

Západočeská univerzita v Plzni

Fakulta pedagogická

Katedra obecné fyziky

Možnosti počítačové podpory výuky fyziky

Rigorózní práce

Plzeň 2007

Autor práce: Mgr. Pavel Masopust

*Prohlašuji, že jsem tuto rigorózní práci vypracoval samostatně a veškeré použité
prameny uvedl v seznamu použité literatury.*

V Plzni dne 13. 11. 2007

Pavel Masopust

Obsah

1	Úvod	1
1.1	Fyzika a specifika její didaktiky	1
1.1.1	Význam experimentu ve vyučování fyziky	2
1.2	Cíl práce	6
1.3	Obecné otázky využití výpočetní techniky ve výuce fyziky	7
1.3.1	Didaktické podmínky tvorby programového vybavení pro zpracování výsledků měření	8
2	Počítačové modely a simulace ve výuce fyziky	10
2.1	Pojem model	10
2.2	Klasifikace modelů	11
2.3	Model v pedagogice	11
2.4	Sestavení fyzikálního modelu	12
2.5	Numerické řešení matematického modelu	16
2.5.1	Eulerova metoda	17
2.5.2	Rungova–Kuttova metoda	17
2.6	Vizualizace výsledků modelu na počítači	19
2.6.1	Použití vývojového prostředí Codegear Delphi	20
2.6.2	Kyvadlo	26
2.6.3	Dvojkývadlo	27
2.6.4	Kyvadlo na kruhu	30
2.6.5	Kyvadlo na vozíku	31
2.6.6	Ohyb částic na štěrbině	33
2.6.7	Použití prostředí Adobe Flash	37
3	Využití programu Mathematica	42
3.1	Základy práce s programem Mathematica	44
3.2	Kompletní příklad I: Lissajousovy obrazce	53
3.3	Kompletní příklad II: Částice v jednorozměrné potenciálové jámě	54
3.4	Kompletní příklad I: 3-D zobrazení intenzity a potenciálu elektrického pole	58
4	Využití počítače jako měřicího přístroje	60
4.1	Popis současného stavu	60
4.1.1	Poissonovo rozdělení	61
4.1.2	Dolet beta částic	61
4.1.3	Zpětný rozptyl beta částic	62
4.2	Poissonovo rozdělení	62

4.2.1	Teorie úlohy <i>Poissonovo rozdělení</i>	62
4.2.2	Ovládání programu pro úlohu <i>Poissonovo rozdělení</i>	62
4.3	Dolet částic beta	66
4.3.1	Teorie úlohy <i>Dolet částic beta</i>	66
4.3.2	Ovládání programu pro úlohu <i>Dolet beta částic</i>	69
4.4	Zpětný rozptyl částic beta	73
4.4.1	Teorie úlohy <i>Zpětný rozptyl částic beta</i>	73
4.4.2	Ovládání programu pro úlohu <i>Zpětný rozptyl beta záření</i> . . .	75
4.5	Technické a programové řešení práce	78
4.5.1	Technické řešení práce	79
4.5.2	Programové řešení práce	84
4.5.3	Základy práce s obslužným software	88
5	Počítač jako prostředek motivace ve výuce fyziky	97
5.1	Pojem motivace	97
5.2	Didaktická hra — fyzikální piškvorky	99
6	Závěr	103

Abstract

Computers are very important part of our every-day life and are used almost everywhere, in every parts of human work. Computers are used also at the schools, in educational process. It would be very hard to find a school without computer classroom today.

Computers can be very helpful in every part of teaching process. This work talks about the computer utilization at the schools. We also refer to it as *computer-aided* education.

The experiments are necessary part of physics lessons. Sometimes we need to discuss the physics phenomenon, where we cannot provide direct (real) experiment. This is normal situation especially when we talk about the modern part of physics.

We have to manipulate with the huge objects (like in astronomy with stars), or, on the other hand with the tiny objects (elementary particles in quantum physics). For some special experiments we need to study systems under extremum conditions (high or low temperature, pressure, etc). Such experiments can't be showed directly. In this situation all these experiments can be showed on the computer screen. But computer simulations shouldn't completely replace the real physics experiment, students shouldn't lost the touch with the physics reality and material experiments.

Chapter 2 discuss the creation of computer physical simulations and models. There are variety of computer programming languages, that can be used for developing of computer simulations. Two of them (*Codegear Delphi* and *Adobe Flash*) are discussed here with the list of advantages and disadvantages.

Four practical examples are also described here: the physical system of *pendulum*, *double pendulum*, *Planck law of spectral radiance of electromagnetic field*, *bouncing ball*, *pendulum on the cart*, *pendulum on the circle*.

The second area where we can use the computer-aided tuition, is to use the computer as computing machine. These are after all the operations what the computers where designed for. So the third chapter of this work talks about the mathematical computer program — *Wolfram Mathematica* and it's usage in the schools.

The fourth chapter describes the computer as a helper for measure process. The main task of this chapter is to design and develop hardware and software equipments that bring new quality to physics education on Department of General Physics at University of West Bohemia, to be specific, to individual students measurement.

This hardware and software equipments speed up the value obtaining process and facilitate creation of final report of measurement.

The chapter describes the hardware part (the electronics circuit connected between personal computer and physics measurement instrument) and the software

part (computer program that communicates with the electronics circuit). Programs help measure three nuclear physics problems: Poisson distribution, Flying range of beta-particles and Back scattering of beta-particles.

Finally, the computer can help to motivate the students. Motivation can be increased by using didactic games. The last chapter describes the computer competition program, physical gomoku with physical questions and it's usage in physical education.

1 Úvod

Počítače hrají v našem životě a dnešní společnosti velmi významnou a v mnoha oblastech i nezastupitelnou roli. S počítači se lze nyní setkat i v oblastech, ve kterých to dříve nebylo myslitelné. Tento trend se nevyhnul ani oblasti vzdělávání.

S rozšířením výpočetní techniky souvisí i rozšíření počítačové sítě internet. Jen stěží bychom tak dnes hledali školu, ve které by nebyl studentům přístupný alespoň jeden počítač připojený k internetu. Důležitost výpočetní techniky ve školách si jistě uvědomuje i Ministerstvo školství, mládeže a tělovýchovy (MŠMT), a tak byl v roce 2002 v rámci programu SIPVZ (Státní informační politiky ve vzdělávání) spuštěn nejprve v testovacím a později v ostrém provozu projekt *Internet do škol*.

Projekt *Internet do škol* přinesl do několika tisíc českých škol prostřednictvím generálního dodavatele počítačové vybavení, internetovou konektivitu a intranetové služby. Ponechme stranou otázku o podezřelém financování projektu, samotném výběru dodavatele (nyní provozovatele) a spokojme se s faktem, že otázka počítačové vzdělanosti a dostupnosti výpočetní techniky není ministerstvu lhostejná. Osoby bez alespoň základních počítačových znalostí mají velmi ztíženou cestu k dalšímu vzdělávání a je to i pozdější handicap na trhu práce. Je tedy chvályhodné, že se základy výpočetní techniky vyučují již na základní škole.

Počítače mohou být velmi platnými pomocníky na mnoha místech vzdělávacího procesu. A právě otázkou možnosti a účelnosti aplikace výpočetní techniky ve výuce fyziky se zabývá tato práce.

1.1 Fyzika a specifika její didaktiky

Vyučovací předměty na všech typech škol mají směřovat k rozvíjení a zdokonalování osobnosti studenta. To platí samozřejmě bezesbýtku i pro vyučování fyziky, která k rozvoji osobnosti přispívá také zlepšením fyzikálního myšlení a rozvojem intelektuálních i manuálních dovedností, které svým praktickým využitím dalece přesahují obor fyziky. Dovedností, které využijí v dalším životě všichni, i ti, kteří se později vlastní fyzikou zabývat nebudou.

Pod pojmem *fyzikální myšlení* rozumíme speciální pohled na realitu, které se projevuje schopností vyhledávat v jevech jejich fyzikální podstatu, tj. rozpoznávat a určovat fyzikální veličiny a stanovit jednoduché vztahy mezi nimi. Impulsem pro myšlení je vznik problémové situace.

Mezi takto rozvíjené dovednosti patří především

- schopnost přemýšlet o přírodních jevech a lidské činnosti z hlediska hledání souvislostí mezi nimi,
- schopnost podrobného analyzování problému a jeho dekompozice na menší

části,

- rozvíjení matematických znalostí.

Pro důkladné fyzikální poznávání je při výuce vždy nevyhnutelné těsné propojení teoretického a experimentálního přístupu. Teorie a experiment mají vzájemně nezeměnitelné funkce. Pouze neustálá konfrontace, resp. kontrola teorie a experimentu garantuje stálé a rozvíjející se fyzikální poznání. Empirické údaje (získané různou formou experimentu) často slouží jako důkaz pro potvrzení platnosti různých fyzikálních hypotéz a teorií. Naproti tomu teorie slouží k popisu a vysvětlení naší empirie a zobrazení reality.

Ve výuce je nutné vždy hledat rovnováhu mezi oběma přístupy, protežování jedné z těchto složek vyvolá asymetrický fyzikální rozvoj osobnosti studenta. Bez empirických poznatků není možno žáka motivovat a dokázat, že naše teoretické poznatky nejsou pouze promyšlenou spekulací, ale exaktním obrazem experimentálně zobrazené reality.

Na druhou stranu přeceněním empirických poznatků nevybavíme studenta hlubšími poznatky potřebnými pro jejich pochopení. Fyzika se pak pro studenta stane zábavnou, ale nepochopenou hříčkou. Student bude pouze pasivní pozorovatel.

Teoretická složka je studentovi předkládána především prostřednictvím *slovních metod*. Sem patří popis, vysvětlování, přednáška a rozhovor. V rámci jedné vyučovací hodiny je účelné kombinovat prvky různých učebních metod. Mezi slovní metody řadíme i studentský referát a práci s knihou. Schopnost práce s knihou a zvládnutí syntézy informací z různých zdrojů je nepostradatelnou vlastností každého vzdělaného člověka.

Empirická složka je ve výuce fyziky realizovaná sledováním frontálních a demonstračních experimentů a dále (a hlavně) studentovou přímou činností při vypracování laboratorních úloh, fyzikálních praktik apod.

V reálném vyučovacím procesu dochází k prolnutí empirického a teoretického náhledu na fyziku. Např. při demonstračním experimentu může vyučující slovně popsat průběh experimentu, zdůvodnit získané výsledky či odkázat na dříve probíranou teorii.

1.1.1 Význam experimentu ve vyučování fyziky

Fyzikální experiment ve fyzice

- je prostředek pro získání poznatků o fyzikálních jevech,
- zvyšuje zájem studentů o fyziku, čímž značně usnadňuje osvojení učiva,
- pomáhá utvářet konkrétní představu o konkrétních fyzikálních pojmech,

- aktivizuje studenty, zvlášť pokud pokusy sami provádějí,
- rozvíjí fyzikální myšlení, pozorovací schopnosti i technické dovednosti studenta.

Pro účel této práce nemá smysl podrobně experimenty klasifikovat, postačí pouze letmá zmínka o *Demonstračním pokusu*, *Laboratorní práci* a *Fyzikálním praktiku*, kterými se jimi budeme blíže zabývat.

Demonstrační pokus Demonstrační pokus lze zařadit mezi metody *názorně demonstrační*. Ty se opírají o přímé (a pasivní) pozorování jevu, jsou důležité především v prvopočáteční fázi poznání, které často začíná právě prožitkem a vjemem.

J. Maňák zpracoval pro učitele obecné požadavky na zásady použití této metody ve výuce. Uvedl, že:

- Na předvádění je třeba předem naplánovat potřebné materiály, pomůcky a prověřit fungování technických zařízení.
- Zlaté pravidlo pro učitele, formulované J. A. Komenským, požaduje podle možností předkládat předměty co největšímu počtu smyslů, neboť jen tak je možno jevy poznat po všech stránkách.
- Složitější předvádění je nutné rozložit v jednotlivé jednodušší prvky.
- Předvádění má probíhat v přiměřeném tempu, má být přístupné všem žákům, kterým je určeno.
- Pokud to dovoluje charakter předváděných jevů, je účelné zapojit do předvádění žáky.
- Při předvádění žáci nemají být pasivní; proto učitel žáky aktivizuje ke spolupráci, podněcuje je k otázkám.
- Po jednotlivých fázích předvádění se osvědčuje prověřovat, zda bylo učivo pochopeno. Při nejasnostech nebo nepochopení je nutno obtížné prvky a části znovu předvést.
- Výsledek předvádění závisí mimo jiné také na tom, jak se předvádění vhodně a výstižně doplňuje slovním vysvětlováním.
- Po ukončení předvádění žáci sami shrnují hlavní poznatky. Nesprávnosti učitel opraví, nedostatky doplní.
- V průběhu předvádění si žáci mají pořizovat zápisky, zachycovat předváděné jevy v kresbách, v schématech, náčrtech apod. (Získat tuto dovednost je proces, který v prvopočáteční fázi musí učitel koordinovat a usměrňovat.)

- Každé předvádění názorného materiálu by současně mělo být výcvikem v pozorování.
- Předváděný předmět se musí žákům demonstrovat tak, aby ho mohli dobře vnímat.
- Názorniny je třeba předkládat žákům ve správnou dobu, tj. nikdy před začátkem výkladu.
- Žáci si hlouběji osvojí učivo, předvádí-li se jim jev ve vývoji, v činnosti, v souvislosti s reálnou skutečností, v přirozeném prostředí.

Jako demonstrační experiment obvykle označujeme fyzikální experimenty prováděné učitelem a pozorované celou třídou.

Tato forma výuky může být při správném provedení velmi účinná. Často se jedná o nejživější část hodiny, vhodně zařazena mezi náročný výklad může osvěžit prostředí ve třídě a přitáhnout zpět pozornost těkajících žáků.

Slouží pro prvotní seznámení studentů se studovaným jevem a umožňuje získat smyslové vjemy, bez nichž by bylo obtížné některé fyzikální pojmy prezentovat. Pojmy jako elektrický obvod, mechanický pohyb apod. lze jen obtížně vytvořit jiným způsobem než *názorně smyslovým jevem*. Je však i mnoho jevů, které nelze snadno demonstračním experimentem ukázat. Takové jevy, které mají příčiny v extrémních fyzikálních podmínkách: vysoké teploty, malé nebo naopak velké rozměry. Patří sem i situace pro jejichž řešení musíme zanedbat tření a odpor prostředí (např. úlohy z teoretické mechaniky). Avšak i pro tyto jevy lze použít demonstrační experiment formou počítačové animace nebo modelu.

Současná psychologie hovoří o pojmu *názorné myšlení*, jež je chápáno jako jedna z úrovní přetváření informace důležitá pro snazší zapamatování. I tato teorie potvrzuje význam demonstračního experimentu. Tomu pomáhá i fixace slyšené informace dalšími smysly než pouze sluchem. Navíc, při správném provedení a *dramatizaci* pokusu, má tento i nezanedbatelnou složku emotivní. To vše napomáhá snazšímu zapamatování učiva. Ostatně, tohoto faktu si byl vědom už Jan Amos Komenský, čehož dokladem budiž citát: „*Smyslem, rozumem, podáním možné dověděti se všeho. Smyslové prvotní a stálí vědy vůdcové. Všechno budiž pojímáno smysly, kolika jen můžeš.*“

Užití demonstračního experimentu ve výuce však nelze zužovat na pouhé prvotní seznámení s fyzikálním pojmem. Své uplatnění nalézá i při studiu *vlastností* fyzikálních objektů ale i v dalších fázích fyzikálního vyučování.

Mezi výhody demonstračního pokusu oproti samostatné žákovské práci řadíme:

- zaměřenost studenta na jediný objekt,

- student není rozptylován okolními jevy, např. nastavováním fyzikálních přístrojů,
- studentům lze ukázat i pokusy se speciálními a drahými pomůckami, případně provádět pokusy pro studenta nebezpečné,
- značná časová úspora.

Naproti tomu mezi nevýhody lze zařadit

- absenci bezprostředního styku studenta s experimentálním materiálem,
- určitou pasivitu studenta.

Ve vyučování fyzice není vždy možné provést všechny potřebné experimenty ať již pro nedostatečné technické vybavení, dříve uvedené fyzikální překážky, či omezenou hodinovou dotaci předmětu. Pak je možné vybrané experimenty předvést formou **počítačové simulace** či **počítačového modelování**. Využití a tvorbě počítačových modelu a simulací ve vyučování fyzice bude věnována kapitola 2.

Laboratorní práce a fyzikální praktikum Student úzce poznává fyzikální jev, navíc si osvojuje metody poznávání.

Výhodou je bezprostřední kontakt s měřenou úlohou. Poskytuje příležitost k opakování, prohloubení a rozšíření dříve získaných poznatků a dovedností.

Laboratorní práce, případně fyzikální praktika, jsou základní formou uplatnění metody praktických prací při výuce fyziky. Kromě rozumové znalosti musí student využívat i své motorické dovednosti. Poznávací činnost studenta je více komplexní a je syntézou vjemů z více smyslových orgánů. Takto získané vědomosti a dovednosti jsou hlubší a trvalejší.

Při samostatné práci se student také lépe seznámí s principy měřicích zařízení, měřicích metod, postupů atd.

Samostatná laboratorní práce klade vysoké nároky na studentovu samostatnost. Nutí ho překonávat různé obtíže které se v průběhu práce objeví. Rozvíjí pozorovací schopnosti, dovednost rozlišení podstatných a nepodstatných znaků, vyloučení náhodných činitelů a zobecnění změřených výsledků. To vše vede k přesnějším a pevněji zafixovaným poznatkům.

Laboratorní práce tedy přirozeně uplatňují základní didaktickou zásadu o spojení teorie a praxe tím, že přesvědčivě ověřují platnost teoretických poznatků získaných ve vyučovací hodině.

K intelektuálním dovednostem získaných při samostatné práci patří též potřeba práce s pomocnou literaturou, učí studenta pečlivosti při vypracování protokolu o měření, zlepšuje znalosti ovládání počítačových programů.

Student ve svém dalším životě využije také zkušenosti s obsluhou kancelářských programů (např. MS Office), které jsou ve fyzice využívány pro zpracování výsledků měření.

Laboratorní práce přináší nejen snazší pochopení a fixaci znalostí získaných v teoretické části výkladu, ale učí studenta také pečlivosti, zodpovědnosti a prezentování získaných výsledků při tvorbě protokolu o měření. Tento protokol bývá v drtivé většině případů vytvářen na počítači. Naměřené výsledky a další početní operace bývají prováděny v tabulkovém procesoru. Tomuto využití počítačů ve fyzice se tato práce věnovat nebude, dostatek informací o tabulkových procesorech je v odborné literatuře běžně k dispozici.

Složitější výpočty a matematické postupy, jimiž je fyzika často provázena, nemusí být ručně nebo v tabulkovém procesoru snadno řešitelné a v takových případech je vhodné použít specializovaný počítačový matematický program. Jedním takovým programem — programem Wolfram Mathematica — se bude zabývat kapitola 3.

1.2 Cíl práce

Tato práce si klade za cíl navrhnout a realizovat vzorové počítačové aplikace, které mohou být použity pro zkvalitnění výuky fyziky. Jmenovitě se bude věnovat následujícím oblastem:

- Použití počítače jako měřicího přístroje a zařízení pro zpracování výsledků měření z fyzikálního praktika, více kapitola 4.
- Použití počítače jako zařízení pro tvorbu a zobrazení počítačových modelů a simulací. Kapitola se bude zabývat též výběrem vhodného programovacího jazyka, nebo prostředí, ve kterém je vhodné simulace vytvářet. Více kapitola 2. Většina fyzikálních situací je popsána diferenciálními rovnicemi, nebo jejich soustavami. Proto se práce zabývá i několika metodami pro numerické řešení diferenciálních rovnic.
- Počítačový program Mathematica a jeho použití ve výuce fyziky. Více kapitola 3.
- Počítač jako prostředek zvyšování motivace studentů, fyzikální soutěže na počítači. Více kapitola 5.

Vytvořené programové vybavení a v dalším textu popsané zkušenosti s tvorbou fyzikálních pomůcek na počítači by měly být snadno použitelné a aplikovatelné i na další oblasti výuky fyziky a být cenným pomocníkem pro ty, kteří se rozhodnou svou výuku použitím počítačů oživit.

Práce je zaměřena především na výuku fyziky na vysokých školách, avšak některé výsledky mohou být beze změn aplikovány i na výuku na školách nižšího stupně. Ve výuce tak může být předvedena a kvalitativně popsána i většina ukázkových simulací a modelů v této práci vyvinutých.

Součástí práce je autorem navržený elektronický obvod zprostředkovávající komunikaci mezi měřicím přístrojem a osobním počítačem a ukázkové aplikace pro zpracování takto získaných hodnot.

Konkrétně je popsáno použití vytvořeného elektronického obvodu pro modernizaci měření fyzikálního praktika z oblasti atomové a jaderné fyziky. Vyvinut byl ovládací software pro úlohy:

- Poissonovo rozdělení,
- Dolet částic beta,
- Měření zpětného rozptylu částic beta.

