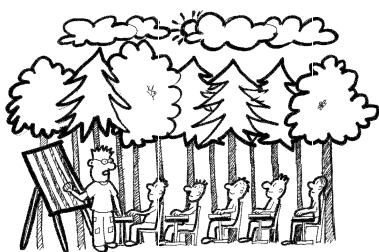


INFORMATIKA

Otazníky středoškolské informatiky

DALIBOR MARTIŠEK

Vysoké učení technické, Brno



Říká se, že ve využívání moderních technologií lze vystopovat dva základní směry: směr čistě uživatelský, „amatérský“, a směr „profesionální“. Středoškolská informatika dnes řeší problém, jak v této situaci vychovávat své matulanty, a musí si přitom klást řadu otázek. Na některé z nich se pokusím odpovédět z pozice vysokoškolského učitele.

1. Má informatika na výběr dva směry?

Uživatelský směr zacházení s moderní technologií lze demonstrovat třeba na příkladu mobilního telefonu. Mohu volat, posílat SMS-ky, fotografovat, nahrávat jednoduchá videa, ale nepotřebuji vědět, jak to funguje, je to pro mne černá skříňka. Telefon prostě používám. Pak se porouchá a nastoupí profesionál. Zjistí závadu, vymění klávesnici, (displej, přeinstaluje software.). Počítač je dnes totéž – spotřební elektronika, kterou mohu používat bez znalosti principů jeho činnosti. Psát dopisy, posílat poštu, vést účetnictví. Pak něco přestane fungovat a musí nastoupit odborník. Diagnostikuje a vymění vadnou komponentu, nakonfiguruje vzdálenou tiskárnu.

Zdá se, že i středoškolská informatika je nerozhodná v problému, zda vychovávat „amatéry“, tj. uživatele výpočetní techniky, anebo „profesionály“, tj. její údržbáře a opraváře. Podle přístupu k tomuto problému se

liší i skladba maturitních otázek. Uvedme příklad. Maturitní otázka zní (viz [1]):

Číslo vyjádřené ve dvojkové soustavě jako 0011 1011 0100 odpovídá číslu:

- A) 1674 v oktalové soustavě
- B) 958 v desítkové soustavě
- C) 1654 v oktalové soustavě
- D) 3B4 v hexadecimální soustavě

Zajímavější než otázka sama jsou reakce odpůrců a zastánců takto postavené maturity:

„Pro běžného uživatele (např. pro sekretářku, učitele při práci na své přípravě na hodinu, internetového surfaře, ...) postrádá tato úloha jakýkoli smysl. Při běžném používání počítače jsou tyto znalosti potřeba asi tolik, jako lékařský titul při poskytování první pomoci“ (viz opět [1]). Zastánci tohoto názoru propagují maturitu z informatiky, která je reprezentována úlohou: Vytvořte typograficky správnou pozvánku na oslavu narozenin a rozešlete ji e-mailem.

„Ten, kdo **maturuje** z informatiky, není **řadovým** uživatelem. Lze předpokládat, že pokud nebude pokračovat na vysoké škole, stane na nějakém nižším správcovském postu, či jako programátor“ (viz [2]). Hlasatelé těchto názorů vidí cíl v programovacích jazycích, serverech a síťových protokolech. Svědčí o tom některé aktuální maturitní otázky a reakce na ně.

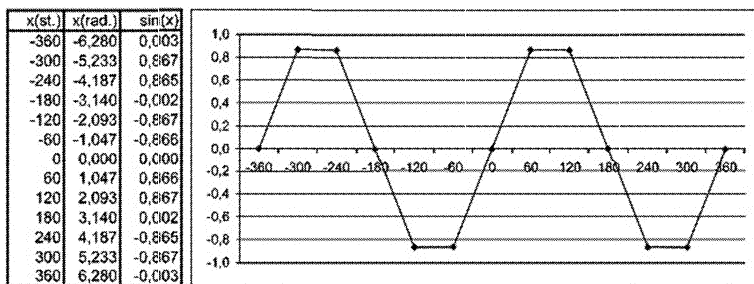
Takže vychovávat „běžného uživatele“, anebo správce – „opraváře“? Myslím, že ani jeden směr není správný. První z nich degraduje maturanta z informačních a komunikačních technologií (ICT) na někoho, kdo používá počítač jako každou jinou spotřební elektroniku. Mám-li použít výše uvedené zdravotnické přirovnání, pak by úkolem maturity na zdravotní škole bylo vychovat zdravotní sestru, která nemá dělat nic jiného, než poskytovat laickou první pomoc. Druhé tvrzení je ovšem taky špatné. Chce totiž naopak po této zdravotní sestře, aby byla uměla vyoperovat žlučník.

Co tedy? Mám za to, že správná by byla třetí cesta, která znamená, že by maturant měl být seznámen s internetem, s vybranými základními softwarovými produkty a s okruhem problémů, které se tím kterým produktem mohou řešit. Měl by chápat a být schopen vytvářet algoritmy vybraných úloh a měl by je umět realizovat pomocí vhodného SW. Měl by též nabýt základní znalosti a zkušenosti, které se uplatňují při „informatickém“ řešení různých (vybraných) problémů.

