhnuli bychom se použití počítače.

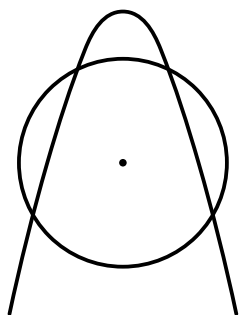
Bohužel většina z vás do řešení nenapsala, jestli uvažuje jen vodorovné vrhy, nebo i šikmé, nicméně to vypadalo, že vesměs uvažujete pouze vodorovné. Jediný, kdo explicitně uvažoval vrhy pod jinými úhly, byl Mgr.^{MM} Tomáš Pokorný, který úlohu simuloval v metapostu. Jako ukázkou, že fyzikální úlohu netřeba řešit fyzikálně, ale že je podstatné něco jiného, otiskujeme jeho řešení.

Řešení Mgr.^{MM} Tomáše Pokorného

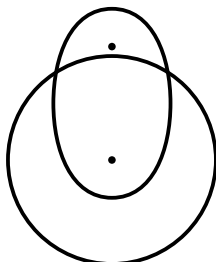
Vycházel jsem ze známého faktu, že gravitační síla klesá kvadraticky se vzdáleností, jako bonus jsem implementoval i chování gravitace uvnitř planety, kde klesá lineárně se vzdáleností od povrchu planety.

Popis programu

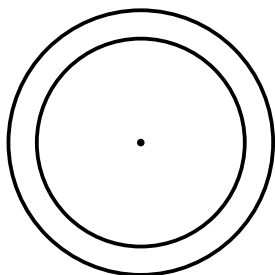
Po inicializaci proměnných nastavíme parametry simulace – počet kroků, poloměr planety, gravitaci a sílu vrhu. Vykreslíme planetu, zvolíme parametr i , který reprezentuje proměnný úhel nebo rychlost vrhu, spočítáme počáteční směr a vstupujeme do hlavního cyklu. Najdeme směr do středu planety a zjistíme, jestli jsme na povrchu nebo pod ním, a podle toho spočítáme velikost gravitační síly. Poté spočítáme vektor výsledné síly. Je to bod na spojnici aktuálního bodu a středu ve vzdálenosti f_g od aktuálního bodu. Protože chceme relativní pozici, musíme tento bod posunout o mínus (aktuální bod), protože potřebujeme polohu vůči nule a ne aktuálnímu bodu. Poté složíme tento vektor s aktuálním směrem a vyjde nám výslednice, o kterou posuneme aktuální bod,



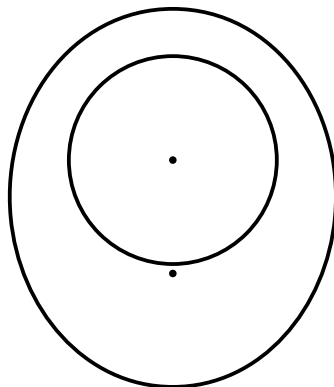
Obr. u2.2.1 – Řešení pro malé rychlosti.



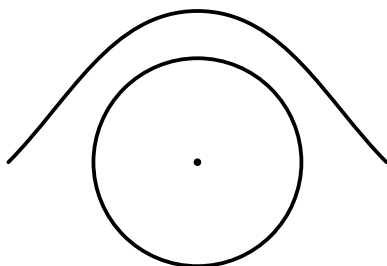
Obr. u2.2.2 – Řešení pro větší rychlosti.



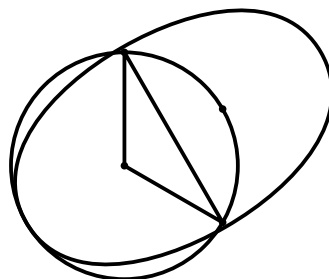
Obr. u2.2.3 – Splnutí obou ohnisek.



Obr. u2.2.4 – Elipsa, která zasahuje mimo planetu.



Obr. u2.2.5 – Překročení únikové rychlosti.



Obr. u2.2.6 – Elipsa při obecném elevačním úhlu.

a zapamatujeme si jeho novou pozici. Po dostatečném počtu kroků vezmeme pole bodů a proložíme jím křivku g . Zjistíme, kde se tato křivka protíná s povrchem planety a vykreslíme jen tu část před průsečíkem, či celou křivku, pokud se s povrchem neprotнула. Pokud zakomentujeme řádky označené „Zakomentuj pro celou trajektorii“, získáme celou trajektorii včetně části uvnitř planety.

Pozn. red.: Protože stejně neznáme hmotnost či poloměr planety ani koule, autor logicky zanedbal všechny konstanty a pracuje s abstraktními nově defi-

novanými veličinami, které s těmi fyzikálními nemají mnoho společného. To ale není na škodu věci.

Výpis programu (ke stažení na <http://mam.mff.cuni.cz/index.php3?stranka=zajimavosti>)

```
beginfig(1);
  path smerstred; % směrnice gravitační síly, tj. směr
                  % spojnice aktuálního bodu a středu planety
  path planeta; % planeta -- kružnice
  path g; % trajektorie koule
  path h; % trajektorie koule před dopadem
  pair f_v; % výsledný vektor gravitační síly
  pair smer; % výsledný vektor trajektorie v daném bodě, tj.
              % okamžitá rychlost
  pair smermem; % vektor trajektorie v předchozím bodě
  pair dopad; % čas dopadu -- pro oříznutí trajektorie na
               % povrchu planety (čas zde není reálný fyzikální
               % čas, ale metapostí parametr pro polohu na
               % křivce)
  numeric f; % Pokorného gravitační konstanta (=kappa mM)
  numeric polomer; % poloměr planety
  numeric f_g; % velikost gravitační síly
  numeric i; % iterační proměnná
  numeric j; % iterační proměnná
  numeric kroku; % počet kroků, jak dlouho se bude trajektorie
                  % počítat
  numeric rychlost; % koeficient rychlosti vůči poloměru
                    % (kolik poloměrů planety urazí koule za
                    % jeden iterační krok)

% stanovení potřebných konstant
kroku=400;
polomer:=8cm;
f=0.08cm;
rychlost=0.14;

% vykreslení planety
planeta:=fullcircle scaled (2*polomer);
pickup pencircle scaled 1mm;
draw planeta withcolor red;

% trajektorie koule
pickup defaultpen;
for i:=0 upto 60:
  z[0][i]=(polomer+3,0);
```

```

smermem:=point (.1*i) of zeme scaled rychlost;
    %Prepinani mezi konstantni silou
%smemem:= point 1 of zeme scaled i/500;
    %a konstantnim uhlem
for j:=0 upto kroku:
    smerstred:=z[j][i]--(0,0); % určení směru do středu
        % planety
    if (arclength(smerstred)<polomer):
        % zjištění, zda jsme v planetě, nebo mimo ni
        f_g:=(arclength(smerstred)/polomer)*f;
        %  $F_g=r*mM\kappa$ 
    else:
        f_g:=(polomer/(arclength(smerstred)))*
            (polomer/(arclength(smerstred)))*f;
        %  $F_g=(1/r^2)mM\kappa$ 
    fi
    f_v:=(point (arctime (f_g) of smerstred) of
        smerstred) shifted -z[j][i];
        % umístění vektoru gravitační síly do správného
        % bodu
    smer:=smermem shifted f_v;
        % složení okamžité rychlosti s gravitační silou
        % (skládat pouhým součtem rychlost se silou zní
        % hrozivě, ale vzhledem k tomu, že rychlost ke
        % středu nabytá za jeden iterační krok vlovem
        % gravitačního zrychlení v daném bodě, je
        % odlišná od síly jen konstantním faktorem --
        % závislým na hmotnosti koule a velikosti
        % iteračního kroku, tak to nevadí)
    z[j+1][i]=z[j][i] shifted smer;
    smermem:=smer;
endfor;
g:=z[0][i] % tvorba trajektorie z pole bodů proložení
    % beziérovou křivkou
for j:=1 upto kroku+1:
    ..z[j][i]
endfor;
dopad:=g intersectiontimes planeta;% protnutí křivky s
    % povrchem planety
if (xpart(dopad)=-1): % Zakomentuj pro celou trajektorii
    h:=g;
else: % Zakomentuj pro celou trajektorii
    h:=subpath(0,xpart(dopad)) of g; % Zakomentuj pro
        % celou trajektorii
fi; % Zakomentuj pro celou trajektorii