1.3 Obecné otázky využití výpočetní techniky ve výuce fyziky

Výpočetní technika nachází ve vyučování stále větší uplatnění. Osobní počítače typu PC jsou snadno dostupné, a ovládat je umí naprostá většina studentů. Zařazení různých forem využití výpočetní techniky do výuky fyziky není proto limitováno ani nedostupností odpovídajícího výpočetního vybavení, ani neschopností studentů s ní pracovat. Obecně nalézáme základní 3 oblasti nasazení výpočetní techniky ve výuce. Jsou to využití jako:

Prostředek pro získání informací. Máme na mysli využití počítače jako prostředku pro přístup do různých forem informačních databází, např. prostřednictvím univerzitních počítačových sítí, nebo internetu.

Lze sem zařadit i použití počítače k výpočetním úkolům, tedy k práci, ke které byl původně navržen — počítání diferenciálních rovnic, různé matematické modely apod.

Prostředek pro práci s informací. Jedná se především o využití různých kancelářských programů — textových editorů, tabulkových editorů, programů pro tvorbu prezentací apod. Výhodou jsou i, jak už bylo výše zdůrazněno, získané zkušenosti v ovládání programů, které student zúročí v další odborné praxi.

Ovládací prvek. K počítači je možné připojit přístroje měřící fyzikální veličiny. Připojení je obvykle řešeno prostřednictvím sériového (COM, USB) nebo paralelního (LPT) portu, nebo formou přídatné PC karty (ISA, PCI, ...). Počítač

může fyzikální přístroj ovládat, případně pouze zaznamenávat přístrojem zjištěné hodnoty. Tato práce v kapitole 4 popisuje tvorbu elektronického zařízení pro komunikaci fyzikálního přístroje (čítače s Geigerovým–Müllerovým počítacem) a osobního počítáče.

Testovací a vyučovací programy. Pro významnost těchto programů byly zařazeny do zvláštní kategorie, ačkoli by je bylo možné vřadit do kategorie druhé.

Patří sem jak rozsáhlé databáze otázek, tak i monotematické testy sloužící k procvičení probírané látky a ke sledování úspěšnosti studenta v určitém časovém období.

Výhody užití výukových programů ve vyučování:

- aktivizace i slabších žáků,
- možnost individualizace tempa žáků,
- i slabší a pomalejší studenti jsou přinuceni každou tematiku probrat až do konce.

Pro zpestření výuky a motivaci studentů může být užita i soutěž s fyzikálními otázkami. V rámci této práce byla vyvinuta aplikace *fyzikální piškvorky*, která tomuto účelu dobře poslouží. Více viz kapitola 5.

1.3.1 Didaktické podmínky tvorby programového vybavení pro zpracování výsledků měření

Výsledkem každého měření je získání určitého vzorku dat. Cíl měření může být různý, v závislosti na typu experimentu který byl prováděn. Může se jednat např. o potvrzení jisté fyzikální teorie vyložené žákům v teoretické hodině, motivační úvod do dále probírané látky apod.

Měření jsou prováděna prostřednictvím fyzikálních přístrojů, které je nutné většinou manuálně obsluhovat. Při manuálním měření jsou údaje z měřicích přístrojů ručně zapisovány do předem připravených tabulek. Z těchto údajů jsou potom matematicky odvozovány obecnější zákonitosti, rýsovány grafy, ...

Požadavek na obslužný software měřicích přístrojů připojitelných k počítači je *suplovat* práci uživatele při pořizování hodnot, nahradit tuto zdlouhavou a často nudnou práci. Tuto část řešení popisované v této práci bezezbytku splňuje. Otázkou je, do jaké míry by měl počítač provádět **zpracování** naměřených výsledků.

Na rozdíl od získávání výsledků, které je mnohdy pouze zdržující, zdlouhavou činností (tedy činností která je s úspěchem ve všech oblastech lidské činnosti nahrazována stroji) vyžaduje zpracování výsledků bližší zapojení studenta. Mnohdy

předpokládá zamyšlení, osvojení jistých dovedností apod. Zpracování výsledků měření umožňuje studentovi hlubší proniknutí do řešeného problému, pomáhá pochopit praktické zákonitosti partie fyziky, již se měřené úloha týká.

Počítač by pro tento úkol neměl sloužit jako *tvůrce* měření, pouze jako *pomocník*. Pomoc chápeme v tomto kontextu opět jako odstranění nutnosti provádět zdlouhavou mechanickou činnost potřebnou např. pro vyčíslování složitých vzorců, práci s velkým počtem hodnot apod.

Lze si snadno představit program, který student pouze spustí a počká, než program získá dostatek hodnot pro tvorbu výsledného protokolu. Program, který bude fungovat zcela automaticky. Cílem samostatného měření studentů ovšem není získat *naměřené* hodnoty jak by tomu bylo např. při vědecké výzkumné práci, cílem je hlubší proniknutí do měřeného problému. Jedinou činnost, kterou by takový počítačový program vyžadoval, by bylo vyzvednutí vytisknutého protokolu z počítačové tiskárny. Student by nezískal dovednosti v obsluhování tabulkových kalkulátorů, nemusel by se zamýšlet nad smysluplností získaných výsledků, aktivně by nepracoval. Takový program by nebyl didakticky vhodný.

Výhodnější je použití poloautomatického programu. Takového, který provede sběr hodnot, případně základní zobrazení výsledků. Oprostí studenta časově náročného odečítání stovek hodnot, ale ponechá volnost v další práci s těmito hodnotami.

Z těchto důvodů obsahují vytvořené programy možnost exportu změřených dat i vykreslených grafů.

2 Počítačové modely a simulace ve výuce fyziky

Studium moderních věd i jakékoli technické praxe s sebou přináší nutnost pracovat s obrovským množstvím informací, principů a pojmů. Se studovaným předmětem navíc často nelze z různých důvodů pracovat přímo. Příčin může být mnoho: technická nedostupnost objektu, ekonomické překážky spojené s jeho získáním

Uplatnění modelů však není výsadou pouze přírodních věd, modely významně pronikají i do věd společenských — pedagogiky, psychologie, filozofie, ...

Množství různých aplikací modelů bohužel přináší značné terminologické problémy. V každé oblasti je pojem model chápán mírně odlišně. V další části proto bude pojem model stručně definován.

2.1 Pojem model

Používání představ, nebo myšlenek s charakterem modelů lze najít již ve starověku. Jako příklad lze uvést atomistický model hmoty představeným Demokritem z Abdér přibližně kolem roku 400 př. n. l. Z představy existence nejmenších částíček hmoty se snažil odvozovat různé vlastnosti látek. Další významný krok byl geocentrický model vesmíru zavedený Ptolemaiem. K současnému chápání pojmu model se ještě více přiblížil Galileo a Newton. Místo zkoumání originálního objektu, který byl příliš složitý, nedostupný, apod., zkoumal se objekt jednodušší, známý, který byl v jistých rysech původnímu objektu podobný, nebo měl podobnou funkci. Ze získaných poznatků modelu se usuzovalo na vlastnosti originálu. Model umožňoval poznání originálu, byť zprostředkované.

Samo slovo model pochází z latinského slova *modus*, *modulus*, v překladu vzor, míra či způsob. Středověké stavitelství používalo slova *model* pro poloměr sloupu, podle níž byly odvozovány veškeré rozměry stavby. Podobný smysl použití slova lze najít ve stavitelství i dnes. Lingvisty nyní uváděný původ slova je italský, ze slova *modello* — vzor, předobraz. Tento smysl je stále užit ve výtvarném umění, kde je slovo používáno pro označení předlohy k dílu, ať již živé nebo neživé. Význam převzala často i technická praxe — model je provedení nějakého předmětu, stroje, nebo stavby ve zmenšeném měřítku.

Pro potřeby této práce lze model definovat jako přírodní nebo umělý objekt, který je v nějakém vztahu se zkoumaným objektem, nebo nějakou jeho částí, a je schopen zastoupit studovaný objekt. Zkoumání modelu umožňuje získat zprostředkované poznání zkoumaného objektu samého. V tomto pojetí je tedy model vždy o něco chudší než originální zkoumaný předmět. Při tvorbě modelu se abstrahuje a schematizuje objekt, nepodstatné části jsou vypouštěny. Kvalita výsledného modelu značně závisí na této prvotní analýze. Vypuštění významné části může celý model znehodnotit. Takto definovaný model lze nazvat model skutečnosti. Např.

Rosenblueth a Wiener v [13] definují: „... model je nahrazení komplexního systému systémem, který je pokládán za jednodušší a o němž se předpokládá, že má jisté vlastnosti, které jsou shodné s těmi, které byly vybrány pro studium na originálním komplexním systému“.

2.2 Klasifikace modelů

Existuje větší množství různých rozdělení a popisu druhů modelů. Jako nejvhodnější z didaktického hlediska lze použít rozdělení uvedené v [13]. Modely jsou tříděny do tří skupin:

Modifikující — vzhledem k originálu se mění pouze měřítko, případně z kvantitativního hlediska nepodstatné detaily. Výsledky získané z modelu lze v širokém měřítku uplatnit u originálního předmětu.

Simulující — model napodobuje děje pouze navenek, jen formálně. Funkční podstata modelu může být zcela jiná než u originálu. Originál a model může být např. popsán formálně stejnými fyzikálními rovnicemi.

Ilustrující — obrazně znázorňují originální děj.

Formalizující — děje jsou popisovány matematickými znakovými systémy (matematické modely).

U reálných modelů lze nacházet znaky z více kategorií předešlého rozdělení. Počítačové modely popisované v této práci jsou pak především modely simulujícími, ilustrujícími a lze nalézt i znaky formalizující.

2.3 Model v pedagogice

Prvky modelování však lze nalézt v pedagogice, především v praktickém vyučování, již od počátků vědecké pedagogiky. Již u J. A. Komenského: „Moudrost dvojnásobně a trojnásobně vidoucí, která jednu věc v druhé vidí a jednu druhou mocně dokazuje, to je: O přemocném užívání synkritické (srovnávací) metody k vypatrávání věcí skrytých, k objasnění temných, k dokázání pochybných a uspořádání zmatených. Synkritickou metodou vyučování nazýváme onu, kde se věc přirovnává k nějaké jiné věci téže podoby, která výrazněji ukazuje své příčiny, části, výsledky a jiné své vlastnosti, takže to, co se ze sebe dosti nepoznává, poznává se z věci sobě podobné“ [15].

V terminologii moderní vědecké pedagogiky se lze s pojmem model setkat ve dvou různých významech. Často se hovoří o modelu jistého výchovného procesu, případně výchovné situace. Model je jakýmsi prostředkem vědeckého bádání, sloužící k získání

dosud nepoznaných vlastností studovaného pedagogického procesu. Narážíme tedy na model vyučování, model učení, model struktury učení, apod. Druhý význam slova model je označení materiálního předmětu, objektu sloužícího jako konkrétní vyučovací pomůcka.

V tomto duchu chápe pojem model i tato práce. Tedy ve smyslu podpůrného prostředku vyučovacího procesu, který má nějaký vztah ke zkoumanému objektu, má s ním shodnou funkčnost či jiné podobné vlastnosti. Rozdělení učebních pomůcek poskytuje didaktika. Rozeznáváme:

1. Přirozené objekty,
2. modely fyzikálních a technických objektů,
3. pomůcky pro demonstrace jevů
 - pomůcky pro demonstrace jevů dostupných přímému smyslovému vnímání,
 - pomůcky pro demonstrace jevů nedostupných přímému smyslovému vnímání,
 - pomůcky pro modelové pokusy.

Toto rozdělení pokrývá dostatečně potřeby této práce; pro použití v teoretické didaktice by bylo zřejmě nutné body 1. a 2. ještě dělit jemněji.

2.4 Sestavení fyzikálního modelu

Pvním krokem při použití počítačových simulací je vytvoření *matematického modelu*. Model může být sestaven z experimentálně zjištěných hodnot, nebo aplikací speciálního, pro tu kterou oblast fyziky určeného, matematického aparátu. Při tvorbě modelu reálné fyzikální situace musíme většinou přistoupit k jistým zjednodušením, aby byl model dostupnými technickými prostředky vůbec řešitelný. Chyby získaných hodnot však nesmí překročit únosnou hranici.

Snaha postihnout v modelu všechny vlastnosti reality by vnesla do modelu značnou složitost a ve výsledku by se jevila jako kontraproduktivní.

Modely v mechanice tak můžeme sestavit aplikací rovnic Newtonovy techniky, Lagrangeových či Hamiltonových rovnic a mnoha dalších. Modely při studiu elektromagnetického pole sestavíme z Maxwellových rovnic apod.

Pro získání matematického modelu musíme zvolit vhodnou metodiku řešení. Při volbě metodiky je nutné zohlednit mnoho hledisek. Metodika by pro řešení daného problému měla ostatní předčít v rychlosti výpočtu, přesnosti, rychlosti konvergence, spolehlivosti, stabilitě a mnoha dalších.

V dalším kroku pro zvolenou metodu vypracujeme algoritmus, tedy soubor několika instrukcí, které jsou k provedení výpočtu potřeba provést. Algoritmus je poté většinou zapsán do počítače v některém programovacím jazyku (C++, Object Pascal, ...). Problém je také možno řešit ve specializovaném matematickém balíku software (MatLab, Mathematica, atd.)

Předposledním krokem výpočtu je řešení vytvořeného matematického modelu zvoleným a vytvořeným algoritmem. Součástí řešení je i ověření smysluplnosti získaných veličin. Často se tak hodnoty které nalezneme aplikací modelu srovnávají s experimentálně získanými hodnotami a je provedena diskuze případných rozdílů nebo naopak shod. Zkoumá se tak vhodnost sestaveného modelu a algoritmu řešení (např. dostatečná přesnost, rychlost konvergence, a další).

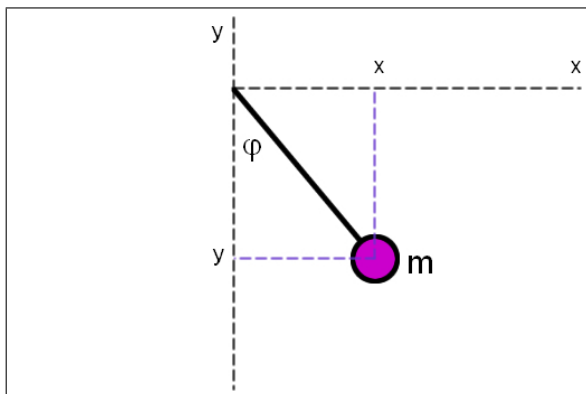
Posledním, avšak velmi významným krokem, který se může částečně překrývat s předchozím krokem, je vyhodnocení a prezentace výsledků. Získané výsledky je nutné vizualizovat a zpřístupnit tak dalšímu zkoumání. Výsledky je možné zpracovat ve formě tabulek (v některém z tabulkových procesorů), animovat časový vývoj veličin na obrazovce počítače, zobrazit závislost pozorovaných veličin formou grafů a podobně.

Často řešeným problémem ve fyzice je časový vývoj některé z vlastností fyzikálního systému. Tak např. při studiu trajektorie částice v mechanice hledáme časový vývoj souřadnic částice v prostoru, časový vývoj hybnosti, energie, ... Můžeme hledat časovou změnu náboje na kondenzátoru a nespočet dalších problémů. Pro časté opakování a podobnost těchto případů, uveďme na tomto místě příklad tvorby a řešení této situace.

Příklad Ukažme postup tvorby modelu na jednoduché fyzikální situaci — na kyvadle. Tuto reálnou fyzikální situaci budeme popisovat jako tzv. matematické kyvadlo. Zanedbáme hmotnost závěsu, závěs bude dokonale tuhý a těleso kyvadla budeme popisovat jako hmotný bod. Tento předpoklad značně zjednoduší popis situace a výsledky modelu budou stále v dobré shodě s realitou. Budeme tedy popisovat situaci pohybu hmotného bodu o hmotnosti m v gravitačním poli s tíhovým zrychlením g , hmotný bod (kyvadlo) bude udržovat stálou vzdálenost od určeného bodu v prostoru (závěsu o délce l). Situaci též zjednoduší to, že situaci nebudeme popisovat jako pohyb kyvadla v prostoru (popisovali bychom hmotný bod se dvěma stupni volnosti), ale pouze v ploše.

Pozn.: Stupeň volnosti udává počet nezávislých veličin, jež jednoznačně popíší stav systému. V případě prostorového kyvadla bychom pracovali se systémem se dvěma stupni volnosti (dva úhly), pro kyvadlo, jehož pohyb je vázán k ploše, pracujeme pouze s jedním stupněm volnosti (jeden úhel).

Pro tvorbu matematického modelu použijeme Lagrangeovy rovnice druhého druhu.



Obrázek 1: Příklad sestavení matematického modelu reálné fyzikální situace — kyvadlo

Zobecněnou souřadnicí zvolíme úhel φ , který svírá osa y se závěsem — obr. 1. Nejprve nalezneme kinetickou energii kyvadla. Kartézské souřadnice hmotného bodu ze zobecněné souřadnice vyjádříme jako

$$\begin{aligned}x &= l \cdot \sin(\varphi) \\ y &= -l \cdot \cos(\varphi)\end{aligned}$$

Pro vyjádření okamžité rychlosti hmotného bodu nalezneme časové derivace jednotlivých souřadnic jako

$$\begin{aligned}\dot{x} &= l \cdot \cos(\varphi) \cdot \dot{\varphi} \\ \dot{y} &= l \cdot \sin(\varphi) \cdot \dot{\varphi}\end{aligned}$$

a rychlost jako

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} = l^2 \cos^2 \varphi \cdot \dot{\varphi}^2 + l^2 \sin^2 \varphi \cdot \dot{\varphi}^2 = l^2 \cdot \dot{\varphi}^2$$

Kinetická energie je pak $T = \frac{1}{2} \cdot m \cdot v^2$, tedy $T = \frac{1}{2} \cdot m \cdot l^2 \cdot \dot{\varphi}^2$.

Jako hladinu nulové potenciální energie zvolíme osu x a potenciální energii vyjádříme jako $V = -m \cdot g \cdot y$. Lagrangeovu funkci systému sestavme jako rozdíl kinetické a potenciální energie: $L = T - V = \frac{1}{2} \cdot m \cdot l^2 \cdot \dot{\varphi}^2 + m \cdot g \cdot l \cdot \cos(\varphi)$. Časový vývoj jednotlivých zobecněných souřadnic systému pak nalezneme aplikací podmínky

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = 0,$$

kde q značí zobecněnou souřadnici (úhel φ) a $\dot{q}$ zobecněnou rychlost ($\dot{\varphi}$ — úhlová

rychlost). Lagrangeova rovnice po dosazení vyjde

$$\frac{d}{dt} (m \cdot l^2 \cdot \dot{\varphi}) + m \cdot g \cdot l \cdot \sin(\varphi) = 0$$

$$\ddot{\varphi} + \frac{g}{l} \cdot \sin(\varphi) = 0.$$

Matematický model matematického kyvadla je tedy obyčejná diferenciální rovnice druhého řádu. Pokud chceme tuto rovnici analyticky řešit a spokojíme se s oblastí malých rozkvyů kyvadla, můžeme položit $\varphi = \sin \varphi$. Do jaké míry je tato volba oprávněná uvádí tabulka 1 — porovnání procentuálního rozdílu mezi φ a $\sin \varphi$. Vidíme tak například, že jednoho procenta rozdílu dosáhnou oba výrazy již pro úhel 15° .

Úhel (x)	Sin(x)	x [rad]	Rozdíl [%]
0	0,0000	0,0000	0,00
1	0,0175	0,0175	0,01
2	0,0349	0,0349	0,02
3	0,0523	0,0524	0,05
4	0,0698	0,0698	0,08
5	0,0872	0,0873	0,13
7	0,1219	0,1222	0,25
9	0,1564	0,1571	0,41
10	0,1736	0,1745	0,51
15	0,2588	0,2618	1,15
20	0,3420	0,3491	2,06
30	0,5000	0,5236	4,72
40	0,6428	0,6981	8,61
50	0,7660	0,8727	13,92
80	0,9848	1,3963	41,78
90	1,0000	1,5708	57,08

Tabulka 1: Porovnání procentuálního rozdílu $\sin(x)$ a x

Jedna s počítačových simulací vytvořených jako součást této práce ukazuje jaký rozdíl v pohybu kyvadla přinese toto zjednodušení.

Řešení diferenciální rovnice je snadno naleznutelné v každé učebnici matematiky a výsledkem je rovnice

$$\varphi = \varphi_0 \cdot \cos\left(\sqrt{\frac{g}{l}} \cdot t\right),$$

kde φ_0 je amplituda kmitavého pohybu vystupující zde jako počáteční podmínka. Pohyb kyvadla je tedy harmonický s frekvencí a periodou

$$f = \frac{1}{2 \cdot \pi} \cdot \sqrt{\frac{g}{l}}$$

$$T = \frac{1}{f} = 2 \cdot \pi \sqrt{\frac{l}{g}}.$$

Toto řešení se shoduje s experimentálními výsledky a naší praktickou zkušeností — se zvětšující se délkou kyvadla se zvětšuje doba kyvu.

Situace, kdy je možné matematický model řešit analyticky je však spíše výjimečná a k řešení musíme použít některou z *numerických metod*.