V souladu s tím bychom také měli naučit studenty kritickému pohledu na výpočetní techniku a seznamovat je s mezemi jejího použití jak po stránce technické, tak i softwarové. Problematika chybných grafů funkcí, kumulace zaokrouhlovacích chyb, sčítání řádově rozdílných čísel v pohyblivé čárce a dlouhých částečných součtů řad aj. je sice vousatá – některé problematické případy se objevovaly už v době prvních programovatelných kalkulaček, ale je stále aktuální. Je třeba, aby si studenti trvale uvědomovali hranice použití *výpočetní techniky* a meze použití daného *software*. I když jde o staré problémy, má je i současná technika a měli bychom s nimi studenty vhodným způsobem, např. pomocí vybraných příkladů, postupně seznamovat. V kapitole 4 probereme některé typické případy.

2. Kam se poděla krása poznání?

Tuto kapitolu začnu příkladem ze své praxe. Každoročně cvičím pět šest studijních skupin v prvním ročníku z počítačové grafiky. Loni to bylo asi sto lidí, z nichž *asi čtyřicet mělo maturitu z informatiky*. Na začátku výuky jsem zadal úkol: za hodinu chci na monitoru počítače vidět graf funkce sinus. Studenti měli k dispozici čtvero programovacích prostředí, dva systémy matematického software, kompletní instalaci MS Office a spojení s celým světem. Jeden student sestrojil „sinusoidu“ v MS Excel (viz obr. 1). Přesně tak ji prý v informatice dělali. Reakce na otázku, proč má křivku tak hranatou, mě vyděsila. Menší krok by prodloužil tabulku. A ta nemůže být delší – nevešla by se totiž na stránku a to by bylo typograficky špatně. Přezil jsem svůj první infarkt a řešení jsem uznal.



Obr. 1 Sinusoida v představách dnešního maturanta z informatiky

Dva studenti nakreslili na papír něco, co sinusoidu připomínalo jen vzdáleně, a sháněli se po skeneru. I toto „řešení“ jsem uznal. Další dva

našli křivku někde na internetu a jeden ji (světe, div se!) naprogramoval. Úroveň znalostí studentů byla neuvěřitelná, můj úkol zvládlo jen ŠEST lidí ze sta. I když připustíme, že šlo zřejmě o značně nekvalitní výběr uchazečů o studium, přesto notný díl neúspěchů musíme přičíst i na vrub výsledků výuky „informatiky“ na středních školách. Představa dnešních maturantů živících se programováním mi v tu chvíli připadala velmi komická.

Pak jsem studentům bez problémů vysvětlil racionální postup při řešení této úlohy a společně jsme ho zakrátko naprogramovali – je to prográmeček na několik řádků s jedním cyklem (viz dále). Po přidání ještě jednoho cyklu se pak sinusoida „rozvlnila“ a začala „pochodovat“, což u studentů vzbudilo nadšení a nad obyčejnou „živou křivkou“ jásali snad daleko víc než nad nějakými efektními triky počítačových her.

Asi poprvé v životě totiž vytvořili (i když s pomocí) něco opravdu sami. Snad poprvé v životě zažili krásu poznání. Tedy něco, co je *v obou výše uvedených případech* (a obávám se, že zřejmě v celé středoškolské informatice) většinou naprosto ignorováno.

3. Tlačit klavír k židli, nebo židli ke klavíru?

Další otazník dnešní středoškolské informatiky: ICT jsou většinou v duchu „uživatelského“ přístupu ztotožňovány s obsluhou počítače. Ba co hůře, s obsluhou softwaru firmy Microsoft. Znam střední školy, kde je produktům této firmy věnováno dva a půl roku, a to údajně proto, že jsou nejrozšířenější. Něco na tom jistě je, např. Excel může dobře posloužit u řady kancelářských prací a Word při psaní textů. Mělo by se však zůstat jen u základních činností s tím, že se to další může člověk naučit sám, až to bude potřebovat (bude-li vůbec), stejně jako se sám naučí třeba obsluhu mobilního telefonu značky Nokia, aniž by z toho musel maturovat.

Dnešní glorifikace kancelářského software může být prostým důsledkem toho, že na jiný software nemá škola peníze nebo učitele. Jde ovšem tak daleko, že se v něm provozují věci, které z něj lze sice jistým způsobem vymáchat, ale ke kterým se příliš nehodí. Na jedné straně je pochopitelné a přirozené, že např. někteří uživatelé Excelu se pokoušejí použít tento prostředek i na úlohy, pro které nebyl primárně vytvořen, např. na grafická znázornění různých jevů (viz např. [3], [5]). Na straně druhé bych ale očekával, že výsledkem těchto pokusů bude konstatování, že např. na vykreslování křivek Excel zrovna vhodný není (s jeho pomocí lze vykreslit jen některé grafy, a to jen přibližně a jen pro základní orientaci) a že

lze nalézt jednodušší, elegantnější a sofistikovanější řešení, třeba pomocí počítačového programu.