```

```

draw h;
endfor;
endfig;
end

```

Vykreslil jsem si různé varianty použité síly a úhlu vrhu, a dospěl k tomu, že lze dosáhnout libovolného místa na planetě, a to jak s konstantní silou (Pokud je alespoň tak velká, abych dohodil na opačné místo, než stojím.) tak s konstantním úhlem (Pokud není kolmý na povrch.).

Zuzka

Úloha 2.3 – Číšník (4b)

Zadání:

Následující planeta byla maličká. Větší část jí zaplňoval čtvercový stůl, který měl uprostřed jednu nožku, takže se s ním dalo otáčet. U stolu stál číšník, který leštil sklenici. Když uviděl malého prince, tak ji odložil na stůl k dalším třem a přivítal malého prince: „Vítej, vážený hoste! Posad' se u tohoto stolu prosím,“ a usadil jej.

„Děkuji,“ špitl malý princ a rozhlížel se po stole. Byly zde čtyři sklenice a jinak nic.

„Mohu ukázat své umění. Pouhé obsluhování hostů mě už přestalo bavit, tak jsem se naučil něčemu novému. Zavaž mi oči, a pak můžeš některé z těchto sklenic obrátit dnem vzhůru. Položím ruce na dvě z nich. Některé otočím. Pak ti řeknu a můžeš stolem otočit o libovolný celočíselný násobek 90° . Následně opět položím ruce na dvě sklenice a některé otočím a tak to půjde dál a dál. Uvidíš, že za chvíli budou všechny sklenice stejně orientované.“

Malého prince to opět zaujalo a tak začal přemýšlet. Podaří se to číšníkovi? Jak to dělá? Co kdyby měl číšník tři ruce a na stole bylo pět sklenic umístěných do vrcholů pravidelného pětiúhelníku? Stůl by se potom otáčel o násobky 72° .

Řešení:

Číšník si vždy vybere dvě sklenice, zjistí jejich orientaci a podle toho je potom může případně nějak otočit. Vzhledem k tomu, že neví, jak byl stůl otočen, můžeme rozlišit pouze dva možné výběry sklenic: vedle sebe nebo na úhlopříčce. Víc toho při svém výběru číšník ovlivnit nemůže.

Zkusme si nejdřív uvědomit jeden zajímavý fakt: může se stát, že na některou sklenici si číšník nesáhne ani jednou. Takže žádné řešení založené třeba na tom, že všechny sklenice bude otáčet dnem vzhůru nebude fungovat. Můžeme předpokládat, že malý princ je zlomyslný a má číšníka na první pohled přečteného, takže nebude fungovat třeba ani zkoušet otáčet sklenice nejdříve dnem vzhůru a potom, když to dlouho nebude fungovat, obráceně.

Přesto úlohu lze vyřešit. Na začátku neví číšník o sklenicích nic. Nejdříve si vybere nějaké dvě sklenice vedle sebe a otočí obě stejně, třeba dnem vzhůru. Po otočení stolu si číšník vybere sklenice na úhlopříčce a otočí obě opět dnem vzhůru. Teď už ví, že určitě jsou tři sklenice otočené vzhůru. Pokud je otočená i čtvrtá, tak mu to malý princ prozradí a číšníkovi se otáčení povedlo. Jinak ví, že poslední sklenice je otočená dnem dolů.

Když už číšník zná polohu sklenic (až na otočení stolu), má hned jednodušší práci. Nejdříve si vybere dvě na úhlopříčce. Pokud je jedna z nich dnem dolů, otočí tu a má všechny otočené stejně. Jinak otočí jednu ze sklenic dnem dolů. Má tedy vedle sebe dvě sklenice dnem dolů a dvě dnem nahoru.

V příštím (už předposledním) kroku si číšník vybere dvě sklenice vedle sebe. Pokud budou stejné, otočí je, a budou opět všechny orientované stejně. Pokud ne, stejně obě sklenice otočí. Dostane tak stejně orientované sklenice na obou úhlopříčkách. V posledním kroku si už jednoduše vybere dvě sklenice na úhlopříčce a ty otočí. Všechny sklenice budou nyní už určitě otočeny stejně.

Po pěti otočeních se mu tedy vždy povede docílit toho, aby byly sklenice stejně orientované.

Pět sklenic

Ukážeme, že v tomto případě se číšníkovi nemusí podařit všechny sklenice otočit stejně. I teď má číšník pouze dva možné tahy. Buďto si vybere tři sklenice vedle sebe nebo dvě vedle sebe a jednu proti nim.

Pokud nejsou všechny sklenice otočené stejně, určitě existují dvě s jinou orientací vedle sebe. A dokonce existují dvě jinak otočené sklenice ob jednu. Jinak by už všechny sklenice musely mít stejnou orientaci.

Předpokládejme, že číšník má nějakou strategii, podle které bude sklenice otáčet. Pokud bude v nějakém kroku otáčet sklenice vedle sebe, umístíme mu dvě různé vedle sebe tak, aby je neotočil. Podobně když bude otáčet sklenice naproti, otočíme stůl tak, aby nemohl otočit dvě různě orientované sklenice ob jedna.

Pokud budeme znát číšníkovu strategii, tak můžeme zařídit, aby nikdy neotočil všechny sklenice do stejné polohy. Takže číšníkovi se otáčení nemusí povést.

Na závěr bych chtěl poznamenat, že úloha měla dvě části. Tedy plný počet bodů bylo možné získat pouze za správné vyřešení obou. Na druhou stranu nějaké body bylo možné dostat za libovolnou z nich. Mnozí se o část s pěti sklenicemi ani nepokoušeli. A možná se tak zbytečně připravili o body.

Kuba

Úloha 2.4 – Kuchařky (2b)

Zadání:

Poslední planeta už z dálky voněla. Planetce vévodila veliká pec, u které stály tři kuchařky. Střídaly se v míchání velkého hrnce a o něčem se dohadovaly.

„O čem se hádáte?“ vpadl jim do rozhovoru malý princ.

„Vaříme polívku. Suroviny do polévky máme společné, ale neměly jsme dřevo, abychom mohly zatopit v peci. Proto jsem donesla ze svých zásob tři polínka,“ řekla jedna.

„Já jsem donesla dokonce pět polínek,“ řekla druhá.

„A já jsem těm dvěma dala 8 peněz, protože už doma žádné dřevo nemám,“ dodala ta poslední.