2.5 Numerické řešení matematického modelu

Analytické metody a způsoby řešení matematického modelu většinou poskytují spojitě řešení na předem známé oblasti, číselné řešení poskytuje řešení jen v omezeném počtu diskretních bodů. Numerická řešení bývají zatížena chybou (např. kvůli diskretnímu charakteru řešení nedávají tyto informaci o chování řešení mezi jednotlivými diskretními body). Numerické řešení tedy v praxi představuje pouhou aproximaci řešení přesného (analytického).

Nutnost numerického řešení vyvstává i v případech, že analytické řešení je příliš složité, např. použitím vyšších funkcí (Besselovy apod.), nebo pokud je řešení popsáno formou nekonečné řady a pro zpřesnění výpočtu je nutné nasčítat velké množství jejich členů. Na tyto problémy věda narážela i v minulosti a tak rozvoj numerických metod šel ruku v ruce s rozvojem automatizovaných počítačích strojů.

Zmiňme se nyní krátce o používaných numerických metodách.

Uvažujme pro jednoduchost systém s jedinou stavovou proměnnou x . Závislost pak může vypadat takto:

$$\dot{x} = f(t, x) \tag{1}$$

Řešením vývoje systému je trajektorie $x = x(t)$. K řešení problému je obvykle známo řešení v čase t_0 , $x_0(t_0) = x_0$, úloha se pak někdy nazývá *počáteční úlohou*.

Základem numerického řešení je obvykle diskretizace proměnných. Spojité řešení (jež bychom obdrželi analytickým řešením) je nahrazeno řadou diskretních bodů $x_0(t_0), x_1(t_1), x_2(t_2), x_3(t_3), \dots$, kde $t_0 < t_1 < t_2, \dots$

Jednotlivým hodnotám nezávislé proměnné (času t) odpovídají takzvané *uzly sítě*. Pakliže je síť dostatečně hustá, lze diskretní řešení považovat za vyjádření fázové trajektorie. Vzdálenost mezi jednotlivými body sítě je:

$$h_i = t_{i+1} - t_i,$$

h_i je takzvaný krok metody. Metody se stálou velikostí kroku nazýváme *ekvidistantní*.

Vhodnou volbou kroku metody ovlivňujeme přesnost metody ale i rychlost vý-

počtu. Nový stav systému (tedy novou hodnotu x_i) odvozujeme ze stavu v předchozích okamžicích $(t_{i-1}, t_{i-2}, \dots, t_{i-k})$. Hodnota k je důležitou charakteristikou metody (říká nám, jak daleká budoucnost ovlivňuje přítomný stav systému) a metoda se podle hodnoty k nazývá jednokroková, dvoukroková, ...

2.5.1 Eulerova metoda

Jedná se o nejjednodušší, jednokrokovou metodu. Přibližnou hodnotu řešení v bodě x_{i+1} nalezneme pomocí Taylorova rozvoje v bodě x_i .

$$x_{i+1} \approx x_i + \frac{dx(t_i)}{dt}(t_{i+1} - t_i) + \frac{1}{2} \frac{d^2x(t_i)}{dt^2}(t_{i+1} - t_i)^2 + \dots$$

Eulerova metoda je lineární aproximací tohoto rozvoje a používá ekvidistantní síť ($h = (t_{i+1} - t_i)$), takže jsme dosazením vzorce (1) pro Eulerovu metodu našli:

$$x_{i+1} = x_i + hf(t_i, x_i).$$

Chyba jednoho kroku metody je úměrná h^2 . Počet kroků v určitém intervalu $< a, b >$ je $N = \frac{b-a}{h}$, celková chyba je pak úměrná:

$$\epsilon \sim N \cdot h^2 \sim \frac{b-a}{h} \cdot h \sim h$$

Pro použití této metody je potřeba volit velmi krátký krok, jinak může chyba přerůst tolerovatelné meze.

2.5.2 Rungova–Kuttova metoda

Jedná se o v praxi daleko více využívanou metodu s lepšími vlastnostmi než metoda Eulerova. Uvažujeme-li opět rovnici (1), tedy

$$\dot{x} = f(t, x),$$

rozumíme Rungovými–Kuttovými vzorci zpravidla tyto vzorce:

$$\begin{aligned} x_{i+1} &= x_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 &= f(t_i, x_i), \\ k_2 &= f(t_i + \frac{h}{2}, x_i + \frac{1}{2}hk_1), \\ k_3 &= f(t_i + \frac{h}{2}, x_i + \frac{1}{2}hk_2), \\ k_4 &= f(t_i + h, x_i + hk_3). \end{aligned}$$

Kvůli významnosti této metody zde uvedme praktický příklad.

Příklad užití Rungovy–Kuttovy metody Uvažujme řešení diferenciální rovnice

$$y' = x \cdot y^2 - y, y(0) = 1$$

a zajímáme se o hodnotu pro $x = 0,3$. Zvolme $h = 0,3$. Nalezněme koeficienty k :

$$k_1 = 0 - 1 = -1$$

$$k_2 = 0,15 \cdot (0,85)^2 - 0,85 = -0,74163$$

$$k_3 = -0,77027$$

$$k_4 = 2 \cdot (4)^2 - 4 = -0,59155$$

$$y_1 = 0,769233.$$

Analytické řešení je

$$y = \frac{1}{1+x},$$

přesná hodnota pro $x = 0,3$ je $y(0,3) = 0,769231$. Vidíme tedy, že i při tak obřím (z hlediska numerické matematiky) zvoleném kroku (0,3) se oba výsledky liší až na šestém desetinném místě.

Popišme nyní postup numerické integrace našeho modelu matematického kyvadla.

Numerická integrace modelu matematického kyvadla Ve shodě s předchozím je matematické kyvadlo popsáno diferenciální rovnicí

$$\ddot{\varphi} + \frac{g}{l} \cdot \sin(\varphi) = 0.$$

Pro řešení této rovnice rozdělíme časovou osu na diskrétní „dílky“ Δt a předpokládejme znalost řešení v čase t_0 , počáteční podmínky — známe počáteční úhel odklonu kyvadla φ_0 a počáteční úhlovou rychlost $\dot{\varphi}_0$. Ze znalosti počátečního úhlu můžeme určit zrychlení hmotného bodu představujícího kyvadlo jako

$$\ddot{\varphi}_0 = -\frac{g}{l} \cdot \sin \varphi_0.$$

Okamžitou úhlovou rychlost v časovém okamžiku $t_0 + \Delta t$ zjistíme jako

$$\dot{\varphi}_{t_0+\Delta t} = \dot{\varphi}_0 + \ddot{\varphi}_0 \cdot \Delta t$$

a konečně „nový“ úhel v časovém okamžiku $t_0 + \Delta t$ bude

$$\varphi_{t_0+\Delta t} = \varphi_0 + \dot{\varphi}_{t_0+\Delta t} \cdot \Delta t.$$

Vidíme, že úhel φ_n v časovém okamžiku t_n je určen hodnotou úhlu a úhlové rychlosti v časovém okamžiku t_{n-1}

$$\varphi_n = \varphi_{n-1} + \dot{\varphi}_{n-1} \cdot \Delta t.$$

Výše popsaný způsob je jeden z nejjednodušších postupů pro numerické řešení diferenciální rovnice a je někdy též nazýván *Eulerovou časovou integrací*.

Uvedme některé výhody této metody:

- Je velmi snadno implementovatelná na počítači.
- Může snadno a rychle poskytnout první („hrubší“) pohled na řešení a hned v počátcích odhalit chybnou volbu modelu.

A jako nevýhody lze jmenovat:

- pro dostatečnou přesnost je potřeba zvolit velice malý interval Δt , což přinese vysokou výpočetní náročnost. Pro vyšší přesnost je potřeba provést mnoho výpočetních kroků.
- Chyba vnesená do jednoho místa řešení se dále „vleče“ s řešením a ovlivňuje všechny pozdější výpočetní kroky.
- Není možné určit hodnotu v libovolném časovém okamžiku, je nutné provést všechny předcházející kroky postupně.

I přes výše uvedené nevýhody může mít použití této metody opodstatnění např. v situacích kdy nepotřebujeme příliš vysokou přesnost a řešený problém není příliš složitý. Metoda pak může posloužit jako první náhled na získané řešení.

2.6 Vizualizace výsledků modelu na počítači

Vizualizace výsledků je velmi důležitým krokem uzavírající životní cyklus tvorby modelu ve fyzice a i jinde. Získané výsledky je nutné příslušně kategorizovat, sjednotit a umožnit tak přístup k výsledkům naší práce i dalším zájemcům o problematiku.

Výsledky je možné zpracovat např. formou počítačové prezentace pokud se předpokládá představení výsledků modelu na veřejnosti, vytvořením tabulek a grafů, animací výsledků na počítači, tvorbou článků do odborných periodik, atd. Prezentovaným výsledkům by neměla chybět diskuze věrnosti a důvěryhodnosti získaných

dat s přihlédnutím ke zjednodušením která jsme použili při tvorbě modelu a jeho matematickém řešení.

Protože se tato práce zabývá možnostmi využití počítačů ve výuce fyziky, kde by neměl vizuální materiál chybět, zaměříme se nyní podrobněji na prezentaci výsledků formou počítačové animace.

2.6.1 Použití vývojového prostředí Codegear Delphi

Vývojové prostředí Codegear Delphi (dříve Borland Delphi) navazuje na úspěšný a velmi používaný programovací jazyk Pascal. Ačkoli byl tento jazyk vytvořen pro výuku programování, jeho snadné osvojení a poměrně velká svoboda ve vyjadřování jej předurčila i pro široké využití praktické. Programy vytvořené v jazyce Pascal byly určené pro běh v nyní již překonaném operačním systému MS-DOS.

Firma Borland se svou verzí Delphi rozšířila možnosti jazyka Pascal a významnou měrou zjednodušila koncept vývoje aplikací založených na systému Windows. V programu Pascal existuje mnoho napsaného kódu (patří sem i fyzikální modely které lze nalézt např. na internetu) a na dnešních moderních počítačích je až na výjimky spuštění těchto již existujících programů problematické, leckdy i nemožné. Mezi operačními systémy MS-DOS a Windows se změnila podpora práce s periferními zařízeními (myš, tiskárna, ...), práce s grafikou, změna používaných grafických formátů, atd.

Autoři existujících aplikací často své programy do moderního prostředí Windows nepřevádějí s odůvodněním, že tvorba programů pro Windows je složitá, je nutné se učit mnoho nových věcí, apod. Částečně je to pravda, avšak programy ve Windows přinášejí mnoho výhod oproti starým programům v MS-DOS. Za všechny jmenujme především:

- Jednotné uživatelské rozhraní.
- Dopředná (alespoň doufejme) kompatibilita.
- Po osvojení základů programování v systému Windows je mnoho práce oproti MS-DOS usnadněno.
- Podpora širokého spektra hardware.

Tato část práce by měla provést nástin základů programování v prostředí Codegear Delphi s přihlédnutím k tvorbě animovaných fyzikálních modelů. Nebudeme se zde zabývat základy jazyka Pascal (u čtenáře bude předpokládána alespoň základní znalost problematiky).

Programování systému Windows a systém řízení zprávami Operační systém MS-DOS byl *jednouúlohový* operační systém. Pro zjednodušení můžeme říct, že v operačním systému vždy běžela jen jedna úloha (jeden program), nebylo možné spouštění více programů najednou. Existovaly sice tzv. TSR programy, které mohly běžet i za běhu jiné úlohy, ale nejednalo se o programy v pravém slova smyslu, byly to spíše ovladače hardware apod.

Oproti tomu Windows je *multitaskingový* operační systém. Je povolen současný běh většího množství programů (omezeno jen dostupnou pamětí) a operační systém Windows rozhoduje o přidělení procesoru tomu kterému běžícímu programu. Program může vykonávat svou práci (hrát hudbu, provádět výpočet, vykreslovat grafiku, ...) jen pokud má zrovna přidělený procesor. Mimo tuto dobu je program uspaný a čeká až na něj „přijde řada“. Díky tomu, že operační systém Windows přepíná (přiděluje procesor) mezi programy vysokou rychlostí, máme dojem, že všechny programy běží současně. Jedná se však pouze o iluzi způsobenou rychlým přepínáním mezi aplikacemi.

Operační systém tak musí mít možnost upozornit běžící programy na to, že nastala nějaká událost. Zatímco část programu pro operační systém MS-DOS mohla vypadat symbolicky takto:

```
nějaké akce ...
dělej nějakou práci
čekej na stisk klávesy
nějaké akce ...
```

v operačním systému Windows by se spíše jednalo o

```
dělej nějakou práci
uspi se a čekej, než operační systém dá vědět o události
```

a na jiném místě programu bychom našli

```
systém Windows nás upozornil na stisknutou klávesu:
dělej nějakou práci
uspi se a čekej, než operační systém dá vědět o další akci
```

Systém musí umět vzbudit čekající program a předat mu informaci (zprávu) o nastalé události. Proto se tento způsob řízení běhu programu nazývá *zprávami řízený*. Zpráva je informace operačního systému o nějaké události, která se našeho programu týká. Každá aplikace (mající hlavní okno) pro systém Windows obsahuje tzv. smyčku obsluhy zpráv, která symbolicky vypadá následovně:

```
while(VyzvedniZprávuOdSystému(zpráva) = true) do
```

```
begin
  case zpráva.Typ of
    Stisknuto_Levé_Tlačátko_Myši: proved' činnost
    Stisknuto_Pravé_Tlačátko_Myši: proved' činnost
    .
    .
    .

    Vykresli_Okno: Proved' vykreslení obsahu hlavního okna
  end;
  Uspi se a čekej na další zprávu.
end;
```

Program tak musí explicitně vyjmenovat všechny zprávy (odpovídající různým událostem), na které chce reagovat. Pokud nevyjmenuje a nedefinuje např. akci pro stisk pravého tlačítka myši, toto zůstane neobsloužené.

V předchozím kusu pseudo-kódu jsme mohli vidět, že je třeba reagovat i na požadavek vykreslení hlavního okna. Požadavek na vykreslení hlavního okna musí být obsloužen, jinak by z našeho programu uživatel viděl pouze rám, který je vykreslován operačním systémem. O obsah okna se musíme postarat sami.

Toto je klíčový poznatek, který musíme při tvorbě animované fyzikální simulace vést v patrnosti a požadavek překreslení okna správně obsluhovat. To je velká změna, se kterou musí uživatel předcházející k programování Windows z Pascalu (z operačního systému MS-DOS) smířit. V MS-DOS čáry a objekty jednou nakreslené na obrazovku zůstaly a musely se, pokud bylo třeba, explicitně mazat. Nyní je celá obrazovka mazána „když se to zrovna operačnímu systému hodí“ a nové nakreslení musí proběhnou po požadavku na obsloužení příslušné zprávy.

Situace je navíc ještě zkomplikovaná tím, že překreslení celé obrazovky může být časově náročné a uživatel pak překreslení nevidí jako jednu operaci, ale jako postupné vykreslování, někdy ještě s nepříjemným blikáním. Tento problém se řeší tak, že se nevykresluje přímo na obrazovku, ale na její virtuální obraz do operační paměti (do bitmapového obrázku) a ten je překreslován na obrazovku při obdržení požadavku na kreslení.

Vývojové prostředí Delphi naštěstí přináší značné ulehčení a většinu výše popsaných problémů řeší za uživatele.

Při tvorbě fyzikálního simulačního programu v Delphi tak můžeme postupovat obecně dvěma způsoby:

Použití zabudovaných komponent Ve vývojovém prostředí Delphi se nacházejí základní geometrické tvary (čára, obdélník, kružnice, ...) jako před-připravené komponenty a mají v sobě zabudovanou obsluhu mnoha zpráv operačního systému,

což programování velmi urychlí a usnadní. Především tak tyto komponenty obsluhují požadavek na překreslení a jejich zobrazení je tak (polo)automatické. Jejich použití ve větším počtu však může běh programu zpomalit (někdy až neúnosně).

Jako předpřipravená komponenta je v Delphi i komponenta `TImage`, která nám umožní v našich animacích použít bitmapové obrázky.

Přímá obsluha vykreslování Použití této metody přináší velkou svobodu v tom, jak a co je na obrazovku kresleno a oproti metodě první může být značně rychlejší. Cenou za tuto rychlost je však nutnost řešit výše popsání problémy.

Jako součást této práce však byla připravena „kostra“ programu, která problém s překreslováním řeší a může být jako základ fyzikální animace použita. V této ukázkové aplikaci je tak např. požadavek na překreslení obsloužen takto:

```
//požadavek na obsloužení kreslení
procedure TFKyvadloKruh.FormPaint(Sender: TObject);
begin
    //překreslení už dříve vyplněného obrázku na obrazovku -
    // ta je interpretována proměnnou Canvas
    Self.Canvas.Draw(0, 0, FBitmap);
end;
```

Bitmapový obrázek (v tomto případě o rozměrech 900 na 700 bodů) sloužící pro přímé kreslení do paměti je vytvořen voláním

```
var
    FBitmap: TBitmap;
begin
    FBitmap := TBitmap.Create;
    FBitmap.Width := 900;
    FBitmap.Height := 700;
end;
```

a konečně nakreslení např. kružnice se středem na souřadnicích 100, 100 a polo-
měru 10 je realizováno kódem

```
// funkci Ellipse se předává obdelník, do kterého je vykreslena
    elipsa, případně se předává čtverec v případě kružnice
    FBitmap.Canvas.Ellipse(100, 100, 110, 110);
end;
```

Pro počítačovou realizaci našeho modelu s matematickým kyvadlem bychom postupovali takto:

Z předchozího textu již víme, že budeme modelovat systém popsany diferenciální rovnicí:

$$\ddot{\varphi} + \frac{g}{l} \cdot \sin(\varphi) = 0$$

V tomto případě nemusíme použít přibližného předpokladu $x = \sin(x)$ a budeme pracovat přímo s rovnicí obsahující $\sin \varphi$.

Procedura v Delphi provádějící numerickou integraci by měla tuto podobu:

```
//definice globálních konstant
var
  FBitmap: TBitmap;
  //bitmapová obrázek na který proběhne vlastní vykreslení
  FFi_, FDt, FFi, Fg, FL : Extended;
  //proměnné představující:
  //Aktuální úhlovou rychlost, časovou konstantu integrace,
  //aktuální úhel, gravitační zrychlení, délku závěsu

//vlastní výpočet hodnot
procedure Kresli;
var
  fi__: Extended; //aktuální úhlové zrychlení kyvadla
  zaves: TPoint; //souřadnice závěsu
  stred: TPoint; //souřadnice středu kyvadla (hmotného bodu
                //reprezentujícího kyvadlo)
begin
  //výpočet aktuálního úhlového zrychlení bodu
  fi__ := -Fg/FL * Sin(FFi);
  //výpočet aktuálního úhlové rychlosti bodu
  FFi_ := FFi_ + fi__ * FDt;
  //výpočet aktuálního úhlu
  FFi := FFi + FFi_*FDt;

  //závěs je umístěn uprostřed obrázku
  zaves.X := FBitmap.Width div 2;

  //a výšce 10 bodů
  zaves.Y := 10;

  //výmaz předchozí polohy kyvadla
```

```
FBitmap.Canvas.FillRect(FBitmap.Canvas.ClipRect);

//výpočet kartézských souřadnic bodu
stred.X := Round(zaves.X+Fl*Sin(FFi));
//výpočet kartézských souřadnic bodu
stred.Y := Round(zaves.Y+Fl*Cos(FFi));

//vykreslení kyvadla
FBitmap.Canvas.Ellipse(stred.X - 10, stred.Y - 10,
    stred.X + 10, stred.Y + 10);

//vykreslení závěsu
FBitmap.Canvas.MoveTo(zaves.X, zaves.Y);
//vykreslení závěsu
FBitmap.Canvas.LineTo(stred.X, stred.Y);
end;
```

Tato funkce pro kreslení je opakovaně volána z jiného místa programu např. tímto způsobem

```
while not FClosing do
begin
    Kresli; //předchozí procedura
    //žádost o fyzické překreslení bitmapového obrázku FBitmap na obrazovku
    Application.ProcessMessages;
end;
```

Hlavní práce na simulačním programu je tímto hotova, do programu chybí doplnit potřebné ovládací prvky pro zvýšení použitelnosti a uživatelské přívětivosti.

V další části textu budou popsány další ukázkové simulační programy. Programy byly již použity při výuce předmětů *Vybrané kapitoly z klasické fyziky*, *Vybrané problémy moderní fyziky* a *Atomové a jaderné fyziky*. U studentů se setkaly s velkým zájmem. Domnívám se, že i přes časový tlak při výuce jednotlivých předmětů je nutné alespoň vybrané problémy „dotáhnout“ až do konce a nespokojit se s „pouhým“ nalezením např. diferenciální rovnice a konstatováním, že problém má to a to řešení a dále se problémem nezabývat. Ačkoli se mezi studenty jistě najdou tací, co by byli schopni předložený problém samostatně řešit, řešení si buď rovnou představit, nebo animovat na PC, ale dotažením problému až do konce (např. zobrazením nalezeného řešení na počítači), umožníme i slabším studentům „účastnit se řešení“ a neodradit je hned na počátku. Předkládané teorie pak nebudou tak snadno od-

souzeny jako nepoužitelné, na rozdíl od těch, u nichž řešení až do konce nebylo předloženo.