Např. pro zobrazení křivky lze výhodně využít univerzální proceduru *Krivka*, která vykresluje grafy velmi kvalitně a se kterou se naši studenti na počátku studia seznamují. Procedura hledá průsečnici grafu funkce $z = F(x; y)$ s rovinou $z = 0$, a to na principu znaménkových změn. Výsledkem je pak křivka daná rovnicí $F(x; y) = 0$. V následující ukázce je tato procedura vložena do programu *Kresli*. Za funkci F je dosazena levá strana implicitní funkce $y - \operatorname{tg} x = 0$.

```
program Kresli;

uses Graph, Crt;

const
  Sirka=640; Vyska=480;

var
  grDriver,grMode: Integer;

procedure NakresliBod(II,JJ: Integer);
begin
  {Zde se zadá nakreslení "bodu" o souřadnicích vlože-
   ných do parametrů II, JJ a s přihlédnutím k požadované
   tloušťce čar výsledné křivky, např. PutPixel(II,JJ,8)}
end;

procedure Krivka;
const X1=-8; X2=8; Y1=-6; Y2=6; {stejná měřítko na obou
                                  osách}

var
  X,Y,HX,HY: Real;
  I,J: Integer;

procedure VyznacOsy;
begin
  {Způsob zobrazení os závisí na tom, jaký komfort je při
   zobrazení grafu požadován. Někdy stačí jen "holé"}
```

```

    osy, avšak jen trochu zkušený programátor je může lehce
    doplnit tak, aby se tiskly jak kóty na osách (stupnice),
    tak i jejich textový popis. Zde zvolené hodnoty paramet-
    rů  $X_1$ ,  $X_2$ ,  $Y_1$ ,  $Y_2$  odpovídají poloze počátku souřadnico-
    vé soustavy uprostřed obrazovky a jednotce 40 pixelů.}
end;

```

```

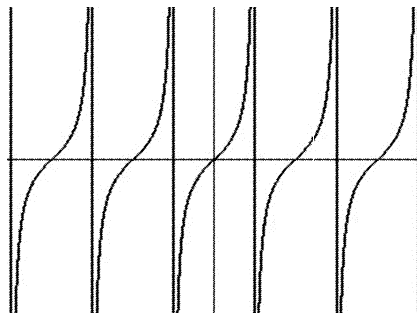
function F(X,Y: Real): Real;
begin F:=Y-Sin(X)/Cos(X) end;
{Sem se vloží levá strana anulované implicitní rovnice
zadané křivky.}

begin {Krivka}
    HX:=(X2-X1)/Sirka;
    HY:=(Y2-Y1)/Vyska;
    X:=X1; I:=0;
    VyznacOsy;
    repeat
        Y:=Y2; J:=0;
        repeat
            if (F(X,Y)*F(X+HX,Y)<0) or (F(X+HX,Y)* F(X+HX,Y+HY)<0)
            then
                NakresliBod(I,J);
                Inc(J); Y:=Y-HY
            until Y<Y1;
            Inc(I); X:=X+HX;
        until X>X2
    end; {Krivka}

begin {Kresli}
    ClrScr;
    InitGraph(grDriver,grMode,'');
    Krivka;
    ReadLn; CloseGraph
end. {Kresli}

```

Výsledkem práce procedury *Krivka* (programu *Kresli*) je tedy v tomto případě graf funkce tangens (obr. 2), neboť funkce F tak byla zadána. (Na obr. 2 je výstup grafu procedurou *VyznacOsy* doplněn jen „holými“



Obr. 2 Funkce $y = \operatorname{tg} x$

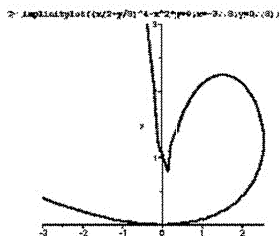
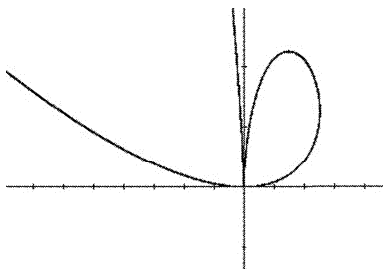
souřadnicovými osami.) Jako přídatek nám procedura *Krivka* zobrazila i vertikální asymptoty.

Proceduru *Krivka* lze samozřejmě využít i pro kreslení složitějších grafů, např. funkcí s implicitním zadáním

$$\left(\frac{x}{2} + \frac{y}{3}\right)^4 - x^2 y = 0.$$

Funkci F v proceduře *Krivka* zapíšeme

$F := \operatorname{sqr}(\operatorname{sqr}(x/2 + y/3)) - x * x * y.$



Obr. 3

Procedura *Krivka* pak vykreslí křivku (obr. 3a), která nejenže již není v možnostech Excelu, ale v této kvalitě ji nezvládne ani mnohý speciální a velmi drahý matematický software, jako je např. Maple 11 (obr. 3b). Při pohledu na výsledné obr. 3a a 3b si pak můžeme spolu se studenty

hrdě říci: „Dámy a pánové, instalace balíku Maple 11 do této učebny stála fakultu tři sta padesát tisíc. Jak sami vidíte, my jsme lepší!“ Značná nad-sázka, samozřejmě. Ale v danou chvíli silně motivující pravda. Další velmi důležitý a bohužel značně opomíjený aspekt vzdělávacího procesu!