„Nevíte jak rozdělit peníze, aby to bylo spravedlivé? Ale to je přece jednoduché, 3 a 5 to není, to ví každý,“ řekl vesele malý princ a už utíkal na další planetu, zpět do svého dobrodružství, které popsal pan Antoine de Saint-Exupéry.

A co vy? Dovedli byste spravedlivě rozdělit peníze?

Řešení:

Na první pohled by se mohlo zdát, že správný poměr dělení peněz je 3 : 5, ale jak si ihned ukážeme není to spravedlivé dělení. Hospodyňky totiž vařili

spolu na jednom ohni, který spotřeboval 8 polínek. V případě spravedlivého zásobování dřevem by tak každá z nich musela přinést do ohně $8/3$ polínka. Jenže první dala 3 polínka, druhá 5 polínek a třetí nedala žádné.

Pokud se na to podíváme tržním systémem, první a druhá hospodyňka si spotřebovali svůj díl polínek potřebný na vaření a zbytek prodali třetí hospodyňce. První tedy prodala třetí hospodyňce

$$3 - \frac{8}{3} = \frac{1}{3} \text{ polínka,}$$

zatímco druhá jí prodala

$$5 - \frac{8}{3} = \frac{7}{3} \text{ polínka.}$$

Třetí hospodyňka zaplatila 8 penízků za $8/3$ polínka, takže jedna třetina polínka jí stála jeden penízek. První hospodyně jí prodala pouze $1/3$ polínka, za co jí náleží právě ten jeden penízek. Druhé, která sponzorovala téměř celé vaření pro třetí, zůstává tedy 7 penízků za $7/3$ polínka. Správný podíl je tedy 1:7.

Honzík

Úloha 2.5 – Měření tíhového zrychlení (2b)

Zadání:

Asi všichni znáte tíhové zrychlení. Tato konstanta určuje velikost gravitační síly, kterou Země působí na objekty v blízkém okolí povrchu. K jejímu měření můžeme využít fyzické kyvadlo. Doba kmitu fyzického kyvadla je dána vztahem

$$T = 2\pi \sqrt{\frac{I}{mgd}} \left(1 + \frac{1}{4} \sin^2 \frac{\alpha}{2} \right), \quad (1)$$

kde I je moment setrvačnosti kyvadla vzhledem k ose otáčení, m hmotnost kyvadla, d vzdálenost těžiště od osy otáčení, α je maximální úhlová výchylka těžiště z rovnovážné polohy a g je hledané místní zrychlení. Z této rovnice vychází následující model.

Matematické kyvadlo je idealizace fyzického kyvadla, kdy je předpokládáno, že máme nehmotnou nit, na které je zavěšen hmotný bod, který vykonává periodický pohyb. Moment setrvačnosti hmotného bodu je zde dán vztahem

$$I = ml^2, \quad (2)$$

kde m je hmotnost bodu a l je délka niti. Pokud předpokládáme, že se kyvadlo pohybuje v malých výchylkách, můžeme člen $(1/4) \cdot \sin^2(\alpha/2)$ ze vztahu (1) zanedbat. Úpravou pak získáváme vztah pro závislost místního gravitačního zrychlení g na periodě T

$$g = \frac{4\pi^2 l}{T^2}. \quad (3)$$

Pokud chceme dosáhnout přesnějších výsledků, musíme hmotný bod nahradit kuličkou. Tím se změní moment setrvačnosti. V našem případě je to

$$I = \frac{2}{5}mr^2 + ml^2, \quad (4)$$

přičemž m rozumíme hmotnost kuličky, r je poloměr kuličky a l je délka závěsu.

Při měření jsme změřili hmotnost kuličky $m = (278,7 \pm 0,1)$ g, délka závěsu $l = (998 \pm 1)$ mm. Kulička byla trochu potlučená, proto jsme ji změřili na několika místech. Naměřené hodnoty jsou v tabulce u2.5.2. Průměr kuličky jsme měřili s chybou $s = \pm 0,2$ mm.

Pak jsme se pustili do měření periody. Změřili jsme dobu 10 kmitů. Data jsou v tabulce u2.5.1. Používali jsme stopky ovládané optozávorou. Měřili jsme tak s chybou $s = \pm 0,005$ s.

Z naměřených dat určete tíhové zrychlení a vypočítejte chybu měření. Srovnejte výpočet pomocí vzorce (3) s případem, že bereme moment setrvačnosti ve tvaru (4). Pokud byste nechtěli jen zpracovávat tato data, můžete provést vlastní měření, pak vás odměníme bonusovými body.

Tyto data byla naměřena ve fyzikálním praktiku I na MFF UK. V případě zájmu naleznete celý studijní text k této úloze na adrese

http://physics.mff.cuni.cz/vyuka/zfp/txt_121.pdf.

Řešení:

Nejdříve vypočítáme zrychlení pomocí vztahu

$$g = \frac{4\pi^2 l}{T^2}.$$

Délku niti $l = (998 \pm 1)$ mm víme. Střední hodnotu doby kmitu určíme jako aritmetický průměr naměřených hodnot. Pro jednotlivé hodnoty určíme také střední kvadratické odchylky, které zapíšeme do tabulky u2.5.1.

Nesmíme zapomenout ověřit, zda se nějaká hodnota neliší o více než $3 \cdot s$ od průměrné hodnoty \overline{T}_{10} . Pokud ano, vyřadíme ji z dalšího zpracování. Jak je v tabulce vidět, musíme vyloučit měření číslo 7. Po přepočítání zjistíme, že se směrodatná odchylka značně zmenšila, ovšem ukázalo se, že měření 2 a 4 taky „ujela“, a musíme je proto vypustit.

K statistické chybě musíme ještě připočítat chybu měření stopek, tedy $s_{T10} = 0,005$ s. Odtud dopočítáme ³

$$T = (2,0383 \pm 0,0005) \text{ s.}$$

Dopočítáme tíhové zrychlení g . Pro určení chyby měření z vypočtené hodnoty můžeme použít pomocné vztahy $a \cdot b = (\bar{a} \cdot \bar{b}) \pm (a \cdot s_b + b \cdot s_a)$ pro výpočet T^2 a $a/b = (\bar{a}/\bar{b}) \pm (s_a/b + a \cdot s_b/b)$ pro výpočet l/T^2 . Tyto vztahy jsou pouze přibližné, proto raději použijeme kvadratický zákon přenosu chyb

$$s_F = \sqrt{\sum_{j=1}^k \left(\frac{\partial F}{\partial x_j} \right)^2 \cdot s_{x_j}^2}.$$

Formulku aplikujeme na g a dostaneme

$$s_g = \sqrt{\left(\frac{\partial g}{\partial l} \right)^2 \cdot s_l^2 + \left(\frac{\partial g}{\partial T} \right)^2 \cdot s_T^2} = \sqrt{\left(\frac{4\pi^2}{T^2} \right)^2 \cdot s_l^2 + \left(\frac{-8\pi^2 l}{T^3} \right)^2 \cdot s_T^2},$$

³ Pokud nevěříte, že můžeme pouze vydělit hodnotu a chybu měření desíti, tak si můžete vytvořit stejnou tabulku pro $T = T_{10}/10$ a uvidíte, že vše vyjde stejně.