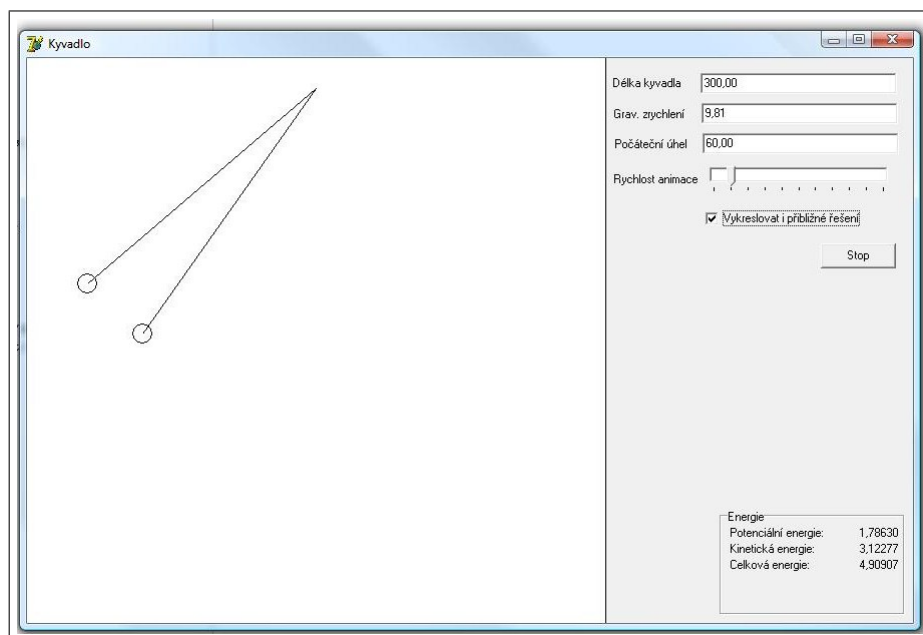
Není samozřejmě nutné a ani možné provádět „dořešení“ důsledně u všech problémů, stačí např. jednou pro jednu použitou matematickou nebo fyzikální metodu (např. Lagrangeovy rovnice).

2.6.2 Kyvadlo

Funkce a podrobný popis tohoto programu byl již proveden v předchozích částech. Aplikace demonstruje řešení diferenciální rovnice

$$\ddot{\varphi} + \frac{g}{l} \cdot \sin \varphi = 0.$$

Aplikaci ukazuje obrázek 2.



Obrázek 2: Demonstrační aplikace kyvadla

V aplikaci je možné nechat vykreslovat i řešení, kdy se ve výše uvedené diferenciální rovnici kvůli analytickému řešení pokládá rovnost $\sin x = x$. O tomto zjednodušení jsme již hovořili v kapitole 2.4. Dále je možné nastavit:

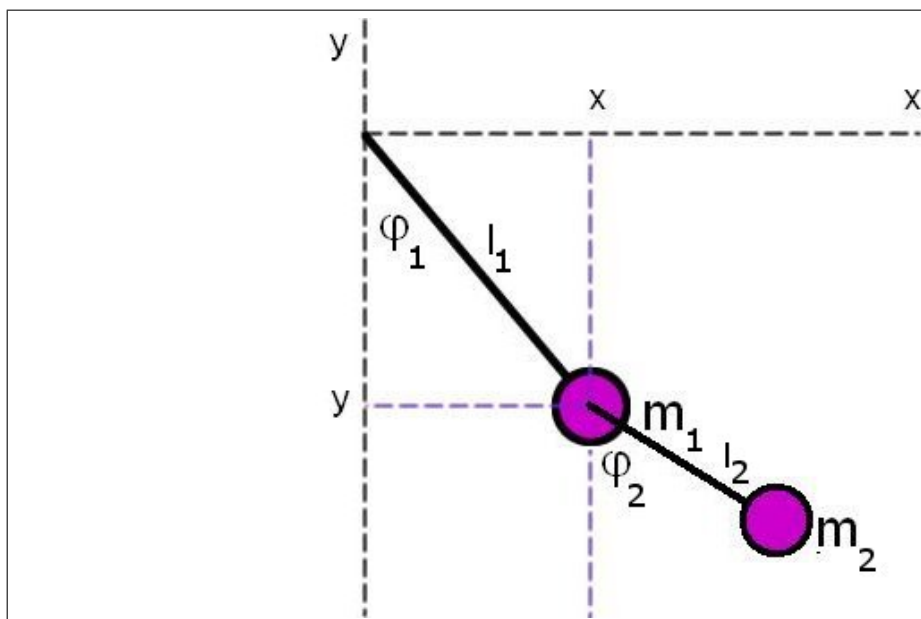
- Délku kyvadla (l),
- velikost gravitačního zrychlení (g),
- počáteční výchylku kyvadla (φ),
- rychlost animace.

Pozn. V této verzi aplikace není možné nastavení počáteční úhlové rychlosti kyvadla, v nové verzi se s tímto počítá.

Aplikace zobrazuje v každém okamžiku hodnotu kinetické, potenciální a celkové energie kyvadla. Numerické řešení diferenciální rovnice bylo provedeno Eulerovou metodou, při větší rychlosti animace (a tedy větším kroku metody) je vidět, že chybou numerického výpočtu kolísá hodnota celkové energie.

2.6.3 Dvojkyvadlo

Aplikace řeší diferenciální rovnice popisující oblíbený příklad z teoretické me-



Obrázek 3: Nákres řešení dvojkyvadla

chaniky. Chceme řešit situaci podle obrázku 3. Souřadnice prvního kyvadla můžeme zapsat jako:

$$\begin{aligned}x_1 &= l_1 \cdot \sin(\varphi_1) \\ y_1 &= -l_1 \cdot \cos(\varphi_1)\end{aligned}$$

Pro vyjádření okamžité rychlosti prvního kyvadla nalezneme časové derivace jednotlivých souřadnic jako

$$\begin{aligned}\dot{x}_1 &= l_1 \cdot \cos(\varphi_1) \cdot \dot{\varphi}_1 \\ \dot{y}_1 &= l_1 \cdot \sin(\varphi_1) \cdot \dot{\varphi}_1\end{aligned}$$

a rychlosti jako

$$v_1 = \sqrt{\dot{x}_1^2 + \dot{y}_1^2} = l_1^2 \cos^2 \varphi_1 \cdot \dot{\varphi}_1^2 + l_1^2 \sin^2 \varphi_1 \cdot \dot{\varphi}_1^2 = l_1^2 \cdot \dot{\varphi}_1^2.$$

U druhého kyvadla bude situace o něco složitější:

$$\begin{aligned} x_2 &= l_1 \cdot \sin(\varphi_1) + l_2 \cdot \sin(\varphi_2) \\ y_2 &= -l_1 \cdot \cos(\varphi_1) - l_2 \cdot \cos(\varphi_2) \end{aligned}$$

Pro vyjádření okamžité rychlosti druhého kyvadla nalezneme časové derivace jednotlivých souřadnic jako

$$\begin{aligned} \dot{x}_2 &= l_1 \cdot \cos(\varphi_1) \cdot \dot{\varphi}_1 + l_2 \cdot \cos(\varphi_2) \cdot \dot{\varphi}_2 \\ \dot{y}_2 &= l_1 \cdot \sin(\varphi_1) \cdot \dot{\varphi}_1 + l_2 \cdot \sin(\varphi_2) \cdot \dot{\varphi}_2 \end{aligned}$$

a rychlost jako

$$v_2 = \sqrt{\dot{x}_2^2 + \dot{y}_2^2} = \dots = l_1^2 \cdot \dot{\varphi}_1^2 + l_2^2 \cdot \dot{\varphi}_2^2 + 2 \cdot l_1 \cdot l_2 \cdot \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2).$$

Langrangeovu funkci stejně jako v předchozích případech vyjádříme rozdílem kinetické a potenciální energie:

$$\begin{aligned} L = T - V &= \frac{1}{2} \cdot m_1 \cdot l_1^2 \cdot \dot{\varphi}_1^2 + \frac{1}{2} \cdot m_2 \cdot l_1^2 \cdot \dot{\varphi}_1^2 + \frac{1}{2} \cdot m_2 \cdot l_2^2 \cdot \dot{\varphi}_2^2 + \\ &+ m_2 \cdot l_1 \cdot l_2 \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2) + (m_1 + m_2) \cdot g \cdot l_1 \cdot \cos(\varphi_1) + \\ &+ m_2 \cdot g \cdot l_2 \cdot \cos(\varphi_2). \end{aligned}$$

Vývoj jednotlivých zobecněných souřadnic systému pak nalezneme aplikací podmínek

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = 0,$$

kde q značí zobecněnou souřadnici (úhel φ) a $\dot{q}$ zobecněnou rychlost ($\dot{\varphi}$ — úhlová rychlost). Lagrangeova rovnice po dosazení vyjde pro úhel φ_1 :

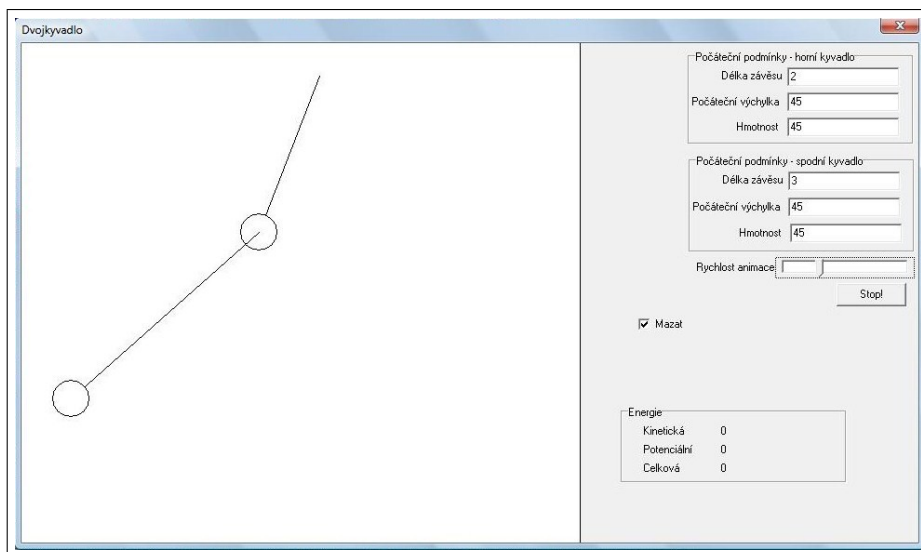
$$\begin{aligned} (m_1 + m_2) \cdot l_1^2 \cdot \ddot{\varphi}_1 + m_2 \cdot l_1 \cdot l_2 \cdot \ddot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2) + m_2 \cdot l_1 \cdot l_2 \cdot \dot{\varphi}_2^2 \cdot \sin(\varphi_1 - \varphi_2) + \\ (m_1 + m_2) \cdot g \cdot l_1 \sin \varphi_1 = 0 \end{aligned}$$

a pro úhel φ_2 :

$$m_2 \cdot l_2 \cdot \ddot{\varphi}_2 + m_2 \cdot l_1 \cdot l_2 \cdot \ddot{\varphi}_1 \cdot \cos(\varphi_1 - \varphi_2) - m_2 \cdot l_1 \cdot l_2 \cdot \dot{\varphi}_1^2 \cdot \sin(\varphi_1 - \varphi_2) +$$

$$m_2 \cdot g \cdot l_2 \cdot \sin(\varphi_2) = 0.$$

Numerické řešení těchto rovnice ukazuje aplikace dle obrázku 4.



Obrázek 4: Demonstrační aplikace dvojkyvadla

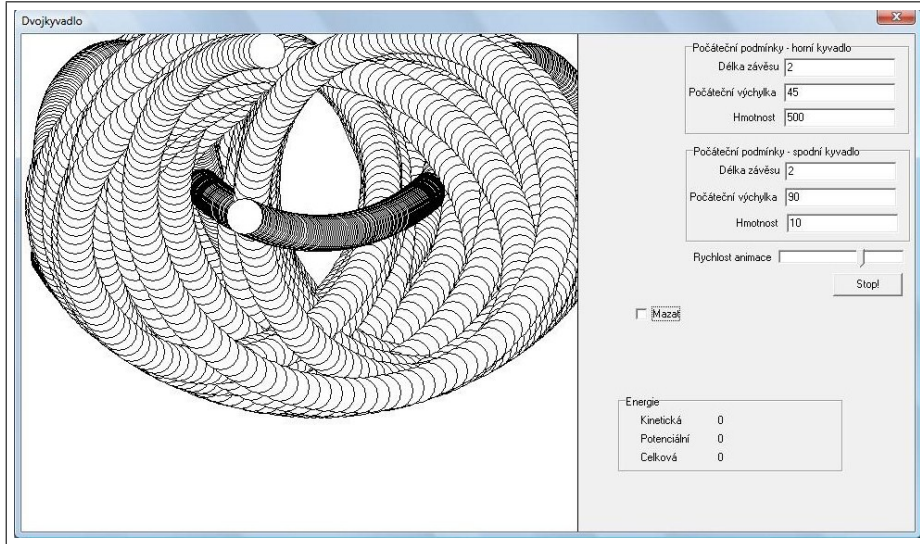
Jednotlivé nastavitelné parametry:

- Délku závěsů obou kyvadel (l_1, l_2),
- hmotnost obou kyvadel (m_1, m_2),
- počáteční výchylku obou kyvadel (φ_1, φ_2),
- rychlost animace,
- zda mazat obrazovku při každém novém kreslení.

Pozn. V této verzi aplikace není možné nastavení počátečních úhlových rychlostí kyvadel, v nové verzi se s tímto počítá.

Zajímavá nastavení parametrů:

- Velká hmotnost prvního kyvadla (500 kg) a malá hmotnost druhého (10 kg), délky závěsů (2 m a 2 m). Druhé kyvadlo se snadno obtáčí okolo prvního.
- Velká hmotnost druhého kyvadla (500 kg) a malá prvního. Výchylka kyvadel (-45° a 90°). Kyvadlo se téměř zastaví.
- Vypnout mazání obrazovky po každém vykreslení vytváří esteticky zajímavé obrazce, viz obrázek 5.



Obrázek 5: Demonstrační aplikace dvojkyvadla, trajektorie hmotných bodů

2.6.4 Kyvadlo na kruhu

Simulace zobrazuje pohyb kyvadla, jehož závěs je nucen pohybovat se konstantní úhlovou rychlostí po kružnici, viz obrázek 6.

Standardním postupem nalezneme souřadnice kyvadla:

$$\begin{aligned}x &= R \cdot \cos(\alpha) + l \cdot \sin(\varphi) \\ y &= R \cdot \sin(\alpha) - l \cdot \cos(\varphi),\end{aligned}$$

kde R je poloměr kružnice, α je úhel mezi osou x a spojnici středu kružnice a závěsu kyvadla, l délka závěsu kyvadla a φ úhel mezi svislicí a závěsem kyvadla.

Pro rychlost nalezneme:

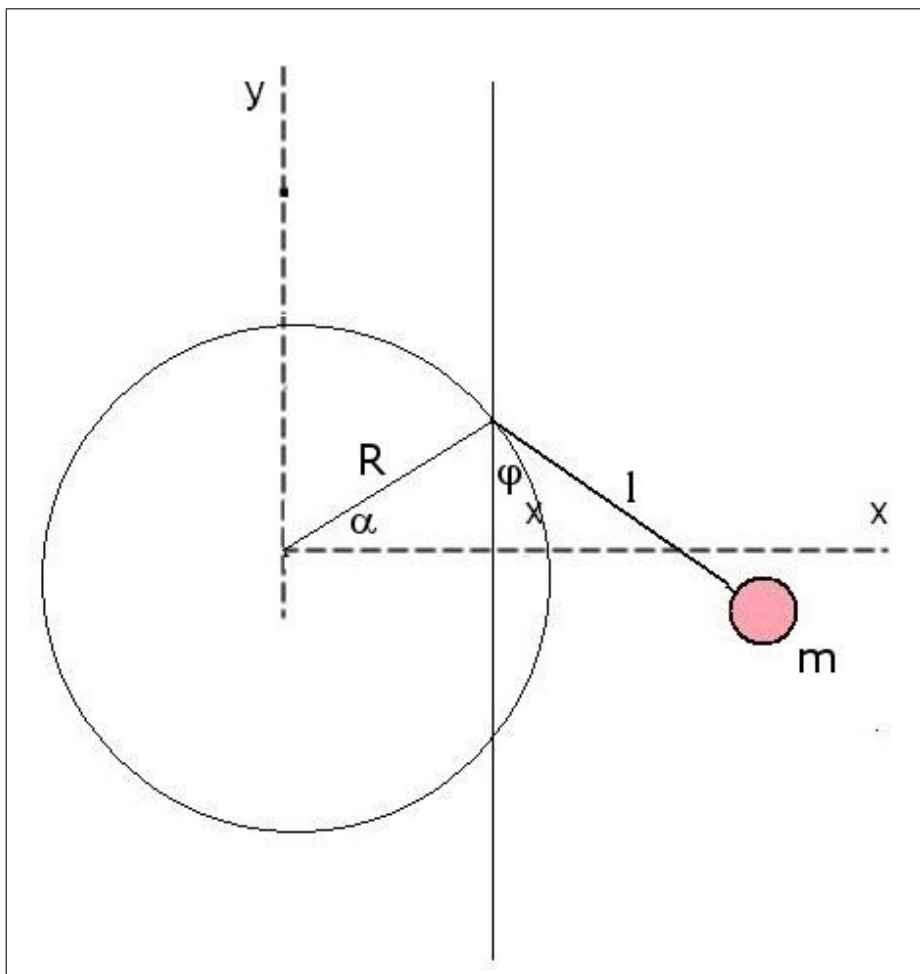
$$v = \sqrt{\dot{x}^2 + \dot{y}^2} = \dots = l^2 \cdot \dot{\varphi}^2 + R^2 \cdot \dot{\alpha}^2 + 2 \cdot l \cdot R \cdot \dot{\varphi} \cdot \dot{\alpha} \cdot \sin(\varphi - \alpha)$$

a Langrangeova funkce systému je:

$$\begin{aligned}L = T - V &= \frac{1}{2} \cdot m \cdot l^2 \cdot \dot{\varphi}^2 + \frac{1}{2} \cdot R^2 \cdot \dot{\alpha}^2 + m \cdot l \cdot R \cdot \dot{\varphi} \cdot \dot{\alpha} \cdot \sin(\varphi - \alpha) - \\ &\quad m \cdot g \cdot R \cdot \cos(\alpha) + m \cdot g \cdot l \cdot \cos(\varphi).\end{aligned}$$

A konečně Lagrangeova rovnice je po úpravě:

$$\ddot{\varphi} - \frac{R}{l} \cdot \cos(\varphi - \alpha) \cdot \dot{\alpha}^2 + \frac{g}{l} \cdot \sin(\varphi) = 0.$$



Obrázek 6: Nákres řešení kyvadla na kruhu

Aplikaci řešící tuto rovnici ukazuje obrázek 7.

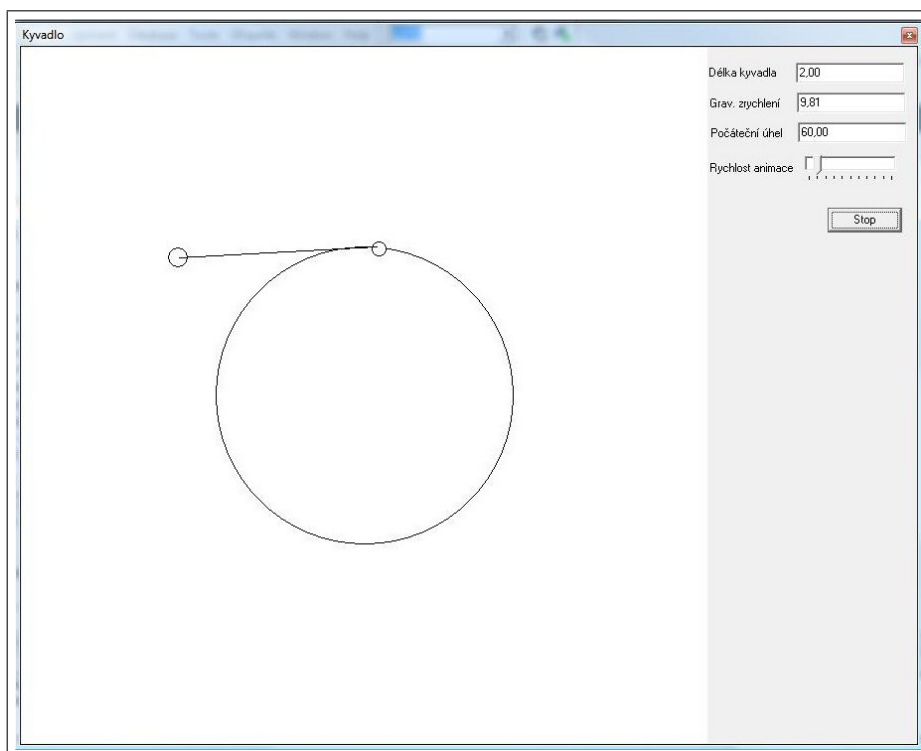
2.6.5 Kyvadlo na vozíku

Simulace zobrazuje pohyb kyvadla, jehož závěs se může vodorovně pohybovat — pevně připevněn k vozíku, jež se bez tření posunuje po ose x . V souladu s obrázkem 8 označíme M hmotnost vozíku, m hmotnost kyvadla, l délku závěsu kyvadla, φ úhel mezi svislicí a závěsem kyvadla.