4. Umění počítače počítat?

Tato otázka se vynořila s prvními „počítacími stroji“, od té doby o „chy-bách“ počítačů už hodně víme. Ve výuce jim ale bohužel nebývá věnována pozornost, přestože nepřipraveného uživatele mohou hodně překvapit. Při-pomeňme některé.

Umí Excel sečítat? V potřebném rozsahu ano, ale někdy to tak nemusí vypadat. Například nám může předvést „překvapující“ výsledek, že součet sedmi nul je 0,03 (obr. 4). Vysvětlení je přitom velmi prosté: v řádku 2 byl zřejmě zadán formát čísel, že v buňkách mají být čísla zaokrouhlena s přesností na 2 desetinná místa. Jestliže ve druhém řádku do buněk A až G vložíme 0,004, zobrazí se tato čísla zaokrouhlena na 0,00. Jejich součet je $7 \cdot 0,004 = 0,028$, takže v H2 se objeví tento výsledek zaokrouhlený na dvě desetinná místa, tedy 0,03. S takovými „nepřesnostmi“ způsobenými zaokrouhlováním na předepsaný počet desetinných míst musí prostě uživatel Excelu, ale i jiné výpočetní techniky, počítat.

	A	B	C	D	E	F	G	H
1								
2	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
3								

Obr. 4 Je $0 + 0 = 0$?

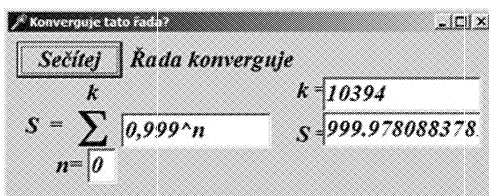
Je harmonická řada divergentní? Každý matematik odpoví: samozřejmě, že je. Každý počítač však řekne: samozřejmě, že není! Tady je nutno studenty důrazně varovat: nevěřte počítači! Nebezpečným klackem pod no-hama je zde *plovoucí desetinná čárka*. Při zjišťování součtu řady, tedy i její konvergence, je asi nejlepší model v této úvaze: Nechejme počítač sečítat (vytvářet částečné součty řady). Pokud se částečný součet přestane měnit,

řadu prohlásíme za konvergentní (a hrdě to oznámíme). Pokud částečný součet přeteče rozsah použitého typu (nebo se zacyklí), řada diverguje. Zjednodušený zápis tohoto algoritmu v pseudokódu je

```
S[0]:=0; n:=0;
repeat
    s[n+1]:=s[n]+a[n];
    n:=n+1;
until s[n]=s[n+1]
```

Zdánlivě nekonečný cyklus v tomto algoritmu nekonečný nebude. Projeví se totiž omezený počet platných cifer, a to u každého datového typu. U běžných typů bychom sice na výsledek čekali neúměrně dlouho, modelujeme-li však proces např. v pascalovských singlech, je to záležitost nejvýše minut (geometrická řada na obr. 5 skončí během vteřiny). Rovněž známá leibnizovská řada $4 = 4 + \frac{4}{3} + \frac{4}{5} + \frac{4}{7} + \dots = \pi$ potvrdí svou konvergenci. Podle typu stroje většinou během necelé minuty sečte 16 777 217 členů a poskytne výsledek s přesností na šest desetinných míst. Zdá se, že můžeme být docela spokojeni ovšem jen do okamžiku, než zadáme divergentní řadu harmonickou (se součtem $+\infty$). Dozvíme se totiž, že konverguje rovněž, a to dokonce daleko rychleji než předchozí řada. Po sečtení 2 097 153 členů je stanoven součet asi 15,4. Zneužitá plovoucí desetinná čárka se krutě pomstila. Na rozdíl od aritmetiky reálných čísel je totiž pro počítač konvergenční podmínka nejen podmínkou nutnou, ale rovněž podmínkou postačující.

Je sčítání komutativní? Tato otázka se může zdát již velmi podivná. Ale někdy i pravdu, která je zřejmá snad už každému prvňákovi, můžeme u počítače zahodit. Uvedme příklad z výpočtu částečných součtů harmonické řady. Miliontý částečný součet v pascalovském typu single počítaný zleva doprava je 14,357..., zprava doleva pak 14,392... Který je správný? Ani jeden. Víme, že většina sčítanců jsou pro počítač racionální zaokrouhlená čísla, a to postupně na nestejný počet platných číslic a zde si zaokrouhlování vybírá svou daň. V typu real se sice tyto výsledky sčítání



Obr. 5 Sčítání geometrické řady

zleva a zprava liší až na devátém desetinném místě a v extended se neliší vůbec, ale u jiných řad lze i v typu extended dostat součty zcela nesmyslné. Ukazuje se, že blíž skutečné hodnotě součtu je sčítání zprava, tedy „od nejmenšího“, kdy jsou sčítance „přesnější“ a zaokrouhlování začíná v nižších řádech. Proto při přesných technických výpočtech je třeba při sčítání mnoha členů věnovat velkou pozornost právě i *způsobu sčítání*.