	T_{10} [s]	$ T_{10} - \overline{T_{10}} $	T_{10} [s]	$ T_{10} - \overline{T_{10}} $	T_{10} [s]	$ T_{10} - \overline{T_{10}} $
1	20,3821	0,0052	20,3821	0,0009	20,3821	0,0009
2	20,3795	0,0078	20,3795	0,0035		
3	20,3836	0,0037	20,3836	0,0006	20,3836	0,0006
4	20,3872	0,0001	20,3872	0,0042		
5	20,3842	0,0031	20,3842	0,0012	20,3842	0,0012
6	20,3820	0,0053	20,3820	0,0010	20,3820	0,0010
7	20,4251	0,0378				
8	20,3820	0,0053	20,3820	0,0010	20,3820	0,0010
9	20,3840	0,0033	20,3840	0,0010	20,3840	0,0010
10	20,3828	0,0045	20,3828	0,0002	20,3828	0,0002
	20,3873	$s = 0,0043$	20,3830	$s = 0,0007$	20,3830	$s = 0,0004$

Tabulka u2.5.1: Určení doby periody T_{10} a chyby měření této veličiny. V prvním sloupci je uvedeno číslo měření, v dalších sloupcích naměřená hodnota a rozdíl hodnoty a průměru, v posledním řádku je uvedena průměrná hodnota $\overline{T_{10}}$ a směrodatná odchylka s .

Po dosažení (nesmíme zapomenout převést milimetry na metry) dostaneme $s = 0,0055 \text{ m} \cdot \text{s}^{-2}$ a tedy

$$g = (9,483 \pm 0,006) \text{ m} \cdot \text{s}^{-2}.$$

V případě, že nezanedbáváme moment setrvačnosti kuličky, budeme tíhové zrychlení počítat pomocí vztahu

$$g = \frac{4\pi^2}{(r+l)T^2} \cdot \left(\frac{2}{5}r^2 + (r+l)^2 \right).$$

Možná vás překvapí, proč ve vzorci máme $r+l$ a ne pouze l , ovšem pokud uvažujeme, že kulička má určitý poloměr, tak si musíme uvědomit, že se nám také zvětšila vzdálenost těžiště.⁴

Ve vzorci nám přibyl poloměr kuličky, který zatím neznáme. Určíme jej z více měření, podobně, jako jsme určovali dobu kmitu.

Z d snadno vypočítáme $r = d/2 = (20,13 \pm 0,04) \text{ mm}$. Po zohlednění chyby měření šuplerou dostaneme $s_r = \pm 0,11 \text{ mm}$. Nyní můžeme spočítat hodnotu g , vyjde nám $g = 9,8711 \text{ m} \cdot \text{s}^{-2}$.

⁴ Pozn. red.: Vzhledem k nejasné formulaci v zadání jsem za to nestrhával body.

	d [mm]	$ d - \bar{d} $	d [mm]	$ d - \bar{d} $
1	40,42	0,25	40,42	0,03
2	39,80	0,37		
3	39,90	0,27	39,90	0,13
4	40,10	0,07	40,10	0,02
5	40,44	0,27	40,44	0,03
6	40,42	0,25	40,42	0,03
7	40,12	0,05	40,12	0,02
8	40,38	0,21	40,38	0,02
9	39,86	0,31		
10	40,26	0,09	40,26	0,00
	40,17	$s = 0,09$	40,26	$s = 0,07$

Tabulka u2.5.2: Určení průměru kuličky a chyby měření této veličiny. V prvním sloupci je uvedeno číslo měření, v dalších naměřené hodnoty a rozdíl hodnot a průměru, v posledním řádku je uvedena průměrná hodnota \bar{d} a směrodatná odchylka s .

Opět využijeme kvadratického zákona přenosu chyb a určíme chybu měření vypočítaného g . Pro větší přehlednost si rozepíšeme jednotlivé členy

$$\frac{\partial g}{\partial l} = 4\pi^2 \frac{2(r+l) \cdot lT^2 - \left(\frac{2}{5}r^2 + (r+l)^2\right)T^2}{((r+l)T^2)^2},$$

$$\frac{\partial g}{\partial r} = 4\pi^2 \frac{\left(\frac{4}{5}r + 2(r+l)\right)(r+l)T^2 - \left(\frac{2}{5}r^2 + (r+l)^2\right)T^2}{((r+l)T^2)^2},$$

$$\frac{\partial g}{\partial T} = -2 \frac{4\pi^2}{(r+l)T^3} \cdot \left(\frac{2}{5}r^2 + (r+l)^2\right).$$

Členy umocníme na druhou, přenásobíme kvadrátem příslušné chyby, sečteme a výsledek odmocníme. Dostaneme $s_g = 0,018 \text{ m} \cdot \text{s}^{-2}$.

Hledaná hodnota je

$$g = (9,676 \pm 0,018) \text{ m} \cdot \text{s}^{-2}.$$

Při započítání momentu setrvačnosti kuličky, jsme dosáhli výsledku, který se více blíží tabulkové hodnotě $g = 9,80665 \dots \text{ m} \cdot \text{s}^{-2}$, než v prvním případě. Přesto se v rámci chyby měření neshodujeme. Je to dáno tím, že jsme zanedbali vliv nití, závěsu a dalších faktorů, které ovlivňovaly měření. Smyslem úlohy bylo seznámit vás se zpracováním měření a ne podrobná diskuse měření, proto se pro tentokrát s tímto výsledkem spokojíme a nebudeme dále diskutovat systematickou chybu měření.

Seznámení s programem gnuplot

Během matematicko-fyzikálně-informatického bádání často narazíme na potřebu zobrazit graficky nějakou funkci, naměřená data, výsledky počítačového výpočtu apod. Pokud se chceme v datech sami zorientovat, hodí se možnost zobrazení nejrůznějšími způsoby upravovat a měnit, naopak pokud chceme naše výsledky publikovat, zajímají nás možnosti grafické úpravy výsledného obrázku.

K většině zmíněných a podobných úkolů můžeme použít program nazvaný gnuplot. Než se pustíme do podrobného popisu, začneme krátkým souhrnem významných vlastností:

- Gnuplot umí kreslit grafy. Jak klasické s dvěma osami, tak trojrozměrné. Umí také kreslit polární grafy a zobrazovat parametrické funkce. Grafy je možné interaktivně prohlížet a zkoumat a s přiměřeným úsilím také získat velmi úhledný výstup vhodný k publikování.
- S gnuplotem si popovídáte převážně pomocí textových příkazů. Názory na takové ovládání budou určitě subjektivní, ale objektivní výhodou je snadnost uchování a znovupoužití příkazů, které jste si odladili pro vytvoření pěkného grafu, stejně tak jako jednoduché využití gnuplotu ve skriptech a pro hromadné zpracování dat.
- Vstupními daty pro kreslení může být obyčejný textový soubor, který jednoduše vytvoříte jakýmkoliv programem včetně vašich vlastních. Data jdou dokonce uložit do jednoho souboru spolu s příkazy pro kreslení, takže použití gnuplotu může být jednoduchou realizací grafického výstupu z vašeho vlastního programu.
- Výstupem může být kromě zobrazení na monitoru mnoho různých vektorových i bitmapových obrázků, i výstup vhodný pro vložení do $\text{T}_{\text{E}}\text{X}$ ového dokumentu.
- Gnuplot umí prokládat data libovolnou zadanou funkcí (tj. umí najít hodnoty parametrů ve funkci tak, aby funkce co nejlépe vystihovala data).
- Gnuplot je open source program, navíc docela malý a nevyžadující složitou instalaci, takže si jej můžete kdekoliv jednoduše stáhnout a spustit. Pokud byste chtěli, můžete si jej i upravovat.