Nalezněme souřadnice kyvadla (značeny malými písmeny) a vozíku (velká písmena):

$$\begin{aligned}x &= X + l \cdot \sin(\varphi) \\ y &= -l \cdot \cos(\varphi).\end{aligned}$$

A pro rychlost $v^2 = \dot{X}^2 + 2 \cdot \dot{X} \cdot l \cdot \cos(\varphi) \cdot \dot{\varphi} + l^2 \cdot \dot{\varphi}^2$. Lagrangeovu funkci systému nalezneme jako:



Obrázek 7: Demonstrační aplikace kyvadla na kružnici

$$L = T - V = \frac{1}{2} \cdot M \cdot \dot{X}^2 + \frac{1}{2} \cdot m \cdot (2 \cdot \dot{X} \cdot l \cdot \cos(\varphi) \cdot \dot{\varphi}) + m \cdot g \cdot l \cdot \cos(\varphi).$$

Lagrangeovy rovnice systému řeší tato ukázková aplikace.

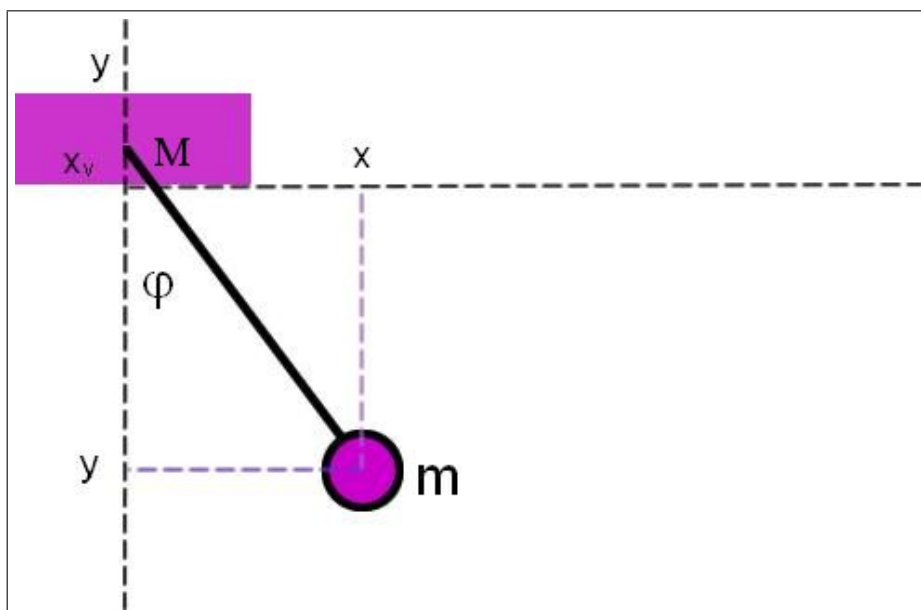
Ovládání programu Rychlost simulace lze ovlivnit vodorovným posuvníkem a běh animace lze spustit tlačítkem **Start!**. Parametry, jež je možné nastavit:

- Hmotnost vozíku (M),
- hmotnost kyvadla (m),
- počáteční výchylka kyvadla (φ),
- rychlost v .

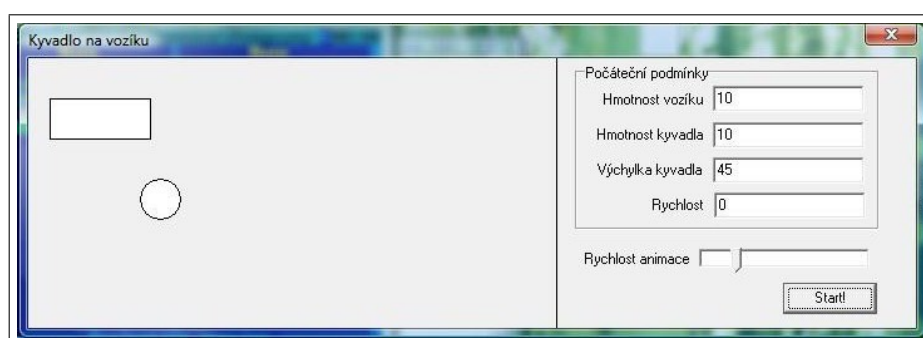
Program bezprostředně po spuštění ukazuje obrázek 9 a po chvíli běhu animace obrázek 10.

Zajímavá nastavení parametrů:

- Velká hmotnost vozíku (100 kg) a malá kyvadla (1 kg). Dle očekávání, o mnoho těžší vozík stojí a kyvadlo se pohybuje, jako kdyby závěs nebyl pohyblivý.



Obrázek 8: Demonstrační aplikace kyvadla na kružnici



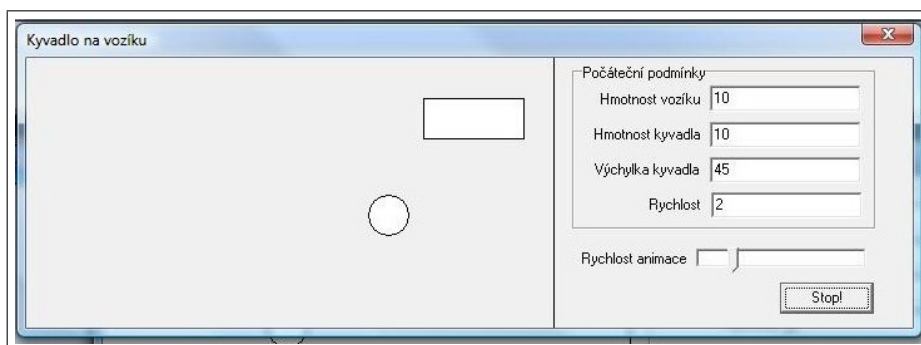
Obrázek 9: Kyvadlo s vodorovně pohyblivým závěsem, start

- Velká hmotnost kyvadla (100 kg) a malá vozíku (1 kg). Kyvadlo kmitá téměř ve vertikálním směru a o mnoho lehčí vozík se harmonicky pohybuje.
- Počáteční rychlost nastavena na např. 5, vodorovný pohyb vozíku v kladném směru osy x je „strháván“ zpětným pohybem kyvadla.

2.6.6 Ohyb částic na štěrbině

Simulace zobrazuje kvantově mechanický ohyb mikročástice na štěrbině a vytvoření ohybového obrazce. V literatuře můžeme pro pravděpodobnost odchýlení o úhel α nalézt vzorec

$$P_{\alpha} = \frac{\sin^2 \left(\frac{\pi \Delta x p}{h} \cdot \sin \alpha \right)}{\left(\frac{\pi \Delta x p}{h} \cdot \sin \alpha \right)^2},$$



Obrázek 10: Kyvadlo s vodorovně pohyblivým závěsem, běžící simulace

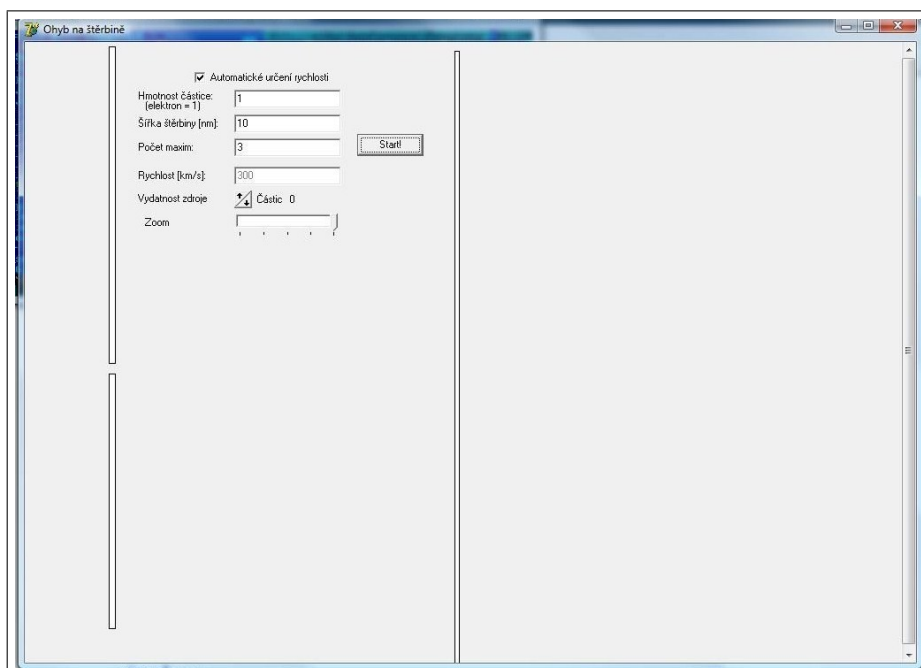
kde p značí hybnost částice, Δx šířku štěrby, h Planckovu konstantu, α úhel odchýlení. Vztah je formálně obdobný vzorci pro intenzitu a interferenci světla po ohybu, avšak podstata děje je jiná. Výška jednotlivých sloupců, na které je obrazovka rozdělena, odpovídá četnosti dopadu částice do tohoto místa. Program ukazuje i „dráhu“ elektronu, ačkoli kvůli pravděpodobnostní podstatě mikro světa není samozřejmě možné elektron jako o hmotný bod uvažovat. V programu je možné libovolné nastavení následujících parametrů simulace:

- Hmotnost částice v jednotkách m_e (hmotnostech elektronu),
- šířku štěrby v nm,
- počet požadovaných maxim (tím je automaticky vypočítána rychlost částic),
- rychlost částic (lze buď přímo zadat, nebo nechat automaticky určit podle počtu zadaných maxim),
- vydatnost zdroje (počet částic vyletujících ze zdroje za jednotku času),
- zoom (přiblížení ohybového jevu).

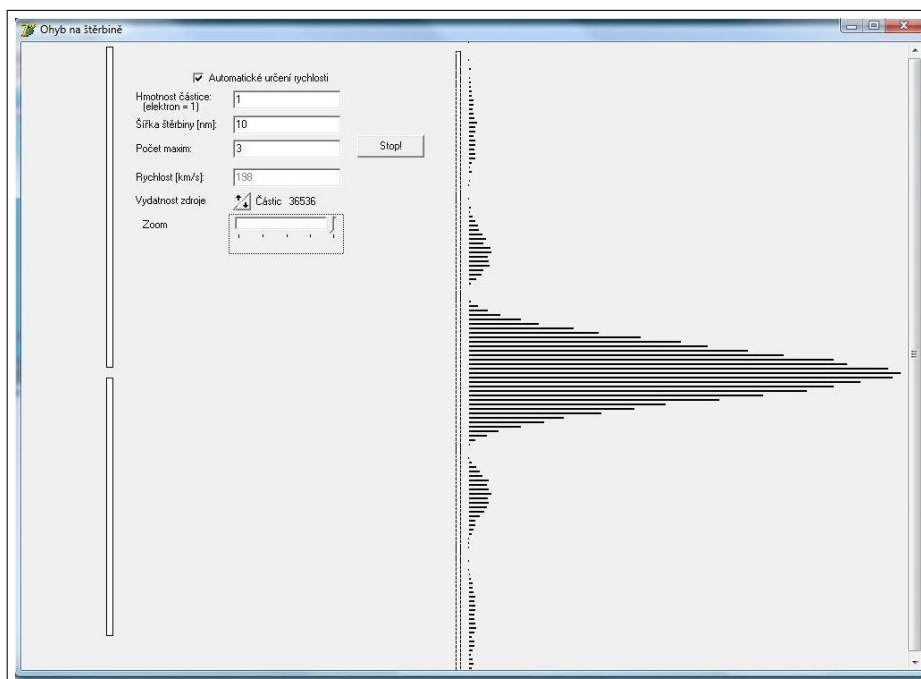
Program může ve výuce posloužit k demonstraci:

- Statistické podoby ohybového obrazce,
- závislosti vlastností ohybového obrazce na jednotlivých parametrech,
- poměru výšek (intenzit) jednotlivých maxim.

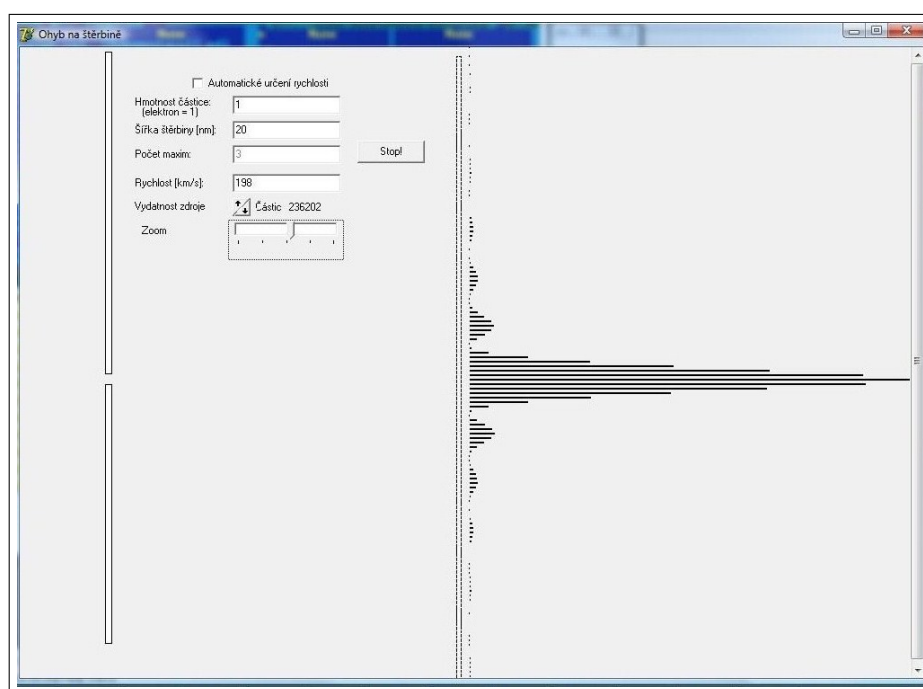
Úvodní obrazovku programu zobrazuje obrázek 11 a po nastavení parametrů a s 36536 emitovanými částicemi obrázek 12. Po rozšíření štěrby na dvojnásobek můžeme sledovat „zhuštění“ difrakčních maxim.



Obrázek 11: Ohyb částic na štěrbíně, start



Obrázek 12: Ohyb částic na štěrbíně, 36536 emitovaných částic, šířka štěrbiny 10 nm



Obrázek 13: Ohyb částic na šterbině, 236204 emitovaných částic, šířka šterbiny 20 nm

2.6.7 Použití prostředí Adobe Flash

Program Flash umožňuje tvorbu animací a prezentaci multimediálního obsahu na počítači. Byl primárně navrhnut pro tvorbu aktivního obsahu pro internet a internetové prohlížeče bez nutnosti instalovat další specializované programy. Program je na tomto poli jasným vítězem a zobrazování flashových animací (tzv. apletů) je podporováno naprostou většinou internetových prohlížečů. Aplety nejsou omezeny jen na oblast internetu, je možné je zobrazovat i v desktopových aplikacích, např. pomocí OLE kontejnerů, nebo jejich přímé zobrazení v přehrávačích Flash.

Uživatel nemusí pro tvorbu Flash apletů „bojovat“ s úskalími programování v operačním systému Windows, ale na druhou stranu se musí učit specializovaný *programovací*, nebo *skriptovací* jazyk — Action Script. Syntaxe tohoto jazyka naštěstí není nijak složitá a po prohlédnutí několika příkladů, je uživatel s alespoň základními zkušenostmi s programováním schopen napsat jednoduchý program sám. Uvedme zde některé výhody a nevýhody programování ve Flashi.

Výhody:

Podpora internetu. Aplety ve Flashi jsou snadno umístitelné na HTML stránky.

Na rozdíl od tvorby přímo spustitelných programů (např. v Delphi) nemusí uživatel instalovat na svůj počítač žádné další programy (kromě modulu Flash do prohlížeče internetu, který již ale nejspíš stejně na jeho počítači byl), uživatel se nemusí obávat virů a spouštění neověřených programů.

Podpora multimédií. Flash přímo podporuje práci se všemi běžnými typy multimédií (zvuků, obrázků, ...).

Nenutí programovat. Jednoduché animace je možné ve Flashi vytvořit i bez nutnosti psaní programu, pouhým použitím animačních nástrojů jež Flash nabízí — (Key frames, ...).

Přenositelnost. Přehrávače Flash jsou k dispozici i pro jiné operační systémy než MS-Windows. Vynikající podporu nabízí i Linux, Macintosh, ...

Nevýhody:

Rychlost Grafická bohatost a snadná aplikace multimédií svádějí k tvorbě „barevných a pohyblivých se“ animací a to může mít negativní dopad na rychlost animace. Aplety jsou náročné na hardware počítače a oproti programům psaných přímo (Delphi apod.) jsou znatelně pomalejší. V běžných jednoduchých aplikacích tento fakt nemusí být limitní, ale Flash se jeví jako nevhodný pro použití na složité vědecké výpočty.

Tvorba simulací v Adobe Flash Díky návrhu programovacího prostředí Adobe Flash nemusí programátor řešit aplikační detaily, jež byly popsány v předchozích oddílech. Pohybující se animace lze v Adobe Flash vytvořit v zásadě dvěma způsoby:

- Simulace pomocí keyframes,
- simulace pomocí Action Script.

První způsob — použití keyframes (klíčových snímků) — zde nebudeme blíže studovat, dostatek informací lze nalézt v kterékoli základní učebnici Adobe Flash. V této animaci nastavíme pouze počáteční a konečný stav animace (rozložení objektů na snímku). Právě tyto snímky jsou označeny jako *key frames* a Adobe Flash vytvoří potřebné mezi-snímky a je tak dosaženo plynulého animování snímku. Výhodou tohoto přístupu je jednoduchost, nevýhodou pak malá možnost ovlivnění výsledné animace. Tento způsob pak nelze v širší míře při tvorbě fyzikálních animací použít.

Druhý způsob — použití Action Script — přináší velkou volnost při tvorbě „hýbajících“ se snímků a jako takový je pro tvorbu simulací velmi vhodný.

Základní obrazovka Flashové aplikace se nazývá *scéna* a jednotlivé objekty na scéně jako takzvané *movie clips*. O příslušných událostech které nastaly v rámci jednoho movie clipu jsme informováni prostřednictvím událostí. Můžeme tak reagovat na stisky kláves, operace s myší, událost inicializace movie clipu, ...

Pro naše účely bude nejdůležitější událostí událost `onClipEvent`, konkrétně `onClipEvent (enterFrame)`. Právě v obsluze této události spočívá elegance druhé metody tvorby animací. Zavoláním této metody dává Adobe Flash programátorovi možnost provést změny na snímku před tím, než je vykreslen na obrazovku. Pokud je tedy pro Flashovou aplikaci nastavena snímkovací frekvence (počet snímků zobrazených za sekundu) např. na hodnotu 10, je tato metoda volána desetkrát za sekundu a jemnou manipulací s objekty na snímku je vytvořen dojem pohybu. V následujícím oddílu podrobně ukážeme jednotlivé kroky potřebné pro vytvoření animace šikmého vrhu hmotného bodu (míčku).

Tvorba matematického modelu: Vytváříme fyzikální simulaci šikmého vrhu míčku

s elevačním úhlem α a počáteční rychlostí v_0 . Počátek souřadnicového systému položíme do bodu ze kterého míček vystřelujeme a standardním způsobem zavedeme souřadný kartézský systém. Pak ve směru osy x nepůsobí na míček žádná síla a míček se v tomto směru pohybuje rovnoměrným pohybem (zanedbáváme vliv prostředí) rychlostí $v_x = v_0 \cdot \cos \alpha$. Ve směru osy y působí na míček gravitace a ten se vlivem této síly pohybuje nejprve rovnoměrně zpomaleným a pak rovnoměrně zrychleným pohybem se zrychlením g . Počáteční rychlost v ose y je $v_{0y} = v_y = v_0 \cdot \sin \alpha$. Okamžitou polohu míčku na ose x tedy okamžitě napíšeme jako $x = v_x \cdot t = v_0 \cdot \cos(\alpha) \cdot t$ a na ose y jako $y = v_{0y} \cdot t - \frac{1}{2} \cdot g t^2$.

Zobrazení v Adobe Flash: Časovou osu rozdělíme na dílky dt a v každém volání události `onClipEvent(enterFrame)` zvýšíme počítadlo času o dílek dt a změníme souřadnice objektu *ball* — obrázek míčku. Abychom dosáhli efektu odražení míčku nastavme ypsilonovou složku rychlosti na 70 % počáteční rychlosti ve směru y , jakmile ypsilonová souřadnice dosáhne hodnotu $y = 0$. Konečně pro zlepšení vzhledu animace nechme míček rotovat. Následuje výpis kódu, kterým tohoto dosáhneme.