5. Kudy na třetí nástupišť?

Vraťme se ke dvěma (extrémním) názorům na vzdělávání maturantů:

- Učme MS Office, protože to *dnes* potřebuje každý úředník!
- Ne, učme číselné soustavy, protože to *dnes* potřebuje každý správce sítě!

Oba tyto názory ženou vodu na mlýn cynikům, kteří se takové škole posmívají, že je jen přípravou na minulou válku. A mají bohužel pravdu. V duchu takto chápané „zásady spojení teorie s praxí“ strávila kdysi moje generace ve škole celý rok manipulací s různými typy logaritmických praviček.

Jeden moudrý aforismus praví, že vzdělání je to, co v člověku zůstane, až zapomene všechno, co se naučil ve škole. Ptejme se, co z informatiky zůstane v těch maturantech, kteří dále už informatiku nebudou studovat. Nic? To bychom určitě nechtěli. Chtějme a postupujme tak, aby v nich zůstaly pěkné vzpomínky na školu a takový vztah k informatice a výpočetní technice, že bez bázně i po letech si budou umět sednout k počítači jako k pomocníkovi při své práci i zábavě.

V 1. kapitole jsem zmínil, co by se mělo očekávat od maturantů z informatiky. Není to ani výchova uživatelů současného software, ani výchova provozovatele informačního systému. Podle mého názoru bychom se měli vydat třetí cestou. Obrazně řečeno: cílem výuky učně – truhláře není hoblík, ani znalost jeho údržby. Učni je třeba ukázat, jak se dělá polička. Ani v informatice nemohou být cílem textové a tabulkové procesory. Neučme ani číselné soustavy ani DNS servery, protokoly, neučme Pascal a neučme ani C#. Nevíme totiž dne ani hodiny, kdy budou tyto nástroje dneška následovat logaritmické pravítko do propadliště dějin.

To je sice pěkné, ale co tedy máme učit? Obecná rada je velmi jednoduchá: Učme *pomocí* textových editorů a tabulkových procesorů. Učme *prostřednictvím* číselných soustav a DNS serverů. Neučme programovat, ale vyučujme programováním. Pomocí nástrojů dneška ukazujeme, jak se zakousnout se do problému. Při práci se současnými softwarovými nástroji nezabíhejme do přílišných podrobností a raději učme studenty porozumět

problémům, vytvářet algoritmy, hledat výhody a slabiny různých metod řešení a schopnost najít nejlepší nástroj, který je momentálně k dispozici. Cílem by měl být vždy *užitečný výstup přiměřený potřebné kvalitě výsledku*, a to *užitím vhodného SW prostředku*. A učme prožívat radost z úspěchu. To je to, co vždycky hnalo, žene a bude hnát člověka dopředu včera, dnes, zítra i za sto let.

Jak tyto poněkud obecné myšlenky realizovat? Na to musí dát odpověď lidé vzdělaní v informatice a s dobrým rozhledem po nejrůznějších oborech. Myslím, že jich máme dost a že se odpovídat snaží. Jen je potřeba dobře naslouchat. Zatím se zdá, že středoškolská informatika dnes trochu zmateně pobíhá mezi dvěma nástupišti a hledá svůj vlak do stanice Zítřek. Jenomže oba vlaky, mezi kterými si tak dlouho a tak marně vybírá, jedou bůhví kam. Měli bychom všichni co nejdříve vyrazit k vlaku na nástupišti číslo tři, který míří do budoucnosti.

(Autorkou úvodní ilustrace je *Mgr. Jaroslava Čermáková* ze Suchých Lazců.)

L i t e r a t u r a

- [1] *Wagner, J.*: Parodie na maturitu, 3 část;
<http://www.ceskaskola.cz/Ceskaskola/AR.asp?ARI=102876&CAI=2124>
- [2] *Wagner, J.*: Parodie na maturitu – diskuse k článku;
<http://www.ceskaskola.cz/Ceskaskola/AR.asp?ARI=102876&CAI=2124>
- [3] *Matyáš, V.*: Užití Excelu při znázorňování limit funkcí, MFI, roč. 15 (2005/2006) str. 548–558.
- [4] *Martišek, D.*: Algoritmizace a programování v Delphi, Littera, Brno 2007.
- [5] *Matyáš, V.*: Posloupnosti a řady v Excelu, MFI, roč. 16 (2006/2007) str. 174–180.

Využití software MAPLE při výuce fyziky 2

JAN HRDÝ

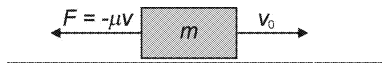
Přírodovědecká fakulta UP, Olomouc

(*Pokračování*)

Příklad 3

Po vodorovné rovině se pohybuje přímočaře počáteční rychlostí v_0 těleso o hmotnosti m , přičemž součinitel smykového tření je roven nule (těleso

se pohybuje např. na vzduchové dráze nebo na vzduchovém stole). Vlivem odporu vzduchu působí na těleso odporová síla $F = -\mu v$, jejíž velikost je přímo úměrná okamžité rychlosti tělesa v , a působí proti pohybu tělesa. Zjistěte závislost zrychlení tělesa a (a působící odporové síly F), okamžité rychlosti v a celkové dráhy s na čase (obr. 5).