Kde vzít . . .

Dle informací na domácí stránce projektu (<http://www.gnuplot.info>) je program ke stažení ze serveru SourceForge na adrese <http://sourceforge.net/projects/gnuplot/files/>. Pokud používáte linuxovou (BSD atd. :-)) distribuci s balíčkovacím systémem, budete mít balíček gnuplotu už pravděpodobně připraven, a stačí jej nainstalovat běžným způsobem.

Jak už jsme zmínili na začátku, licence gnuplotu umožňuje volné použití bez omezení. V případě zájmu si také můžete dle libosti upravovat zdrojové kódy programu.

Nerozumím řeči vašeho kmene . . .

Po spuštění gnuplotu (příkazem `gnuplot`, ikonkou, jak je libo) se objeví kurzor, který trpělivě čeká na naše příkazy. Příkaz je jedno slovo, případné parametry, či další upřesnění jsou oddělené mezerami (až na pár výjimek zmíněných dále). Každý příkaz nebo klíčové slovo se dá zkrátit až na nejkratší jednoznačnou délku a gnuplot nám bude stále rozumět.

Kromě příkazů vyvolávajících nějakou akci (kreslení grafu, výpis hodnoty proměnné, ukončení programu apod.) existuje trojice příkazů `set`, `unset` a `show`, které souvisí s nastavením chování gnuplotu (jak se bude kreslit, jak počítat apod.). Za `set` následuje název parametru, který nastavujeme, a za ním (pokud je to potřeba) požadovaná hodnota. Vše oddělené mezerami. Příkaz `unset` má význam vypnutí příslušné vlastnosti. Nakonec, `show` vypíše hodnotu příslušného parametru.

Dalším důležitým příkazem je `help`. Buďto zadaný samostatně, anebo následovaný mezerou a názvem hledaného tématu, funkce, či příkazu. Dostanete se tak ke kompletní dokumentaci gnuplotu, můžete si přečíst popis konkrétní funkce apod.

Mimo příkazů můžeme definovat funkce a zadávat hodnoty proměnných. Název proměnné nebo funkce může být libovolná posloupnost písmen, která není rezervovaná pro příkaz gnuplotu. Rozlišují se (stejně jako všude jinde v gnuplotu) velká a malá písmena. Hodnotu proměnné přiřadíme rovnítkem, tedy například⁵

```
gnuplot> a=42
```

Můžeme vypsát hodnotu konkrétní proměnné (`print a`), anebo všech nastavených proměnných (`show variables`).

Zde je třeba upozornit, že gnuplot rozlišuje celočíselné a neceločíselné hodnoty a pokud budou kolem lomítka jen celočíselné hodnoty, provede se i dělení v celých číslech. Odpomoci se tomu dá tím, že budeme za celočíselné hodnoty psát desetinnou tečku „navíc“:

```
gnuplot> print 3/2, 3./2, 3./2., a/4
```

První číslo odpovědi bude jednička, protože oba operandy jsou celočíselné. Stejně tak poslednímu výsledku chybí desetinná část, protože hodnotu proměnné `a` jsme také zadali celočíselně.

Funkce se od proměnných liší tím, že za jejich názvem následuje závorka se seznamem názvů argumentů (při deklaraci), případně hodnot argumentů (při volání), oddělených čárkou. Zdefinujeme funkci `norma`, která počítá velikost dvousložkového vektoru:

```
gnuplot> norma(a,b)=sqrt(a**2+b**2)
```

⁵ Doslovné příkazy pro gnuplot budeme v tomto článku sázet na samostatné řádce uvozené textem `gnuplot>`.

Zápis `sqrt` označuje druhou odmocninu, tak jak je zvykem ve většině programovacích jazyků. Seznam všech předdefinovaných funkcí si přečteme po zadání příkazu `help functions`. Dvě hvězdičky jsou v gnuplotu výraz pro mocninu. Další operátory jsou vcelku standardní, jak je můžete znát z jiných jazyků nebo programů. Gnuplot umí i otazníkový operátor ve stylu jazyka C, umožňující vložit podmínku přímo do výrazu. Například funkci vracející maximum dvou čísel tak můžeme zapsat jako

```
gnuplot> max(a,b)=(a>b)?a:b
```

Stojí za zmínku, že gnuplot umí pracovat i s komplexními čísly. Zadat komplexní číslo můžeme (mimo postupů ve stylu `sqrt(-1)`) jako dvojici čísel (reálná a imaginární složka) ve složených závorkách oddělených čárkou. Jednotlivé části komplexního čísla získáme funkcemi `real` a `imag`, příp. `abs` a `arg` (vrací absolutní hodnotu a úhel průvodiče v komplexní rovině). Například

```
gnuplot> set ang d
gnuplot> x={2.,1.}
gnuplot> print arg(x), arg(x**3)
```

vypíše úhly čísla `x` a jeho třetí mocniny, tj. 26,57 a 79,70. Úhly jsou ve stupních, jak jsme nastavili prvním příkazem (zkrácená verze `set angles degrees`).

Nakresli mi obrázek

Kreslení klasických grafů se dvěma osami zajistí příkaz `plot`. Za mezerou následují čárkou oddělené definice jednotlivých křivek. Může to být buďto matematický výraz s `x` na místě argumentu, anebo název souboru s `daty`. Můžete zkusit například

```
gnuplot> plot sin(x), x*cos(x)
```

Po potvrzení tohoto příkazu „enterem“ uvidíte graf funkcí. Zobrazení se dá interaktivně upravovat. Podle konkrétního kreslicího výstupu (může se lišit s verzí gnuplotu, operačním systémem, vašim nastavením apod.) mohou být k dispozici různé ikonky, ale pro každý interaktivní výstup bude fungovat následující.

Při pohybu myši po grafu vidíte v dolní části souřadnice a dvojklikem můžete aktuální souřadnice uložit do schránky. Kliknutím pravým tlačítkem začnete vybírat oblast grafu pro zvětšení. Pokud je okno s grafem aktivní, fungují navíc i některé klávesy: `K` výchozí velikosti grafu se vrátíte klávesou `u`, `k` předchozímu zvětšení pak klávesou `p`. Pomocí `g` zobrazíte či schováte mřížku v grafu, klávesou `l` přepnete svislou osu do, resp. z logaritmické škály. Klávesou `L` přepínáte logaritmickou škálu u osy blízké kurzoru myši. Klávesou `e` pro znovunakreslení grafu využijete hlavně při zobrazování dat ze souboru, když chcete aby se graf aktualizoval podle nového obsahu souboru. Stisknutí `h` vám v příkazovém okně vypíše kompletní seznam možností.

Hned za příkaz `plot` můžeme do hranatých závorek přidat rozsahy pro x a y osu grafu. Například

```
gnuplot> plot [0:30] [-5:] sin(x), x*cos(x)
```

První závorka udává rozsah pro x , druhá závorka pro y . Hodnoty, které nechceme zadat, můžeme vynechat (můžeme vynechat i celou závorku pro svislou osu).

Gnuplot ve většině nastavení umí pracovat s historií příkazů. Stiskem šipky nahoru se dostanete k dříve zadaným příkazům a můžete je upravovat a znovu použít. Takže není třeba při úpravě grafu psát vše znovu, stačí si najít předchozí příkaz, upravit a potvrdit.