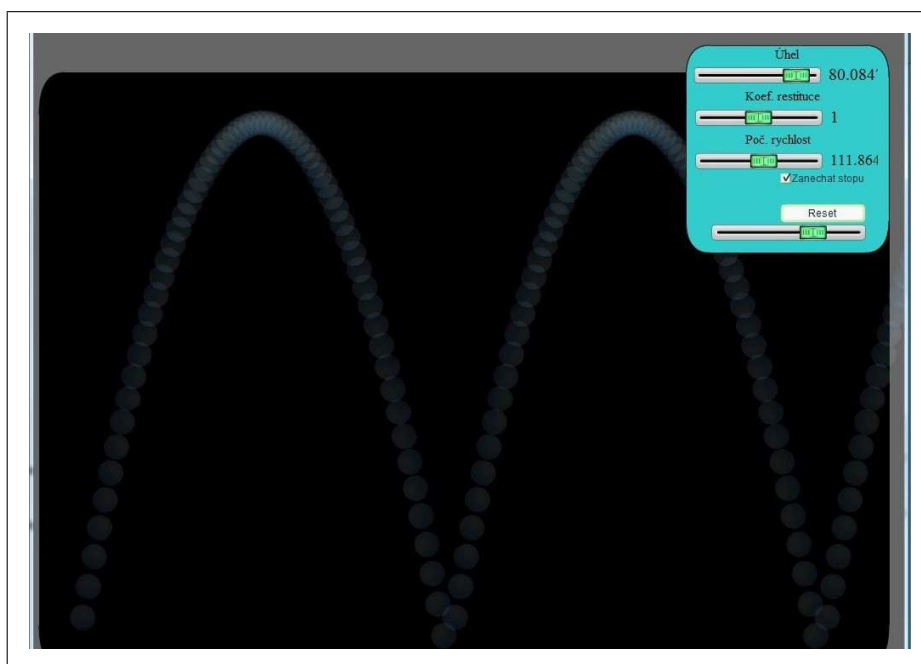
```
//tato funkce počítá polohu na ose x
function poloha_x(rychlost,cas){
return rychlost*Math.cos(alpha*3.14/180)*cas;
}
//tato funkce počítá polohu na ose y
function poloha_y(rychlost,cas){
return rychlost*Math.sin(alpha*3.14/180)*cas-0.5*9.81*cas*cas;
}
//počáteční hodnoty
x0=30; //počáteční poloha x
y0=400; //počáteční poloha y
v=100; //počáteční rychlost
t=0;
//funkce onEnterFrame
ball.onEnterFrame=function(){
    //přičtíme čas
    t+=0.3;
    //nastavme nové souřadnice míčku a rotaci

    ball._x=x0+poloha_x(v,t);
    ball._y=y0-poloha_y(v,t);
    ball._rotation=t*30;

    if (ball._y>inity){
        initx=ball._x;
        t=0;
        v*=0.7;
    }
}
```

Uvedným okomentovaným kódem jsme vytvořili fyzikální podobu šikmého vrhu míčku. Na doprovodném CD je vytvořena ukázková aplikace vycházející z výše popsaného matematického modelu a programového kódu. Tato ukázková aplikace bude nyní popsána.

Flashová animace skákajícího míčku Simulace vychází z výše popsaného matematického modelu a zobrazuje jej na obrazovku počítače. Zdrojový kód programu je uložen na doprovodném CD a může tak posloužit zájemcům k bližšímu prozkoumání. Obrazovka s vykresleným odrazovým obrazcem je na obrázku 14.



Obrázek 14: Šikmý vrh míčku v Adobe Flash

Míčku lze ovládacími prvky nastavit:

- Elevační úhel,
- koeficient restituce — poměr rychlosti dopadajícího a odraženého míčku,
- počáteční rychlost,
- rychlost animace.

Tvorba fyzikálních apletů není omezena pouze na pohybující se modely/simulace. Statické aplety mohou být výhodně použity na zobrazování různých grafů, průběhů apod. Nastavením různých parametrů toho kterého grafu je možné snadno studovat vlastnosti jednotlivých průběhů, závislost na parametrech apod.

Jako příklad tvorby statického modelu zde uvedme zpracování Planckova vyzařovacího zákona.

Planckův vyzařovací zákon Planckův vyzařovací zákon popisuje souvislost mezi intenzitou a frekvencí vyzařování absolutně černého tělesa (AČT). Problém je dostatečně znám svou odolností k nalezení řešení a dlouho byl fyzikální záhadou.

Klasický popis problému (a řešení klasickou statistickou fyzikou) vedl k zákonu, jež je na počest dvou anglických fyziků kteří ho odvodili nazýván Rayleighův–Jeansův zákon. Zákon pro hustotu spektrální energie udává:

$$\varepsilon(\omega, T) = \frac{\omega^2}{\pi^2 \cdot c^3} \cdot k \cdot T,$$

kde k je Boltzmanova konstanta, T termodynamická teplota AČT, ω úhlová frekvence vyzařování a c rychlost světla. Úhlová frekvence vyzařování souvisí s frekvencí f vyzařování známým vztahem $\omega = 2 \cdot \pi \cdot f$.

Tento zákon však dává souhlas s experimentem pouze pro malé hodnoty ω (frekvencí) a v oblasti vysokých frekvencí selhává. Kromě tohoto problému při hledání celkové objemové hustoty energie v dutině AČT, tedy vyčíslení vzorce

$$\int_0^\infty \varepsilon(\omega, T) d\omega.$$

dává nekonečno. Právě kvůli selhání vzorce v oblasti vysokých frekvencí bývá tento problém někdy označován jako ultrafialová katastrofa.

V oblasti vysokých frekvencí pak problém přijatelně popisuje Wienův zákon

$$\varepsilon(\omega, T) = a \cdot \omega^3 \cdot \exp\left(-b \cdot \frac{\omega}{T}\right),$$

kde konstanty a a b byly určeny z experimentálně zjištěných hodnot.

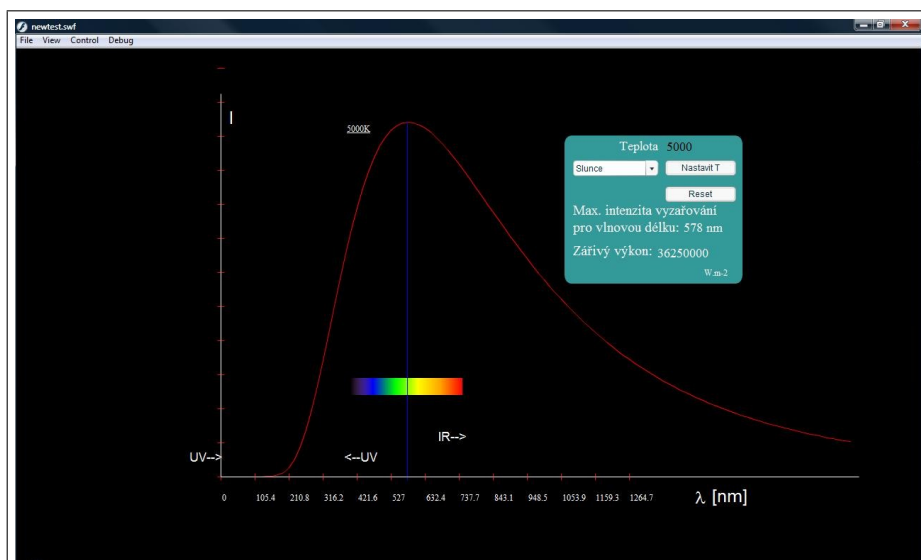
K vzorci, který považujeme jako platný dnes, dospěl až Max Planck postulováním přenosu energie po kvantech. Ač toto původně považoval za pouhý matematický „trik“, později se ukázalo že kvantování je s mikrosvětlem nerozlučně spjato.

Planckova teorie tak má pro počátky moderní fyziky nesmírně důležitý význam a její pochopení je klíčové. V dalším popsany počítačový model může zájemcům pomoci s pochopením tohoto fundamentálního fyzikálního problému.

V aplikaci lze zadat teplotu AČT a po stisku tlačítka *Nastavit* T bude vykreslena příslušná vyzařovací křivka závislosti intenzity záření (osa y) a vlnové délky záření (osa x). Výběr z rolovacího seznamu nabízí k zobrazení několik typických teplot.

Spuštěnou aplikaci ukazuje obrázek 15.

Vynesení většího počtu křivek pro různé teploty pěkně ukazuje posun maxima intenzity vyzařování směrem k větším vlnovým délkám (nižší frekvenci a energii), viz obrázek 16.



Obrázek 15: Planckův vyzařovací zákon, teplota AČT 5000 K

3 Využití programu Mathematica

Tato část práce zhodnotí možnosti využití programu Mathematica ve výuce fyziky. Při studiu fyziky často postupujeme následujícím způsobem:

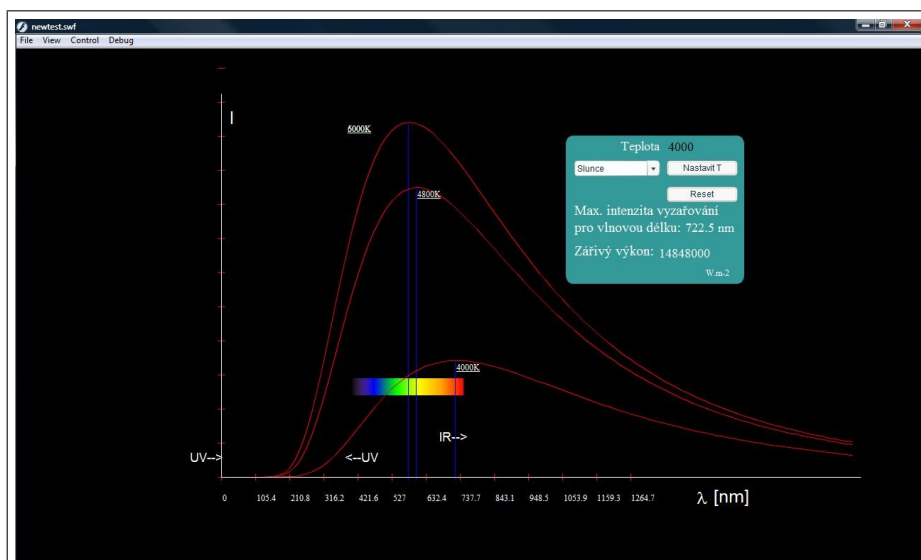
- Nalezneme problém, který se rozhodneme řešit.
- Sestavíme matematický popis problému (matematický model).
- Provedeme matematické řešení.
- Zkoumáme výsledky řešení matematického modelu.

Ačkoli je někdy největším problémem sestavení příslušného matematického modelu, často je možné uvážnout i na třetím bodu postupu — matematickém řešení.

Pro matematické řešení je mnohdy nutné ovládat velké množství metod, vzorců a postupů, jejichž osvojení nelze vždy pokládat za samozřejmé. Někdy ve výuce narazíme i na to, že je student schopen sestavit model, nalézt potřebné vztahy a závislosti a poté nedokončit řešení kvůli slabší matematické přípravě. To, že je při výuce fyziky a řešení fyzikálních problémů procvičována i matematická znalost studenta je jistě chválihodné, avšak nebylo by správné strávit např. deset minut z hodiny vystavením fyzikálního problému a zbytek času jeho matematickým řešením.

V takovéto situaci je možné postupovat dvěma způsoby:

Rezignujeme na podrobné matematické řešení a prohlásíme, že problém je matematicky řešitelný a bez dalších důkazů napíšeme hotový výsledek. Ač se s tímto způsobem můžeme často setkat, není to způsob vždy vhodný. Dochází při něm



Obrázek 16: Plankův vyzařovací zákon, posun maxima vyzařování s chladnutím AČT

k přerušení přirozeného postupu (zmíněného výše) přeskočením třetího bodu. Student tak může nabýt dojmu, že se ani fyziku a její postupy učit nemusí, že sestavené rovnice stejně poté není schopen vyřešit a je k dalšímu učení se fyzice demotivován. Pokud je však matematické řešení problému opravdu nad jeho síly (nebo časové možnosti ve výuce), můžeme postupovat podle následujícího bodu.

Řešíme problém pomocí automatizovaného matematického programu. Student tak neztrácí kontinuitu řešení, hotové matematické řešení je možné na obrazovce počítače přehledně vizualizovat a s řešením je možné dále pracovat, např. vnášet do řešení otázky „co kdyby“, a tím do problému a celé fyziky, jež je v problému skryta, lépe proniknout. Otázkami „co kdyby“ myslíme práci s fyzikálními parametry situace, např: Co kdyby v tomto řešení šikmého vrhu figurovala obrácená nebo silnější(slabší) gravitační síla. Nebo co kdybychom v tomto řešení použili místo elektronu pozitron apod. Změna těchto parametrů a okamžitá odezva v podobě změněného výsledku je didakticky velmi výhodná. I samotný fakt, že jsme sestavili tak obtížné zadání, že jeho řešení musíme zadat počítači, může být pro studenta značně motivující.

Rozhodneme-li se tedy postupovat ve výuce podle druhého bodu předchozího výčtu, musíme nalézt matematický software, který by vyhovoval nastíněným požadavkům. Ačkoli byly předchozí příklady platné spíše pro středoškolské žáky, potřeba matematického řešení složitých vztahů vyvstane i na škole vysoké.

Fakt, že v podstatě „přeskakujeme“ třetí bod z námi popsaného postupu, je výhodný i z toho pohledu, že student není samotným matematickým řešením „roz-

ptylován“ a pozornost není odváděna od vlastní fyziky. V opačném případě může dojít k tomu, že student vyřeší složitý matematický problém, a mezitím zapomene, jaká fyzika je v problému ukryta.

Programů, které by splňovaly výše zmíněné požadavky lze nalézt celou řadu. Lze najít program specializovaný na samotné výpočty, jiný na tvorbu grafů a ještě jiný např. na řešení diferenciálních rovnic. S výhodou lze nalézt a použít programový balík, jež by měl vše v sobě. Jednou z takovýchto sad software může být **Wolfram Mathematica**. O možném použití tohoto programu mluví další text této kapitoly.

Ačkoli má program Mathematica velmi dobře zpracovaný systém nápovědy a na internetu lze nalézt velké množství informací (bohužel hlavně cizojazyčných), budou na tomto místě uvedeny základní pokyny k obsluze tohoto mocného nástroje. Ukázky obrazovek programu budou vztaženy k verzi 8.0, základy ovládání a syntaxe příkazů je však stejná i v ostatních verzích programu.

CD přiložené k této práci obsahuje většinu zde zpracovaných ukázek. K prohlédnutí je nutné na počítač nainstalovat buď přímo program Mathematica, nebo postačí pouze volně stažitelný program Mathematica Player. Ten je možné stáhnout z adresy <http://www.mathematica.com/products/player>.

3.1 Základy práce s programem Mathematica

Po spuštění programu nás uvítá základní okno programu. Jednotlivé dokumenty otevřené v rámci jedné instance programu jsou terminologií Mathematica nazývány jako *notebooks* — zápisníky. Pro podobnost se sešity v programu Excel se zde přidržíme označení *sešit*.

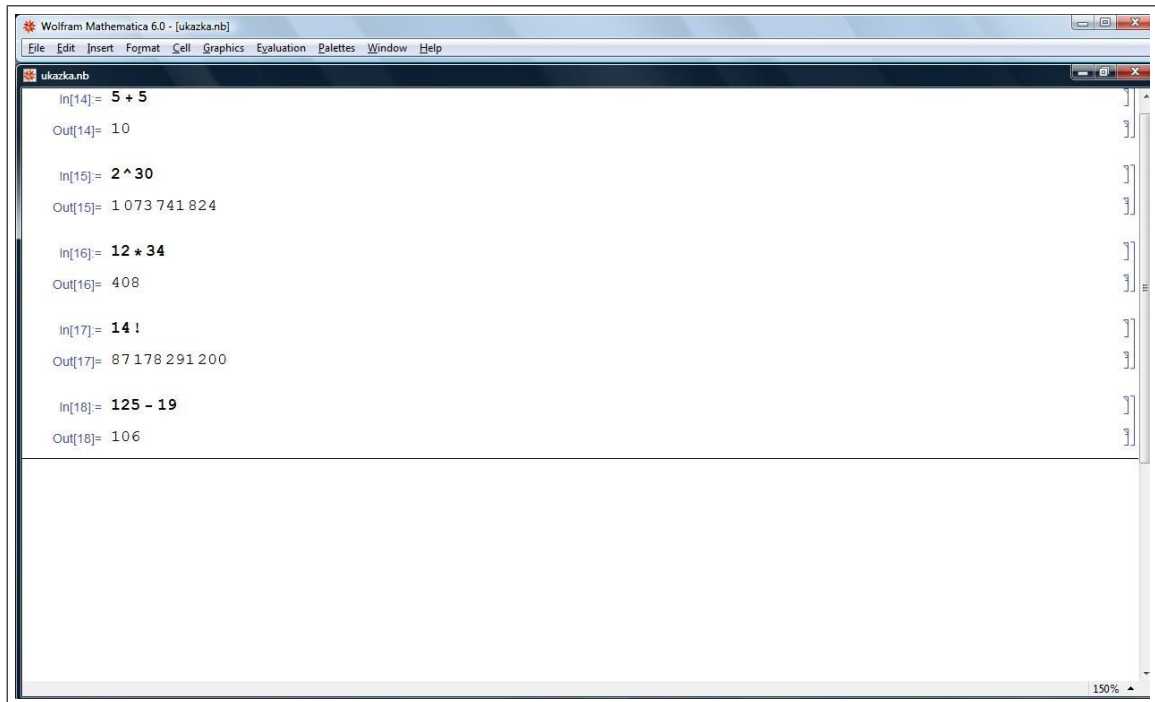
V rámci jednoho sešitu můžeme provádět větší počet různých výpočtů, grafů apod. Zběžně nyní popíšeme provádění základních matematických operací.

Základní matematické operace Napsání požadovaného řádku s řešením potvrdíme buď klávesou **Enter**, což nás posune na další řádek, nebo stiskem kláves **Shift&Enter**, což vyhodnotí námi zadaný výraz. Pro jednoduché sečtení bychom tedy zadali a obdrželi:

```
In[1] := 5+5
Out[1] = 10
```

Po zapsání výrazů $5 + 5$ program automaticky označí, že se jedná o vstup (**In[1]**) a výsledek jako výstup (**Out[1]**). Okno se zadanými operacemi může poté vypadat jako obrázek 17.

Kvůli úspoře místa zde nebudeme uvádět snímky všech obrazovek s příklady. Pokud to bude možné, použijeme úspornější zápis:



Obrázek 17: Obrazovka programu Mathematica se základními mat. operacemi

```
In[14] := 5 + 5
Out[14] = 10
In[15] := 2^30
Out[15] = 1073741824
In[16] := 12*34
Out[16] = 408
In[17] := 14!
Out[17] = 87178291200
In[18] := 125 - 19
Out[18] = 106
```

Syntaktický zápis funkcí Názvy vestavěných funkcí v programu Mathematica začínají vždy velkým písmenem, např. funkce **Plus** provede součet zadaných parametrů. Parametry jsou předávány v hranatých závorkách. Funkci **Plus** zavoláme zápisem **Plus[5,4]**. Obdržíme:

```
In[20] := Plus[2, 4]
Out[20] = 6

In[22] := Times[7, Plus[2, 1]]
Out[22] = 21
```

Funkce `Times` z předchozího příkladu provede vynásobení argumentů. Funkcí vestavěných do programu Mathematica existuje velké množství. V dalším uvidíme použití několika dalších, kompletní výčet lze nalézt v nápovědě.

Řešení rovnic Řešení rovnice lze nalézt použitím funkce `Solve`.

```
In[2] := Solve[a*x^2 + b*x + c == 0, x]
```

```
Out[2] = {{x -> (-b - Sqrt[b^2 - 4ac])/2a}, {x -> (-b + Sqrt[b^2 - 4ac])/2a}}
```

Funkce jako parametr přijímá rovnici (nebo seznam rovnic, viz dále) a jako druhý parametr pro kterou proměnnou chceme rovnici řešit. Pro zapsání podmínky se použije dvojité rovnítko `==`.

K řešení soustavy rovnic můžeme zapsat:

```
In[4] := Solve[{x^2 - 2*y == 1, x + y == 12}, {x, y}]
```

```
Out[4] = {{y -> 13 - Sqrt[26], x -> -1 + Sqrt[26]}, {y -> 13 + Sqrt[26], x -> -1 - Sqrt[26]}}
```

Rovnice má tedy kořeny $y = 13 - \sqrt{26}$, $x = -1 + \sqrt{26}$ a $y = 13 + \sqrt{26}$, $x = -1 - \sqrt{26}$.

Zápis se složenými závorkami je v programu Mathematica velmi často používán a jedná se o takzvaný „List expression“ — výraz typu seznam. Pro zjednodušení můžeme říci, že je použijeme tam, kde předáváme do jednoho parametru více hodnot.

Pro řešení soustavy rovnice v předchozí ukázce se jako seznam předaly obě rovnice ($x^2 - 2 \cdot y = 1$ a $x + y = 12$), stejně tak se jako seznam předaly proměnné (x , y).