Obr. 5 Pohyb tělesa pod vlivem odporu vzduchu

Na těleso o hmotnosti m , které se pohybuje setrvačností rychlostí v , působí jediná síla a tou silou je síla odporu prostředí F , pro kterou platí

$$F = -\mu v = ma. \quad (24)$$

Tato síla uděluje tělesu zrychlení a opačného směru, než je směr jeho rychlosti v

$$a = -\frac{\mu}{m}v. \quad (25)$$

Tento vztah je obdobou vztahu (3). Stejně jako v předcházejícím případě (4) tuto rovnici derivujeme podle času. Hlavně si musíme uvědomit, že se nejedná o pohyb rovnoměrně (zrychlený) zpomalený, protože ani působící síla a ani udělované zrychlení **nejdou konstantní**. Derivováním dostaneme

$$\frac{da}{dt} = -\frac{\mu}{m} \frac{dv}{dt} \quad (26)$$

A protože pro zrychlení platí následující vztah $a = dv/dt$, dostaneme po dosazení do předcházející rovnice a po následné separaci proměnných rovnici odpovídající tvarem a významem rovnici (6)

$$\frac{1}{a} da = -\frac{\mu}{m} dt. \quad (27)$$

Stejně jako v **příkladu 1** můžeme rovnici již přímo integrovat, protože jednotlivé proměnné jsou navzájem separovány. Dostaneme tak obdobu vztahu (8)

$$\ln a = -\frac{\mu}{m}t + a_0, \quad (28)$$

kde integrační konstanta a_0 má význam počáteční hodnoty zrychlení. Platí pro ni

$$a_0 = \frac{F}{m} = -\frac{\mu}{m}v_0. \quad (29)$$

Ze vztahů (28) a (29) potom pro výsledné zrychlení a platí

$$a = a_0 e^{-\frac{\mu}{m}t} = -\frac{\mu}{m}v_0 e^{-\frac{\mu}{m}t}. \quad (30)$$

Protože odporová síla F je v tomto jednoduchém případě přímo úměrná realizovanému zrychlení (obě veličiny jsou vzájemně pevně svázány číselnou konstantou m), dostaneme po vynásobení vztahu (30) hmotností m vztah

$$F = ma = ma_0 e^{-\frac{\mu}{m}t} = F_0 e^{-\frac{\mu}{m}t} = -\mu v_0 e^{-\frac{\mu}{m}t}, \quad (31)$$

neboť pro počáteční velikost síly F_0 platí

$$F = ma_0 = -\mu v_0. \quad (32)$$

Odporová síla F je však také přímo úměrná rychlosti tělesa v a proto můžeme odvodit ze vztahu (31) a s využitím relace (24) rovněž závislost rychlosti tělesa na čase

$$F = -\mu v = -\mu v_0 e^{-\frac{\mu}{m}t} \quad \text{neboli} \quad v = v_0 e^{-\frac{\mu}{m}t}. \quad (33)$$

Tuto závislost můžeme získat také využitím řešení určitého integrálu

$$v = v_0 + \int_0^t a dt = v_0 - \int_0^t \frac{\mu}{m} v_0 e^{-\frac{\mu}{m}t} dt = v_0 - \frac{\mu}{m} v_0 \int_0^t e^{-\frac{\mu}{m}t} dt \quad (34)$$

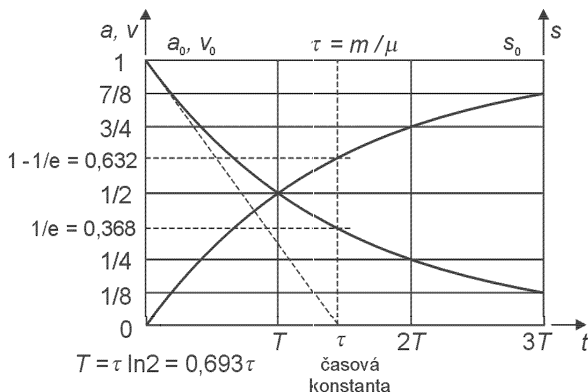
$$v = v_0 - \frac{\mu}{m} v_0 \left[-\frac{m}{\mu} e^{-\frac{\mu}{m}t} \right]_0^t = v_0 + \frac{\mu}{m} v_0 \frac{m}{\mu} e^{-\frac{\mu}{m}t} - \frac{\mu}{m} v_0 \frac{m}{\mu} = v_0 e^{-\frac{\mu}{m}t} \quad (35)$$

Tento způsob je také vhodný pro výpočet dráhy s , kterou urazilo sledované těleso za čas t

$$s = \int_0^t v dt = \int_0^t v_0 e^{-\frac{\mu}{m}t} dt = v_0 \left[-\frac{m}{\mu} e^{-\frac{\mu}{m}t} \right]_0^t = \frac{mv_0}{\mu} \left(1 - e^{-\frac{\mu}{m}t} \right). \quad (36)$$