Dvojměrné grafy jdou mimo předchozího režimu kreslit také jako parametrické anebo jako polární (případně kombinace obojího). Při parametrickém kreslení používáme na místě parametru proměnnou t a každá křivka musí být zadána jako dva výrazy oddělené čárkou, kde první je souřadnice x a druhá souřadnice y . Pro polární grafy se použije také proměnná t , která má ale význam úhlu a nabývá postupně hodnot od nuly do plného úhlu (anebo tolik, kolik určí první hranatá závorka). Určuje tedy úhel kresleného bodu od vodorovné osy. Funkce definující křivku musí na základě hodnoty t vrátit vzdálenost příslušného bodu od počátku. Např.

```
gnuplot> set ang r
gnuplot> set size sq
gnuplot> set polar
gnuplot> plot [0:pi] sin(5.*t) t "kyticka"
```

nakreslí kytičku. První `set` nastaví úhly v radiánech (což je potřeba jen pokud jsme je změnili, jinak jsou radiány výchozí), druhý `set` nastavuje čtvercový graf (`square`), poslední pak přepíná na zadávání křivek polárně. Písmenko t v kreslicím příkazu je zkratkou od `title` a nastaví popisek v legendě grafu.

Data ze souboru se kreslí podobně – místo funkce zadáme název souboru uzavřený do uvozovek. Základní formát souboru s daty je jednoduchý. Každý řádek je jeden vzorek dat, na řádku jsou mezerou nebo tabulátorem oddělené hodnoty patřící do jednotlivých sloupců.⁶ Standardně se první sloupec považuje za souřadnici x , druhý za y . Další sloupce se využijí v některých režimech kreslení, anebo je možné při vykreslování datového souboru explicitně určit, jak se sloupce interpretují. (V případě jediného sloupce se tento považuje za hodnoty y a hodnotou x je číslo řádku.) Řádky začínající znakem mířičky (`#`) se ignorují a prázdný řádek má význam oddělení bloku dat (způsobí třeba přerušování čáry při vykreslování dat lomenou čarou).

⁶ Takový formát souboru můžete mimo jiné získat z většiny tabulkových kalkulátorů, třeba kopírováním přes schránku.

Pro vyzkoušení si vytvoříme jednoduchý textový soubor obsahující

1.	6.	-3.
6.	2.	-2.
14.	-1.	3.
17.	-3.	10.

a nazveme jej třeba „data“. Pokud je ve stejném adresáři, odkud pouštíme `gnuplot`, můžeme zkusit⁷

```
gnuplot> plot "data" w l, "" u 1:3 w l, \
> "" u 1:((($2-$3)**2/10.) w l
```

Písmenko `w` je zkratkou `with` a říká, že body budou v grafu spojené čarami (ve výchozím nastavení se datové soubory kreslí jen jako jednotlivé body). Pokud bychom místo `l` uvedli třeba `lp`, bude na čáře navíc značka v místě každého zadaného bodu. Prázdné uvozovky znamenají znovupoužití předchozího souboru, `u` je zkratka `using` určující, co budou souřadnice bodu. Výraz před dvojtečkou je předpis pro souřadnici x , za dvojtečkou pro y . Pokud je souřadnice tvořena konkrétním sloupcem v datech, uvede se číslo sloupce. Ale můžeme použít libovolný výraz, jak je vidět u třetí křivky (v tomto případě se sloupce označují znakem dolaru a číslem). Třetí křivka tedy bude čtverec rozdílu prvních dvou vydělený deseti.

Fungují i některé speciální názvy na místě souboru. Pomlčka (`plot "-"`) říká, že se data očekávají ze stejného vstupu jako příkazy, tedy v tomto případě z klávesnice. Název začínající „většítkem“ (`<`) říká, že se spustí program uvedený za většítkem a jeho výstup se použije místo datového souboru.

`Gnuplot` umí kreslit i třírozměrné grafy. A to jak ve třírozměrném pohledu, tak jako dvojrozměrnou barevnou mapu. Tyto funkce se skrývají za příkazem `splot`, ale jejich popis je už nad možností tohoto článku.

Ať to nějak vypadá

Změnu popisku konkrétní křivky přidáním `title` do kreslicího příkazu už jsme uvedli. K tomu se sluší zmínit, že naopak `notitle` danou křivku z legendy vyjme. Celu legendou je možné různě posouvat, anebo ji kompletně schovat příkazy `set key`, resp. `unset key`. Popisky os nastavíme příkazem `set xlabel`, případně `set ylabel`. Zobrazení mřížky v grafu ovlivňuje příkaz `set grid`. Hodit se také může `set samples`, což je příkaz nastavující počet bodů, ve kterých se počítá graf funkce zadané výrazem. Pokud je tedy nějaká křivka moc kostrbatá, je namísto hodnotu zvýšit (výchozí je jen 100).⁸

⁷ Všimněte si možnosti rozdělit příkaz na více řádek, pokud je posledním znakem na řádce zpětné lomítko.

⁸ Ale příliš velké hodnoty mohou viditelně zpomalit vykreslování a hlavně neúměrně zvětšit výstupní vektorové obrázky.

U každé položky v příkazu `plot` můžete uvést, jakým způsobem se má kreslit. Klíčové slovo `linetype` (zkráceně `lt`) následované číslem určuje typ čáry. Konkrétní interpretace čísla závisí na výstupním zařízení. Podle možnosti barva čáry anebo různé styly čárkování. Standardně má čára v příkazu `plot` takové číslo `linetype`, kolikátá je v pořadí v tomto příkazu. Analogicky funguje parametr `pointtype` alias `pt`, který nastavuje styl kreslení bodů (třeba pro `w lp`). Šířku čáry upravíme parametrem `lw` alias `linewidth`, velikost značky bodu pak `ps` alias `pointsize`. K dispozici je také mnoho stylů za `with`, namátkou svislé či vodorovné „fousky“ označující chybu měření, sloupečky od vodorovné osy, stupínky apod.

Do grafu je možné vkládat popisky na libovolné místo příkazem `set label`. Příkazem `set format` je možné ovlivnit formát popisků osy. Na logaritmickou škálu přepneme osu příkazem `set log`.

Jedna ukázka na závěr:

```
gnuplot> set key above
gnuplot> set samples 1000
gnuplot> set xlabel "cas [s]"
gnuplot> set ylabel "amplituda"
gnuplot> F(f, t)=sin(t*2.*pi*f)
gnuplot> plot [0:2] F(5, x) t "5 Hz", F(6, x) t "6 Hz", \
> F(5, x)+F(6, x) lw 2 t "soucet", \
> 2.*F(.5,x-.5) lt 4 lw 2 t "vysledna amplituda", \
> 2.*F(.5,x+.5) lt 4 lw 2 notitle
```

Vytváříme obrázky, ukládáme

Když už jsme se na naše výtvořiny dost vynadávali, bylo by dobré, umět je také převést na formát vhodný k dalšímu zpracování (vlození do textu, tisku apod.). Gnuplot toto řeší nastavením výstupního souboru příkazem `set output` a výstupního formátu příkazem `set term`. Příkazem `set term` bez dalších parametrů získáme výpis dostupných výstupních formátů. Na výběr jsou jak bitmapové, tak vektorové, vcelku široká je možnost integrace s \TeX em.⁹

Ukážeme si jednoduchý výstup do obrázku ve formátu PNG:

```
gnuplot> set output "obrazek.png"
gnuplot> set term png
gnuplot> plot sin(x)
```

⁹ Pro výrobu obrázků vhodných pro \TeX je cest více, jedna ze zajímavějších je využít výstup ve formátu metapostu, který jde buďto vložit přímo do \TeX u anebo převést na PDF. Výhodou je (i když občas tato vlastnost umí být záladná), že všechny popisky jsou zpracovány \TeX em, takže si můžeme vyhrát dle libosti. Po nastavení vhodných formátů popisků os vypadá výstup téměř dokonale :), ale na podrobný návod zde bohužel není místo.