Operace přiřazení a definice funkce Použitím jednoduchého znaku rovná se (`=`) jako například ve výrazu `g=9.81`, nebo ekvivalentně použitím vestavěné funkce `Set[g, 9.81]`, jsme použili operátor přiřazení a kdekoli dále v sešitu použijeme výraz `g`, bude dosazena hodnota 9,81. Pokud je potřeba takto nastavené přiřazení zrušit, musíme to provést explicitně voláním funkce

```
Clear[g].
```

Pro příklad operace přiřazení uveďme řešení rovnice hledající výšku geostacionární družice nad povrchem Země. Z předpokladu, že gravitační síla mezi Zemí a naší družicí je dostředivou silou která nutí družici vykonávat kruhový pohyb a toho, že doba oběhu družice musí být 23h56min4,09s, můžeme snadno sestavit rovnici

$$G \cdot \frac{M \cdot m}{(h + R)^2} = \frac{m \cdot 4 \cdot \pi^2 \cdot (h + R)}{T^2},$$

kde G, M, m, h, R, T značí pořadě gravitační konstantu, hmotnost Země, hmotnost družice, výšku družice nad povrchem — hledaná veličina, poloměr Země, dobu oběhu družice. Nalezení h bychom provedli následovně:

```
In[46] := T = 24*3600;
R = 6378000;
G = 6.672*10^-11;
M = 5.9736*10^24;
m = 500;
rovnice = G*(M*m)/(h + R)^2 == (m*4*\[Pi]^2*(h + R))/T^2;
Solve[rovnice, h]
```

```
Out[52] = {{h -> -2.74978*10^7 - 3.65806*10^7 i },
{h -> -2.74978*10^7 + 3.65806*10^7 i}, {h -> 3.58616*10^7}}
```

Řešení rovnice nesmí ležet v oboru imaginárních čísel a proto naše hledané h odpovídá třetímu řešení a tedy hodnotě $h = 35861$ km. Středníky byly v předchozím případě použity tam, kde jsme nepožadovali, aby byl řádek zopakován na výstupu. Rozdíl ukáže následující případ:

Se středníkem:

```
In[56] := x = 12;
x^2
Out[57] = 144
```

Bez středníku:

```
In[58] := x = 12
x^2
Out[58] = 12
Out[59] = 144
```

Definici funkce provedeme ale poněkud odlišným způsobem:

```
In[53] := f[x_] = x^3 + 3*x + 2;
f[9]
```

```
Out[54] = 758
```

Grafika Možnost okamžitého zobrazení nalezeného výsledku je velmi výhodná. Zkrátí čas potřebný k řešení příkladu, usnadňuje i pochopení řešení. Možnost tvorby grafů a další grafiky je v programu Mathematica bohatě podporovaná. V dalším

uvidíme, jak lze vytvořit 2-D i 3-D grafy, sloupcový graf nebo zobrazení vektorového pole.

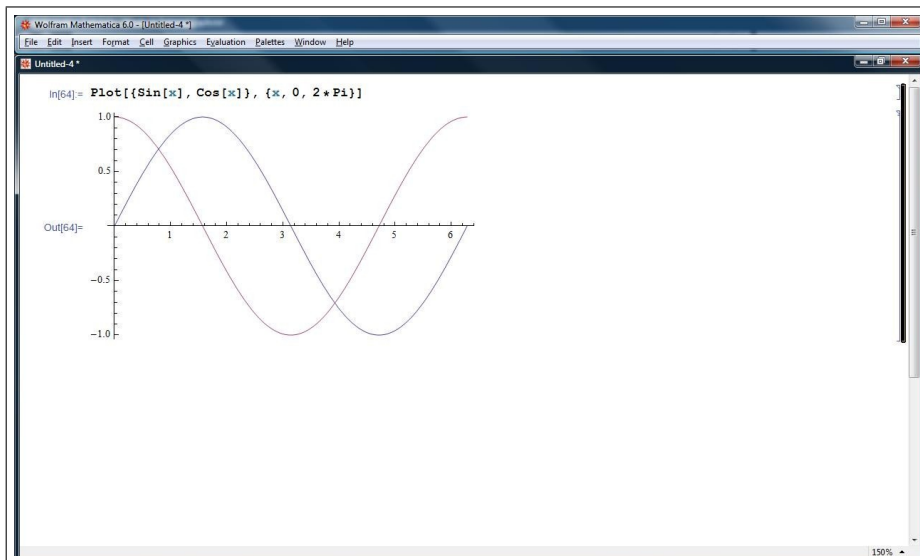
2-D graf vytvoříme voláním funkce `Plot`. Jako parametr, případně seznam parametrů, předáme funkci kterou chceme vykreslit a jako druhý parametr seznam v podobě `{proměnná, od, do}`. Pro vykreslení funkce $\sin(x)$ od 0 do 2π použijeme

```
In[63]:= Plot[Sin[x], {x, 0, 2*Pi}].
```

Podobu předání parametru ve formě seznamu demonstruje:

```
In[64]:= Plot[{Sin[x], Cos[x]}, {x, 0, 2*Pi}]
```

a dostaneme výsledek dle obrázku 18.



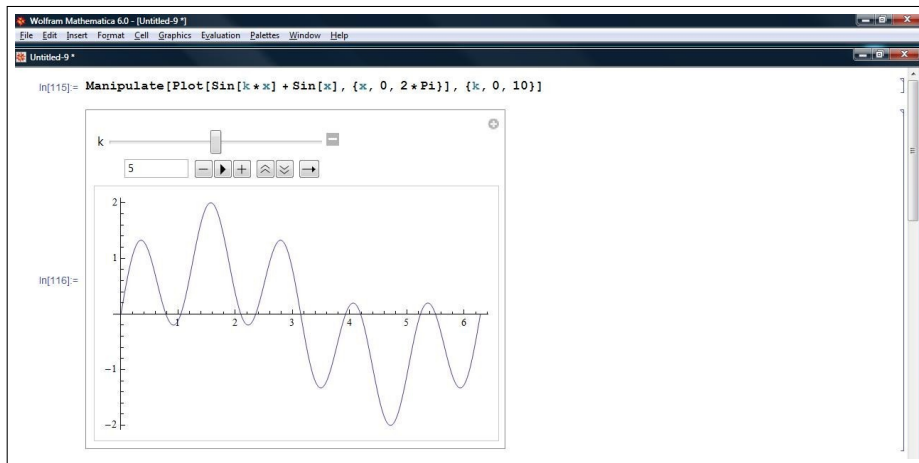
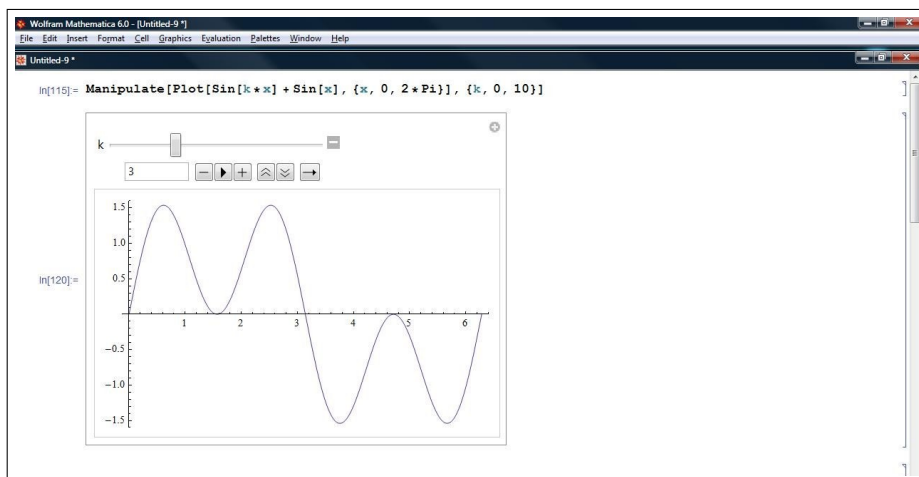
Obrázek 18: Demonstrace funkce Plot

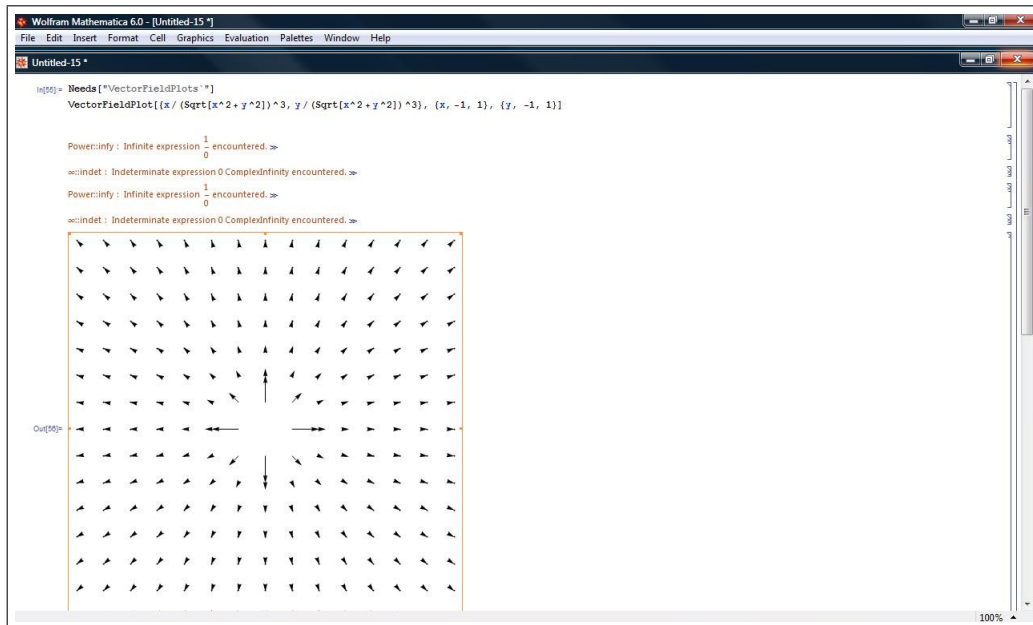
Parametricky zadanou funkci je možné zobrazit příkazem `ParametricPlot`. Parametry jsou obdobné jako v případě funkce `Plot`.

3-D graf zobrazíme funkcí `Plot3D`. Existuje i funkce `ParametricPlot3D` s obdobnými parametry jako předchozí funkce.

Na tomto místě je vhodné zmínit se i o funkci `Manipulate`. Funkce umožňuje změnu jednoho nebo více parametrů zadaného grafu, či jiného výpočtu, a bez přepisování vstupního zadání sledovat závislost výstupu na tomto parametru. To je výhodné právě u grafů. Chceme-li například pozorovat skládání funkcí $\sin(x)$ a $\sin(k \cdot x)$ s proměnným parametrem k , zadáme vstup:

```
In[1]:= Manipulate[Plot[Sin[x] + Sin[k*x], {x, 0, 2*Pi}], {k, 0, 10}]
```

Obrázek 19: Funkce $\sin(x) + \sin(5 \cdot x)$ Obrázek 20: Funkce $\sin(x) + \sin(3 \cdot x)$



Obrázek 21: Pole vektoru intenzity elektrostatického pole

Na obrazovce se objeví posuvník umožňující plynulou změnu parametru k a graf na obrazovce je okamžitě překreslován. Výsledek pro dvě různé polohy posuvníku ukazuje obrázek 19 a 20.

Při studiu lineární algebry je dobré mít možnost přímého zobrazení vektorových polí. Je pak o mnoho snazší vyložit pojmy *rotace* nebo *divergence*. Zobrazení vektorových polí samozřejmě nabízí i program Mathematica. Poskytuje k tomu funkci `VectorFieldPlot`. Před použitím této funkce je potřeba zavést balík `VectorFieldPlots` jako v následujícím příkladu:

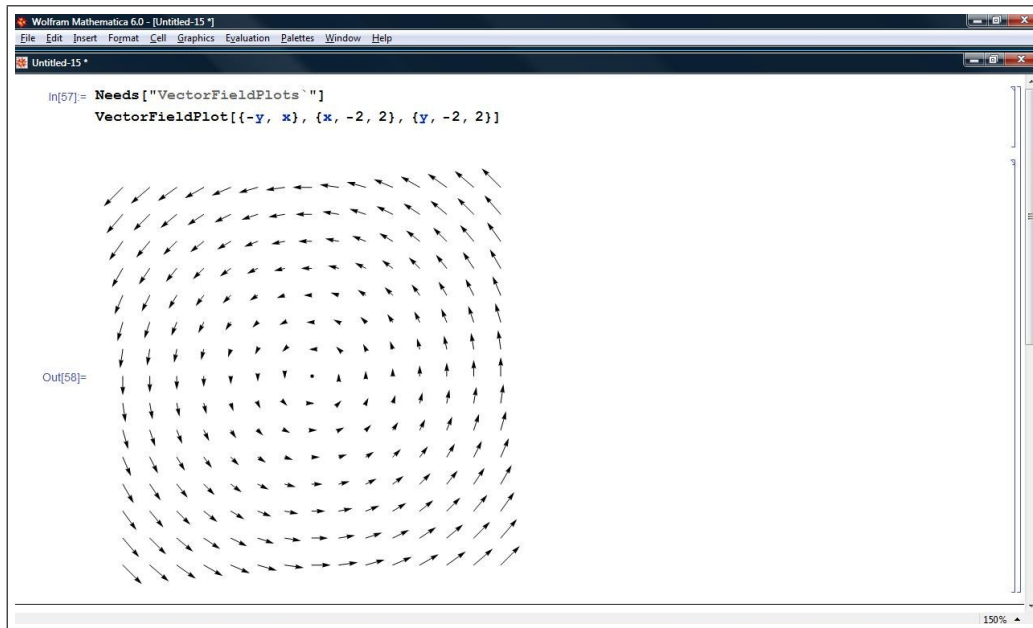
```
In[55]:= Needs["VectorFieldPlots"]
VectorFieldPlot[{x/(Sqrt[x^2 + y^2])^3, y/(Sqrt[x^2 + y^2])^3},
  {x, -1, 1}, {y, -1, 1}]
```

Předchozí příklad lze využít při popisu elektrostatického pole, například zobrazit vektorové pole vektoru intenzity elektrického pole $|\vec{E}| \sim \frac{1}{r^2}$. Pole je zobrazeno na obrázku 21. Obrázek přehledně ukazuje, že elektrostatické pole je polem nevírovým a obsahuje zdroje pole (v tomto případě elektrický náboj $+Q$ v počátku souřadného systému).

Vírové pole lze pak ukázat na vektorovém poli $\vec{K} = (-y, x)$, obrázek 22.

Grafům lze nastavit mnoho parametrů: barvu, styl čáry, popis os a mnoho dalších. Jejich popis přesahuje rámec této práce a všechny jsou přehledně popsány v nápovědě programu Mathematica.

Další příklady použití grafů budou uvedeny na konci této kapitoly.



```
Integrate[Sin[omega*t], {t, 0, 5}]
```

```
Out[27]= Sin[25]^2/5
```

```
In[28]:= N[%]
```

```
Out[28]= 0.0035034
```

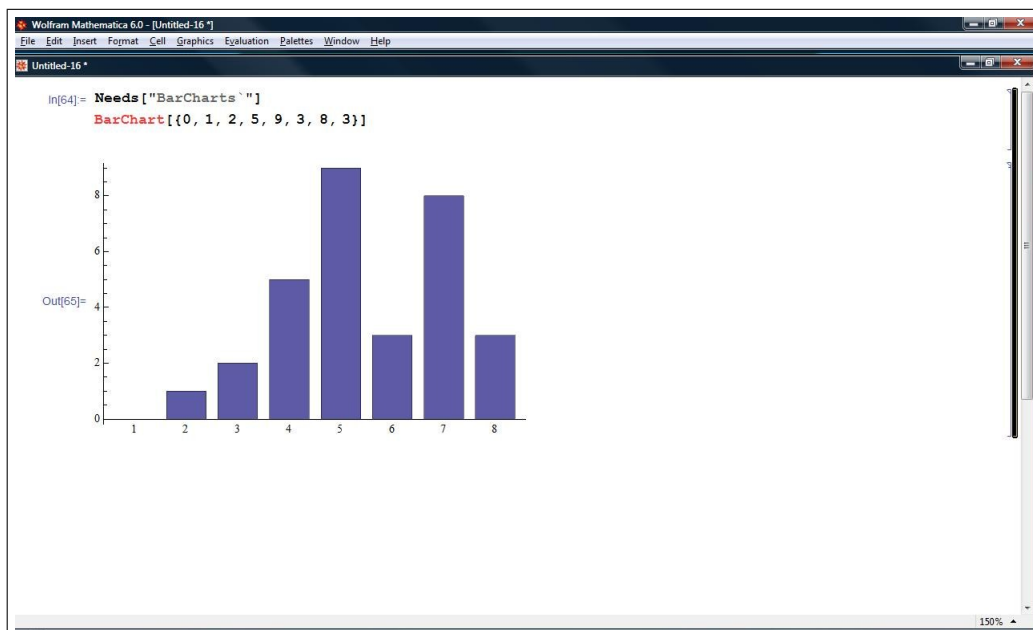
V příkladu jsme jako hodnotu ω zvolili hodnotu 10 a na předposledním řádku zadáním `N[%]` provedli vyčíslení předcházejícího řádku (předchozí řádek zastupuje symbol `%` a funkce pro vyčíslení je `N[]`). Vodičem by tedy prošel náboj $Q = 0,0035$ C.

V programu Mathematica je možné nakreslit i sloupcový graf funkcí `BarChart`, například:

```
In[64]:= Needs["BarCharts`"]
```

```
BarChart[{0, 1, 2, 5, 9, 3, 8, 3}]
```

vede k výsledku podle obrázku 23.



Obrázek 23: Sloupcový graf

Vektory K často používaným operacím ve fyzice patří operace s vektory. Vektor v programu Mathematica zapíšeme jako výraz typu seznam, např. `{0, 2, 3}` definuje vektor $0 \cdot \vec{i} + 2 \cdot \vec{j} + 3 \cdot \vec{k}$. Operace sčítání, skalárního součinu a vektorového součinu ukazuje následující výpis:

```
In[71]:= vektor1 = {2, 0, 5};
```

```
vektor2 = {3, 1, 5};
```

```
In[73]:= vektor1 + vektor2
In[74]:= vektor1*vektor2
In[75]:= vektor1\[Cross]vektor2
```

```
Out[73]= {5, 1, 10}
Out[74]= 31
Out[75]= {-5, 5, 2}
```

Na závěr této kapitoly uveďme několik prakticky využitelných příkladů použití balíku Mathematica ve výuce fyziky.

3.2 Kompletní příklad I: Lissajousovy obrazce

S výkladem Lissajousových obrazců se studenti běžně setkávají při výuce kmitů a jejich skládání již na střední, nebo i na základní škole. Jedná se o obrazce, které získáme složením dvou harmonických kmitů ve dvou na sebe kolmých směrech.

Pro studenty s lepší znalostí matematiky lze ukázat i matematický popis řešení problému. Obecně lze polohu kmitajícího bodu popsat rovnicemi

$$A \cdot \sin(\omega \cdot t + \varphi),$$

kde A je amplituda kmitavého pohybu, ω je úhlová frekvence, t čas a φ fáze kmitavého pohybu.

V případě kmitání bodu ve dvou směrech (ose x a y) bodu můžeme psát:

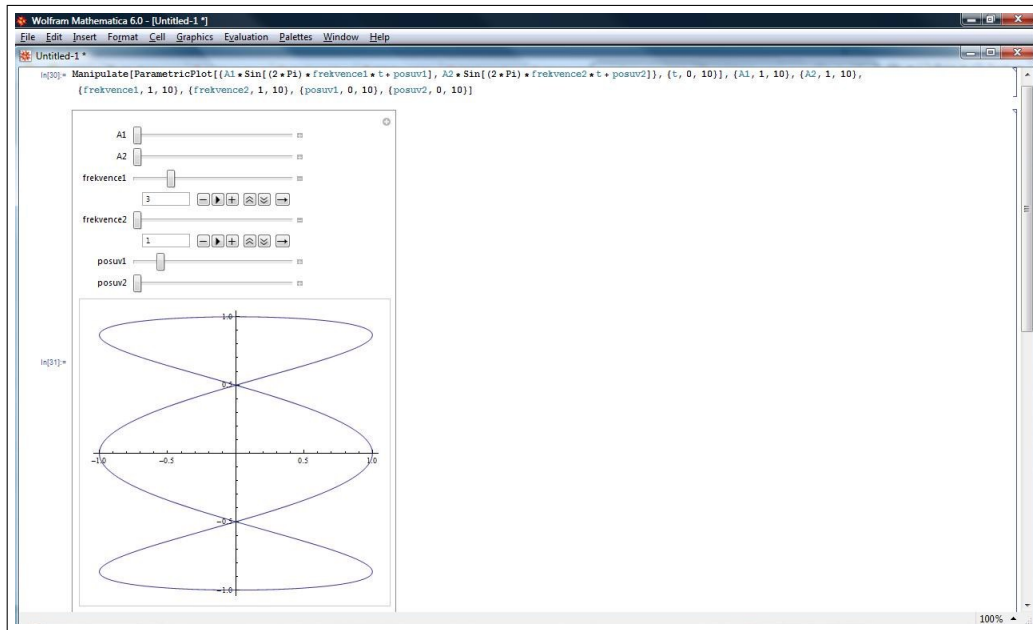
$$\begin{aligned} x &= A_1 \cdot \sin(\omega_1 \cdot t + \varphi_1) \\ y &= A_2 \cdot \sin(\omega_2 \cdot t + \varphi_2) \end{aligned}$$

Tyto rovnice mohou být považovány za parametrické rovnice trajektorie kmitajícího bodu. Jejich zobrazení je možné na obrazovce osciloskopu, případně pomocí historických přístrojů k tomuto účelu určených (Blackburnovo kyvadlo), nebo, jako tomu bude v tomto případě, na obrazovce počítače.