Ze vztahu (36) je vidět, že v limitním případě, tj. pro t velké, urazí těleso celkovou dráhu s_0 , pro kterou platí

$$s_0 = \lim_{t \rightarrow \infty} s = \lim_{t \rightarrow \infty} \frac{mv_0}{\mu} \left(1 - e^{-\frac{\mu t}{m}}\right) = \lim_{t \rightarrow \infty} \frac{mv_0}{\mu} \left(1 - \frac{1}{e^{\frac{\mu t}{m}}}\right) = \frac{mv_0}{\mu}. \quad (37)$$



Obr. 6 Časový průběh velikosti okamžitého zrychlení pohybujícího se tělesa

I v tomto případě můžeme definovat **časovou konstantu** řešeného fyzikálního děje $\tau = m/\mu$, která má obdobný význam jako u předcházejících příkladů. Je to doba, během které poklesne velikost odporové síly na $e^{-1} = 1/e = 0,36787944 \approx 36,8 \%$ své počáteční hodnoty F_0 , je to též doba, za kterou poklesne okamžité zrychlení a nebo okamžitá rychlost v na stejnou hodnotu a také je to doba, za kterou urazí těleso $1 - e^{-1} = 1 - 1/e \approx 63,8 \%$ své celkové dráhy s_0 . Rovněž ji můžeme interpretovat tak, že je to doba potřebná k úplnému zastavení tělesa za předpokladu, že působící odporová síla má stále **konstantní velikost** F_0 .

Důležitá je rovněž doba T , za kterou poklesne velikost odporové síly F (nebo velikost okamžité rychlosti v nebo zrychlení a) na jednu polovinu výchozí hodnoty F_0

$$F = F_0 e^{-\frac{\mu T}{m}} = \frac{F_0}{2}, \quad (38)$$

z čehož plyne obdobně jako v předcházejícím příkladu, že $\mu T/m = \ln 2$ a tedy platí opět vztahy obdobné vztahům (14) a 15)

$$T = m \ln 2 / \mu = \tau \ln 2 \quad (39)$$

$$T \approx 0,693m/\mu = 0,693\tau. \quad (40)$$

Je zřejmé, že za dvojnásobek této doby $2T$ poklesne odporová síla (nebo zrychlení nebo rychlost) na jednu čtvrtinu, za trojnásobek $3T$ na jednu osminu své výchozí hodnoty atd., viz obr. 6. Za tuto dobu T těleso urazí polovinu své celkové dráhy $s = s_0/2 = mv_0/2\mu$, za dvojnásobek této doby $2T$ urazí tři čtvrtiny celkové dráhy $s = 3s_0/4 = 3mv_0/4\mu$, za trojnásobek této doby $3T$ urazí sedm osmin celkové dráhy $s = 7s_0/8 = 7mv_0/8\mu$ atd.

Příklad 4

Nechť radioaktivní látka obsahuje n_0 ještě nerozpadlých atomů. Potom se za dobu dt rozpadne $dn = -\lambda ndt$ atomů a počet nerozpadlých atomů se zmenší na $n = n_0 + dn = n_0 - \lambda ndt$, kde λ je tzv. **rozpadová konstanta**, která udává počet rozpadlých atomů za jednotku času vztažených k jednomu atomu, součin λn potom udává počet rozpadlých atomů za jednotku času vztažených k celkovému počtu atomů n a konečně součin λndt udává počet rozpadlých atomů za dobu dt vztažených k celkovému počtu atomů n .

Tento příklad je ze všech příkladů zde uvedených asi nejsnáze řešitelný, protože už základní vztah popisující tento problém

$$dn = -\lambda ndt \quad (41)$$

se dá snadno převést na hledanou diferenciální rovnici se separovanými proměnnými

$$\frac{1}{n}dn = -\lambda dt, \quad (42)$$

která je logickou obdobou dříve uvedených rovnic (6), (18) a (27). Po integraci (levou i pravou stranu rovnice integrujeme samostatně) dostaneme

$$\ln n = -\lambda t + n_0. \quad (43)$$

Po odlogaritmování dostaneme vztah pro závislost počtu rozpadávajících se atomů n na čase

$$n = n_0 e^{-\lambda t}, \quad (44)$$

kde integrační konstanta n_0 má význam počátečního počtu atomů (počet atomů v čase $t = 0$).

I v tomto případě můžeme definovat **časovou konstantu** řešeného fyzikálního děje $\tau = 1/\lambda$, která má obdobný význam jako u předcházejících

Velmi důležitá je rovněž doba T , která má tentokrát také vlastní název **poločas rozpadu** a vyjadřuje dobu, za kterou poklesne velikost ještě nerozpadlých atomů na jednu polovinu výchozí hodnoty n_0

Z toho plyne obdobně jako v předcházejících příkladech, že $\lambda T = \ln 2$ a tedy platí opět vztahy obdobné vztahům (14) a (15)

$$T \approx 0,693/\lambda = 0,693\tau. \quad (47)$$

The graph shows the decay of a quantity n over time t . The vertical axis is labeled n and has markings at $0, 1/8, 1/4, 1/2, 3/4, 7/8, 1$. The horizontal axis is labeled t and has markings at $0, T, \tau, 2T, 3T$. A solid curve starts at $(0, 1)$ and passes through $(T, 1/2)$ and $(\tau, 1/e)$. A dashed line connects $(0, 1)$ to $(T, 0)$. A vertical dashed line is drawn at $t = \tau$, and a horizontal dashed line is drawn at $n = 1/e$. The text $\tau = 1/\lambda$ is written above the vertical dashed line. Below the horizontal axis, the text $T = \tau \ln 2 = 0,693\tau$ is written, followed by "polčas rozpadu" and "časová konštanta".