Interaktivní psaní příkazů je šikovné ve chvíli, kdy prohlížíme data anebo něco zkusíme a zkoumáme, ale finální sekvenci příkazů vedoucí k výslednému obrázku by bylo lepší mít někde uloženou. Pak ji můžeme dle potřeby dodatečně upravovat, znovu použít apod. Asi nejjednodušší cestou je vytvořit sekvenci příkazů v obyčejném textovém souboru a spustit gnuplot s přídatným parametrem – názvem tohoto souboru. Příkazy by měly obsahovat i nastavení výstupního formátu a výstupního souboru.¹⁰ Potom spuštěním gnuplotu docílíme okamžitého vytvoření obrázku bez potřeby další interakce. Někdy se hodí příkaz `save`, který uloží do souboru kompletní aktuální nastavení gnuplotu včetně posledního příkazu `plot`. Příkazem `load` docílíme naopak vykonání příkazů gnuplotu uložených v souboru, který specifikujeme jako parametr příkazu.

Pokud se vám nelíbí některé z výchozích nastavení gnuplotu, můžete si vytvořit vlastní soubor přenastavující požadované věci (bude tedy pravděpodobně obsahovat hlavně příkazy `set`), který uložíte na místo, kde jej gnuplot najde. V případě Linuxového systému to bude soubor nazvaný `.gnuplot`, v případě Windows pak `gnuplot.ini`. Umístěný musí být buďto v domácím adresáři, anebo v aktuálním adresáři platném při spuštění gnuplotu.

Fitujeme

Hodně zajímavou a užitečnou funkcí gnuplotu je prokládání dat zadanou funkcí, většinou nazývané „fitování“. Máme nějaká naměřená data a chceme je proložit funkcí, u které známe obecný tvar, ale ne číselné koeficienty. Například máme kádinku, do které kapeme kapky. Kádinku s kapalinou umíme zvážit, kapky umíme počítat. Chceme zjistit, jakou hmotnost má jedna kapka. Znamenáme hmotnosti pro několik různých počtů kapek v kádince a víme, že závislost hmotnosti na počtu kapek by měla být lineární. Ale neznáme přesně ani absolutní člen (tj. hmotnost kádinky – nezapomeňte, že každé měření je zatíženo nějakou chybou), ani koeficient úměrnosti (to je hledaná hmotnost jedné kapky). Pokud budeme mít v souboru `kapky` dva sloupečky dat, počet kapek a zjištěnou hmotnost, můžeme v gnuplotu psát

```
gnuplot> m(n)=m0+mk*n
gnuplot> fit m(x) "kapky" via m0,mk
gnuplot> plot m(x), "kapky"
```

Prvním příkazem zadefinujeme tvar hledané funkce – hmotnost kádinky m závisí na počtu kapek n tak, jak je uvedeno. Hmotnost kapky je mk a prázdné kádinky $m0$. Následující příkaz říká gnuplotu, aby našel takové hodnoty $m0$ a mk , aby celková odchylka mezi daty v souboru „kapky“ a funkcí $m()$ byla minimální. Po skončení nám gnuplot na obrazovku vypíše spočítané hodnoty i s odhadovanou chybou. Posledním příkazem jen výsledek zobrazíme.

¹⁰ Pokud máte rádi přeměňování výstupu programů, je samozřejmě možné nechat výchozí nastavení na standardní výstup a konkrétní soubor vybrat přeměňováním až při spuštění gnuplotu.

Gnuplot umí prokládat data libovolnou zadanou funkcí, což je vcelku silný nástroj, ale má své meze. Je dobré vědět, jak prokládání přibližně funguje.¹¹ Gnuplot vezme nějaké výchozí hodnoty hledaných parametrů (můžete mu je sami zadat, někdy je potřeba zvolit ručně počáteční hodnoty přibližně odpovídající datům). Spočte hodnoty prokládané funkce v bodech, kde jsou i naměřená data a v každém bodě určí druhou mocninou rozdílu funkční hodnoty a hodnoty ze souboru. Součet těchto čtverců rozdílů určuje „kvalitu“ proložení. Poté gnuplot zkusí o trochu změnit hodnoty parametrů a zjistí, jak se při tom změní suma odchylek. Najde takový směr změny parametrů, který odpovídá největšímu poklesu odchylky a tím směrem udělá malý krok (tj. změní hodnoty parametrů). Poté se cyklus opakuje dokud se nedostane do místa, kde krok jakýmkoliv směrem situaci zhorší.

Tento postup má základní slabinu ve věci zvané lokální minimum. Většina složitějších funkcí bude mít takové hodnoty hledaných parametrů, které jsou v nějakém malém okolí z hlediska odchylky nejlepší, ale pro úplně jiné hodnoty parametrů existuje mnohem menší odchylka. Nicméně gnuplot zkouší jen blízké hodnoty parametrů, takže nepřeleze „kopeček“ horších řešení, aby našel lepší řešení v sousedním „dolíku“. Proto je vhodné fit vizuálně ověřit vykreslením funkce a dat a případně zkusit ručně upravit hodnoty parametrů a spustit další fit.

Také je potřeba dát pozor, abychom pro fitování nezadali vzájemně závislé parametry. Pokud se budeme snažit fitovat funkci $a \cdot b \cdot x$ s parametry a a b , rozumného výsledku se nedočkáme, protože stejné řešení můžeme dostat libovolnou kombinací těchto parametrů. V podobných případech je potřeba sloučit vzájemně závislé parametry do jednoho. (Pak nám samozřejmě fit nedá odpověď na to, jaká je hodnota jednotlivých „složek“, ale takovou odpověď nemůžeme očekávat už z principu.)

Marble

Úloha 4.5 – Gnuplot (6b)

A) Fibonacci

Vykreslete pomocí gnuplotu graf 100 prvních Fibonacciho čísel (posloupnost můžete vytvořit i externím programem). Najděte pěknou jednoduchou analytickou funkci, která přibližně vystihuje (tuto část) Fibonacciho posloupnosti (můžete využít fitování v gnuplotu) a vytvořte graf (anebo více grafů), které rozumným způsobem ukazují, jak dobře tato funkce posloupnost vystihuje.

Jako řešení posílejte jak příslušné příkazy pro gnuplot, tak i obrázky, pokud je vytvoříte.

B) Odporový teploměr

Představte si následující experiment: Do vakua jsme umístili wolframový drát, který můžeme žhavit průchodem elektrického proudu. Drát je umístěný ve va-

¹¹ Následující popis je zjednodušený, pro přesný a podrobný popis se podívejte třeba do nápovědy gnuplotu.

kuu a chladí se jen zářením¹². Ztrátový výkon chlazení bude tedy splňovat Stefan-Boltzmannův zákon

$$P = \varepsilon \sigma S T^4,$$

kde S je plocha vlákna, T jeho termodynamická teplota, σ je Stefan-Boltzmannova konstanta a ε emisivita wolframu. Emisivita wolframu bude s dostatečnou přesností pro tepelné záření přímo úměrná termodynamické teplotě. Navíc nesmíme zapomenout na tepelné záření ze stěn vakuové komory, které dopadají zpátky na vlákno. Výsledný vyzářený výkon tedy bude

$$P = k (T^5 - T_0^5),$$

kde k je konstanta charakterizující vlákno (plocha, koeficient emisivity), T je teplota vlákna a T_0 teplota stěn vakuové komory.

Vlákno je zahříváno procházejícím proudem I a měříme na něm napětí U . Z naměřených dvojic I a U bychom rádi zjistili, jak vypadá závislost odporu wolframu na teplotě. Předpokládáme, že odpor půže v celém rozsahu teplot vystihnout výrazem ve tvaru

$$R(T) = R_0 [1 + \alpha(T - T_0) + \beta(T - T_0)^2 + \gamma(T - T_0)^3].$$

Odpor R_0 má vlákno za teploty T_0 a koeficienty α , β a γ určují teplotní závislost odporu (tedy to, co nás zajímá).

Za situace, kdy $T_0 = 293$ K, byly změřeny následující dvojice proudu a napětí:¹³

#	I [A]	U [V]				
4.	5.611	3.0905	7.1691	6.6622	9.4244	10.7185
4.	9.969	3.5887	7.5498	7.2937	9.7411	11.4316
5.	1.146	3.7689	7.6562	7.4823	10.144	12.2395
5.	7.449	4.5521	7.9009	7.9177	10.334	12.689
6.	0.473	4.9856	8.1375	8.326	10.533	13.111
6.	4.435	5.5276	8.4241	8.8325	10.794	13.6605
6.	6.478	5.8647	8.8754	9.703	10.869	13.8397
6.	9.925	6.4106	9.1319	10.1839		

Určete z těchto informací teplotní závislost odporu wolframu a k výsledným hodnotám nezapomeňte poslat i pěkné ilustrační obrázky a podstatné příkazy gnuplotu použité pro spočtení výsledků a vytvoření obrázků.

Malá nápověda: Při používání příkazu `fit` bude pravděpodobně potřeba gnuplotu poněkud napovědět. Uvědomte si, že koeficient k v sobě obsahuje Stefan-Boltzmannovu konstantu a bude to tedy velmi malé číslo. Také není nutné hledat výraz, který vystihuje přímo závislost napětí na proudu. Klíčové slovo `using` můžete v gnuplotu použít i u příkazu `fit` a vstupní data si díky němu předzpracovat do vhodnější podoby.

¹² Předpokládejme, že experiment je postavený tak, že ztráty vedením jsou skutečně zanedbatelné.

¹³ Tato data jsou k dispozici ke stažení na adrese <http://mam.mff.cuni.cz/vstuply/gnuplot.txt>

Výsledková listina

Poř.	Jméno	R.	Σ_{-1}	Úlohy							Σ_0
				r1	r2	r3	r4	r5	t6	+	
1.	Doc. ^{MM} Štěpán Šimsa	1.	154	5		4	2	2	8	3	24
2.	Dr. ^{MM} Karel Kraus	4.	60	5	1		2			0	8
3.	Mgr. ^{MM} Karel Král	4.	31	5	1	3	2			0	11
4.	Mgr. ^{MM} Vojtěch Kolář	4.	23			3	2			0	5
5–6.	Mgr. ^{MM} Michaela Kochmanová	3.	48	3		0	2			0	5
	Mgr. ^{MM} Vojtěch Miloš	4.	37	5		3	2			0	10
7.	Mgr. ^{MM} Kristýna Onderková	4.	20	5	0		2	1		0	8
8–10.	Mgr. ^{MM} Alena Harlenderová	2.	21	4			2			0	6
	Bc. ^{MM} Michaela Hubatová	3.	17								
	Bc. ^{MM} Kateřina Márová	4.	17								
11–12.	Mgr. ^{MM} Tomáš Pokorný	3.	27		5	0				0	5
	Bc. ^{MM} Matěj Kocián	3.	15	5		2	2			0	9
13.	Bc. ^{MM} Petra Vahalová	4.	14	4			2			0	6
14–16.	Mgr. ^{MM} Martina Bekrová	3.	30	4			2			0	6
	Mgr. ^{MM} Jan Škoda	3.	28								
	Bc. ^{MM} Lukáš Langer	4.	16	4			2			0	6
17.	Dr. ^{MM} Alena Bušáková	3.	94								
18.	Bc. ^{MM} Karel Tesař	4.	10	3						0	3
19–21.	Bc. ^{MM} Martin Holeček	4.	13								
	Jakub Štoček	3.	9				2			0	2
	Tomáš Trégner	2.	9								
22.	Mgr. ^{MM} Anna Chejnovská	3.	26	4			0			0	4
23.	Gabriela Kubíčková	3.	7	5			2			0	7
24–27.	Mgr. ^{MM} Filip Štědrný	3.	49	4			2			0	6
	Mgr. ^{MM} Kateřina Honzáková	4.	39								
	Mgr. ^{MM} Filip Hlášek	3.	28								
	Ondřej Mička	1.	6								
28–31.	Doc. ^{MM} Petr Pecha	3.	158				2			0	2
	Mgr. ^{MM} Alena Jurásková	2.	26								
	Bc. ^{MM} Barbora Böhmová	2.	12	2		1	2			0	5
	Marek Stehlík	3.	5								
32–34.	Mgr. ^{MM} Michal Husek	4.	25	1		0	3			0	4
	Mgr. ^{MM} Martina Vaváčková	4.	20								
	Daniel Šafka	3.	4								
35–36.	Pavel Kratochvíl	2.	8								
	Pavel Pejřimovský	3.	3				3			0	3

Poř.	Jméno	R.	\sum_{-1}	Úlohy						\sum_0
				r1	r2	r3	r4	r5	t6	
37–41.	Mgr. ^{MM} Zuzana Dočekalová	4.	36							
	Bc. ^{MM} Pavel Novotný	4.	18					2	0	2
	Bc. ^{MM} Barbora Šmídová	1.	10							
	Dominik Miketa	3.	2							
	Kristýna Zemková	2.	2		0	2			0	2
42–43.	Josef Klesa	2.	0							
	Simona Ondrčková	1.	0							

Sloupeček \sum_{-1} ve výsledkové listině je součet všech bodů získaných v našem semináři, \sum_0 je součet bodů získaných v aktuální první sérii.

Ve sloupci „R.“ je uveden ročník (přepočtený na čtyřleté gymnázium, minimální hodnota je první ročník). Pokud máte v tomto sloupci uvedeno špatné (nebo žádné) číslo, napište nám svůj rok maturity, a my si opravíme údaj v databázi. Sloupeček „+“ značí bonusové body udělované podle ročníku a součtu bodů za úlohy.

Tituly uvedené v předchozím textu slouží pouze pro účely M&M.

Adresa redakce:

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

Telefon: +420 221 911 235

E-mail: MaM@atrey.karlin.mff.cuni.cz

WWW: <http://mam.mff.cuni.cz>



Časopis M&M je zastřešen Oddělením pro vnější vztahy a propagaci Univerzity Karlovy, Matematicko-fyzikální fakulty a vydáván za podpory středočeské pobočky Jednoty českých matematiků a fyziků.