5. Vytváření dynamických modelů s použitím programu FAMU-LUS

178

Zadáním problému: Při určování stáří rostlinných archeologických nálezů radiouhlikovou metodou se využívá skutečnosti, že v zemské atmosféře je v oxidu uhličitém kromě stabilních izotopů uhlíku $^{12}_6\text{C}$ a $^{13}_6\text{C}$ také malé množství nestabilního izotopu $^{14}_6\text{C}$. Množství tohoto izotopu v atmosféře je dlouhodobě konstantní, protože se neustále obnovuje působením kosmického záření. Rostliny využívají uhlík z oxidu uhličitého na stavbu svých těl. Za svého života proto obsahují konstantní množství izotopu $^{14}_6\text{C}$, tj. asi jeden atom $^{14}_6\text{C}$ na bilion atomů $^{12}_6\text{C}$ a $^{13}_6\text{C}$. V odumřelých zbytcích rostlin se ovšem jádra $^{14}_6\text{C}$ stále rozpadají, ale neobnovují se z atmosféry. A proto zastoupení těchto jader obsažených ve zbytku rostliny s časem klesá. Za rok se rozpadne $k = 0,000\,121$ část jader, tj. z každého milionu jader $^{14}_6\text{C}$ se rozpadne za rok 121 jader (odpovídající poločas rozpadu izotopu $^{14}_6\text{C}$ je asi 5 000 let). Určete grafickou závislost počtu nerozpadlých jader izotopu $^{14}_6\text{C}$ na čase. Při řešení úlohy předpokládáme, že počáteční počet atomů $^{14}_6\text{C}$ je 1 milion, konstanta $k = 0,000\,121$ a délka sledovaného časového období je 10 000 let.

Při řešení této úlohy bychom mohli v programu *Famulus* [9] sestavit následující model (obr. 8). Postup dalšího řešení (nastavení parametrů zobrazení a vlastní výpočet) zde vzhledem k všeobecné rozšířenosti a znalosti tohoto programu není dále rozebírán.

```

MODEL:
=====
----- promenne, konstanty, procedury a funkce -----
k = 0.000121
dt = 10
----- pocatecni hodnoty -----
t = 0
N = 1000000
----- model -----
dN = k * N * dt
N = N - dN
t = t + dt
IF t > 10000 THEN STOP END
=====

```

Obr. 8 Ukázkový model v programu FAMULUS

V tabulce 1 jsou pro přehlednost soustředěny podklady pro všechny zde vyřešené příklady, která slouží pro sestavování podobných dynamických modelů.

Tab. 1 Podklady pro sestavování dynamických modelů

Číslo řešeného příkladu	Vztah mezi diferenciály proměnných	Závisle proměnná (svislá osa)	Nezávisle proměnná (vod. osa)	Číslo výchozího vztahu	Číslo ilustrač. obrázku
1.	$di = -\frac{i_0}{\tau}dt = -\frac{i_0}{RC}dt$	$i := i + di$	$t := t + dt$	(6)	obr. 2
	$du = -\frac{u_0}{\tau}dt = -\frac{u_0}{RC}dt$	$u := u + du$			
	$dq = -\frac{q_0}{\tau}dt = -\frac{q_0}{RC}dt$	$q := q + dq$			
2.	$di = -\frac{i_0}{\tau}dt = -\frac{i_0}{RC}dt$	$i := i + di$	$t := t + dt$	(18)	obr. 4
	$du = \frac{u_0}{\tau}dt = \frac{u_0}{RC}dt$	$u := u + du$			
	$dq = \frac{q_0}{\tau}dt = \frac{q_0}{RC}dt$	$q := q + dq$			
3.	$da = -\frac{a_0}{\tau}dt = -\frac{\mu a_0}{m}dt$	$a := a + da$	$t := t + dt$	(27)	obr. 6
	$dv = -\frac{v_0}{\tau}dt = -\frac{\mu v_0}{m}dt$	$v := v + dv$			
	$ds = \frac{s_0}{\tau}dt = \frac{\mu s_0}{m}dt$	$s := s + ds$			
4.	$dn = -\frac{n_0}{\tau}dt = -\lambda n_0 dt$	$n := n + dn$	$t := t + dt$	(41)	obr. 7

Podobný způsob zápisu dynamického modelu do „připraveného formuláře“ používá také např. program Modeling, který je jednou ze čtyř základních součástí softwarového balíku Coach 5, viz také [10].

(Pokračování)