Požadujeme, aby ve výsledku bylo možné manipulovat se všemi proměnnými z předchozích rovnic (amplitudou, fázovým posuvem i frekvencí kmitů). V programu Mathematica toho dosáhneme následujícím zápisem:

```
In[19]:= Manipulate[
  ParametricPlot[{A1*Sin[frekvence1*t + posuv1],
    A2*Sin[frekvence2*t + posuv2]}, {t, 0, 10}], {A1, 1, 10}, {A2, 1,
```

10}, {frekvence1, 1, 10}, {frekvence2, 1, 10}, {posuv1, 0, 10}, {posuv2, 0, 10}]



Obrázek 24: Lissajoussův obr. pro poměr frekvencí 1:3 a fázový posuv $\pi/2$

Výsledné obrazce lze vidět na obrázcích 24 a 25. Obrázek 24 ukazuje situaci pro poměr frekvencí 1:3 a fázový posuv $\pi/2$ a obrázek 25 pro poměr frekvencí 1:2 a posuv $\pi/3$.

3.3 Kompletní příklad II: Částice v jednorozměrné potenciálové jámě

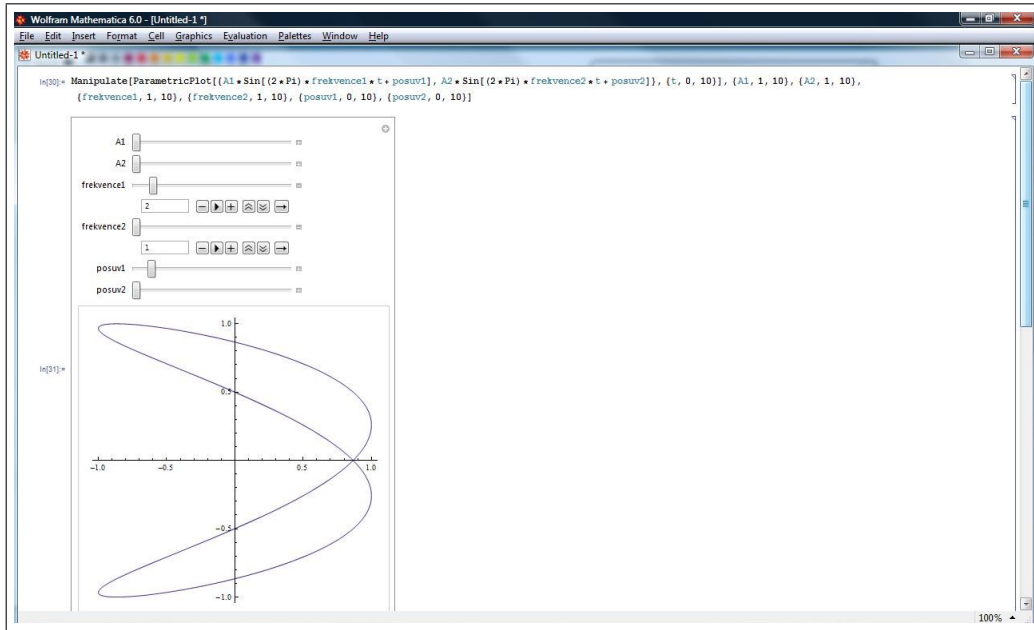
Řešením Schrödingerovy rovnice pro elektron v nekonečně hluboké potenciálové jámě s šířkou L nalezneme vlnovou funkci ve tvaru

$$\psi_n(x) = A \cdot \sin\left(\frac{n \cdot \pi}{L} \cdot x\right),$$

kde A , n , L , x jsou pořadě v dalším určená konstanta, kvantové číslo $n = 1, 2, 3, \dots$ a poloha částice v jámě. Vlnová funkce nemá sama o sobě přímou fyzikální interpretaci, kvantová mechanika však ukazuje, že veličina $|\psi|^2$, neboli modul vlnové funkce umocněný na druhou stanovuje *hustotu pravděpodobnosti výskytu*.

V případě naší jednorozměrné potenciálové jámy můžeme vyjádřit pravděpodobnost nalezení částice mezi body x a $x + dx$ jako $\psi_n^2(x)dx$.

Součástí hodiny pak může být zobrazení průběhu hustoty pravděpodobnosti v jasné, snadno pochopitelné podobě. Snáze pak vysvětlíme, že na rozdíl od klasické



Obrázek 25: Lissajoussův obr. pro poměr frekvencí 1:2 a fázový posuv $\pi/3$

představy není pravděpodobnost nalezení částice ve všech bodech pasti stejná. Před zobrazením funkce hustoty pravděpodobnosti však obvykle aplikujeme na nalezenou hustotu pravděpodobnosti takzvanou *normovací podmínku*. Ta říká, že pravděpodobnost nalezení částice někde v prostoru celé jámy se musí rovnat jedné. Matematické vyjádření normovací podmínky zapíšeme jako:

$$\int_{-\infty}^{\infty} \psi_n^2(x) dx = 1$$

Jelikož v našem příkladu pracujeme s nekonečně hlubokou potenciálovou jámou můžeme výše uvedenou podmínku zjednodušit na

$$\int_0^L \psi_n^2(x) dx = 1$$

protože částice se mimo jámu vyskytovat nemůže. Pro nalezení normovací podmínky tedy v našem případě hledáme integrál

$$\int_0^L \psi_n^2(x) dx = \int_0^L A^2 \cdot \sin^2\left(\frac{n \cdot \pi}{L} \cdot x\right) dx$$

a po položení jeho hodnoty rovno jedné hledáme hodnotu A , dosud neurčenou konstantu odpovídající amplitudě vlnové funkce. Nalezení hodnoty tohoto integrálu není nijak složité, ale pro ilustraci ji můžeme najít v programu Mathematica. Použijeme následující zápis:

```
In[136]:= VlnovaRovnice = A*Sin[(n*Pi)/L*x];
integral = Integrate[A^2*Sin[(n*Pi)/L*x]^2, {x, 0, L}]
```

```
Out[137]= 1/4 A^2 L (2 - Sin[2 n \[Pi]]/(n \[Pi]))
```

V zápisu jsme použili několik věcí, jež byly v předešlém textu již popsány: na prvním a druhém řádku operaci přiřazení, na druhém vyčíslení určitého integrálu v mezích od 0 do L . Výsledkem integrace je

$$\frac{1}{4} \cdot A^2 \cdot L \left(2 - \frac{\sin(2\pi n)}{n \cdot \pi} \right).$$

My víme, že pro n dosazujeme přirozená čísla $n = 1, 2, \dots$, hodnota funkce $\sin(2 \cdot n \cdot \pi)$ bude proto vždy nula. Proto pro nalezení hodnoty A řešíme pouze rovnici

$$\frac{1}{2} \cdot A^2 \cdot L = 1.$$

Pro ilustraci řešíme opět v programu Mathematica a celý zápis je:

```
In[136]:= VlnovaRovnice = A*Sin[(n*Pi)/L*x];
integral = Integrate[A^2*Sin[(n*Pi)/L*x]^2, {x, 0, L}]
```

```
Out[138]= 1/4 A^2 L (2 - Sin[2 n \[Pi]]/(n \[Pi]))
```

```
In[139]:= Solve[1/2*A^2*L == 1, A]
```

```
Out[139]= {{A-> -(Sqrt[2]/Sqrt[L])}, {A-> Sqrt[2]/Sqrt[L]}}
```

Pro konstantu A jsme tedy našli hodnotu $A = \sqrt{\frac{2}{L}}$. Celá vlnová rovnice má pak tvar

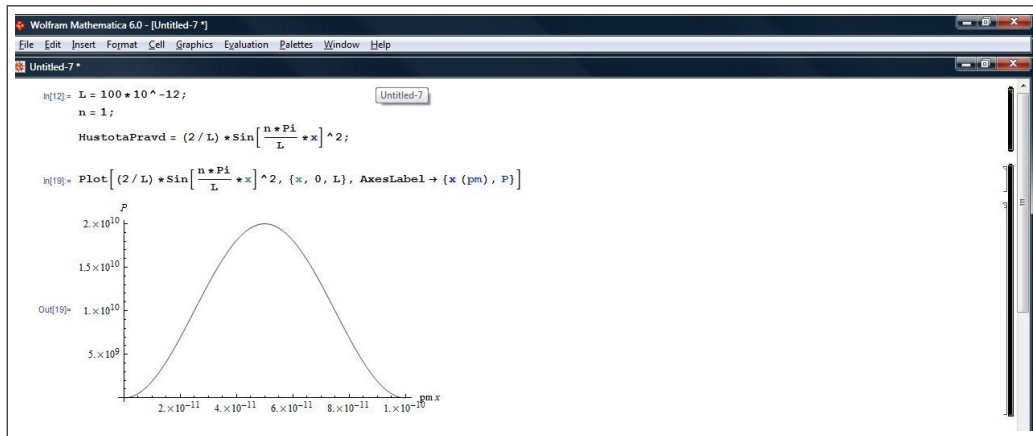
$$\psi_n(x) = \sqrt{\frac{2}{L}} \cdot \sin\left(\frac{n \cdot \pi}{L} \cdot x\right),$$

tuto rovnici chceme zobrazit. To umožní zápis:

```
In[12]:= L = 100*10^-12;
n = 1;
HustotaPravd = (2/L)*Sin[(n*Pi)/L*x]^2;
```

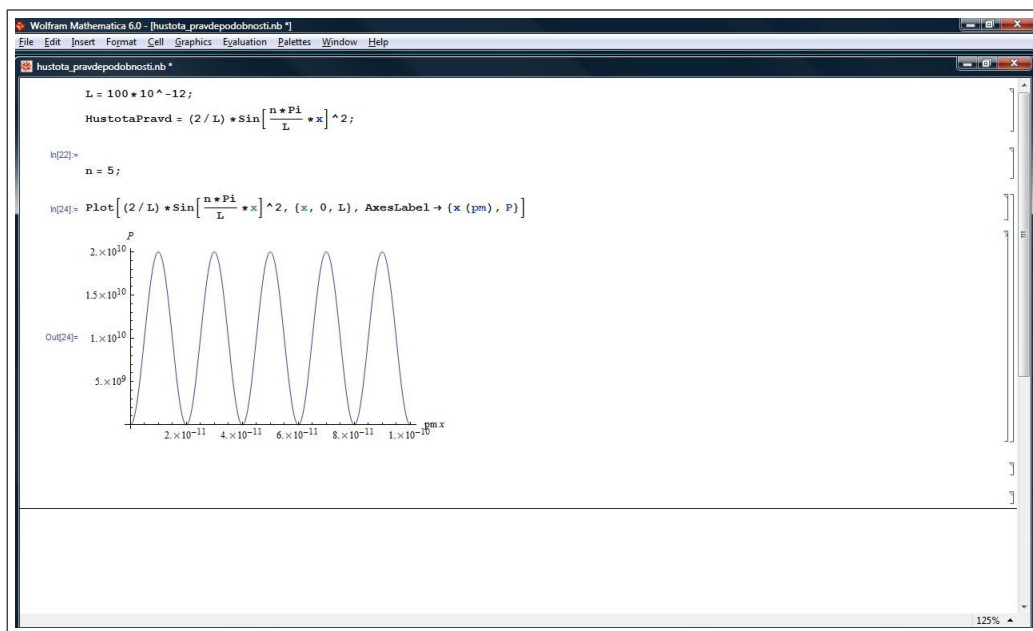
```
In[19]:= Plot[(2/L)*Sin[(n*Pi)/L*x]^2, {x, 0, L},
  AxesLabel -> {x (pm), P}]
```

Výsledkem jsou obrázky 26 a 27. Obrázek 26 pěkně ukazuje, že oproti představě klasické mechaniky není pravděpodobnost nalezení částice kdekoli v jámě stejná, ale



Obrázek 26: Hustota pravděpodobnosti výskytu částice v jednorozměrné potenciálové jámě šířky 100 pm, kvantové číslo 1

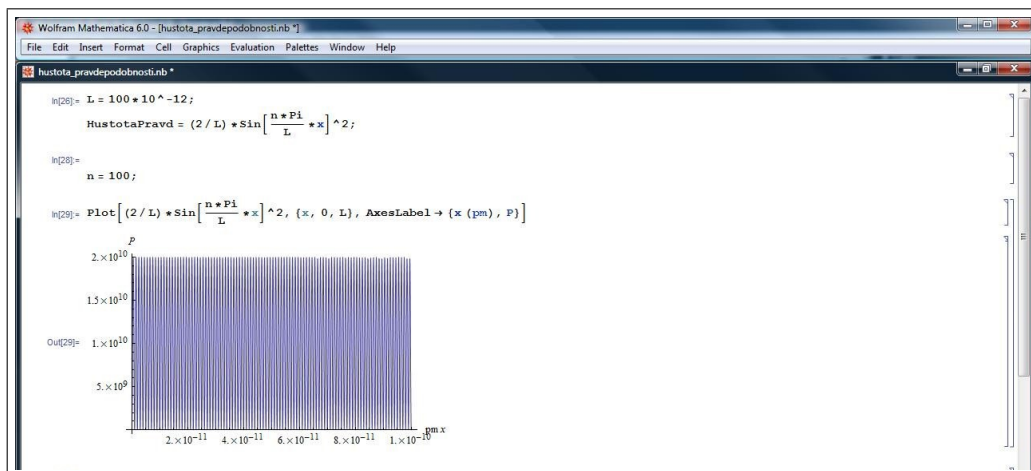
největší pravděpodobnost výskytu částice je v polovině potenciálové jámy (souřadnice $x = 50$ pm).



Obrázek 27: Hustota pravděpodobnosti výskytu částice v jednorozměrné potenciálové jámě šířky 100 pm, kvantové číslo 5

A na závěr lze ilustrovat princip korespondence, který říká, že pro velmi vysoká kvantová čísla se výsledky kvantové mechaniky musí blížit výsledkům fyziky klasické. Obrázek 28 tak ukazuje hustotu pravděpodobnosti výskytu částice s kvantovým číslem 100. Vidíme, že pravděpodobnost nalezení částice ve všech místech jámy je již prakticky stejná.

Program Mathematica tak mohl významně ulehčit a přehledně znázornit výklad tohoto kvantově mechanického jevu.



Obrázek 28: Hustota pravděpodobnosti výskytu částice v jednorozměrné potenciálové jámě šířky 100 pm, kvantové číslo 100

3.4 Kompletní příklad I: 3-D zobrazení intenzity a potenciálu elektrického pole

Mathematica podporuje i práci s 3-D (prostorovými) grafy. Jako příklad ukažme graf intenzity elektrického pole v okolí bodového náboje ve vakuu. Velikost intenzity elektrického pole můžeme vyjádřit jako:

$$E = \frac{1}{4 \cdot \pi \cdot \epsilon_0} \cdot \frac{Q}{r^2},$$

kde ϵ_0 je permitivita vakua, r vzdálenost mezi místem ve kterém určujeme intenzitu a bodovým nábojem.

Vykreslení grafu dosáhneme v programu Mathematica zápisem:

`Plot3D[(1/4*Pi*8.854 E - 12)/((x^2 + y^2)^2), {x, -2, 2}, {y, -2, 2}].`

Výsledkem je obrázek 29. „Sloup“ uprostřed obrázku je zobrazení intenzity el. pole rostoucí nade všechny meze, v případě přibližování se k tomuto bodovému náboji na velmi malou vzdálenost. Sloup by měl být samozřejmě „nekonečně“ vysoký. Byl zobrazen v takovém měřítku, aby byla vidět s druhou mocninou vzdálenosti klesající hodnota velikosti intenzity elektrického pole.

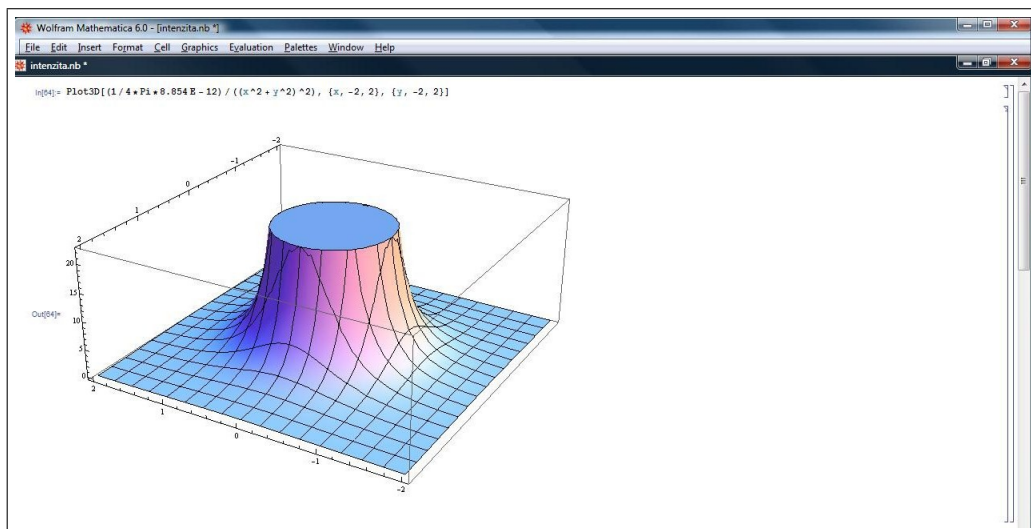
Pro úplnost ještě připojme graf průběhu potenciálu tohoto náboje daného vzorcem

$$\phi = -\frac{1}{4 \cdot \pi \cdot \epsilon_0} \cdot \frac{Q}{r}.$$

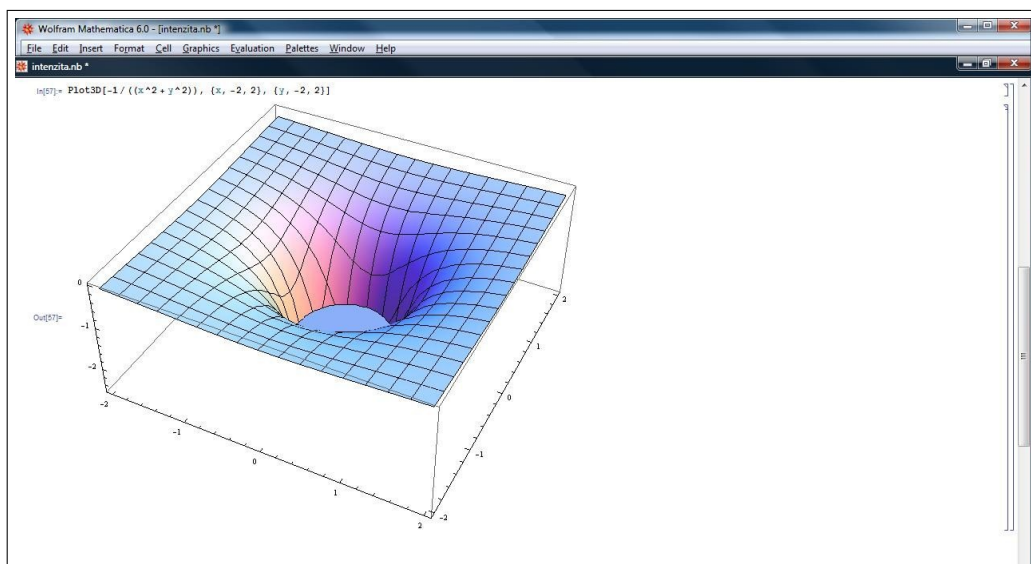
Vykreslení grafu dosáhneme v programu Mathematica zápisem:

`Plot3D[(1/4*Pi*8.854 E - 12)/((x^2 + y^2)), {x, -2, 2}, {y, -2, 2}].`

a obdržíme obrázek 30.



Obrázek 29: Intenzita elektrického pole v okolí jednotkového náboje ve vakuu



Obrázek 30: Potenciál elektrického pole v okolí jednotkového náboje ve vakuu

4 Využití počítače jako měřicího přístroje

Možnostmi použití počítače jako měřicího přístroje, konkrétně jako prostředku pro zpracování a získání experimentálních dat, se zabývá tato kapitola. Úkolem bude nalézt řešení umožňující zkvalitnění měření vybraných úloh fyzikálního praktika.

4.1 Popis současného stavu

První dva pokusy (*Poissonovo rozdělení* a *Dolet částic beta*) jsou v současné době zařazeny jako standardní součást fyzikálního praktika na katedře obecné fyziky Pedagogické fakulty Západočeské univerzity v Plzni. Pro detekci beta částic je použit Geigerův–Müllerův počítač, respektive scintilační detektor.

Hodnoty získané měřením bylo nutné zadat do počítače, kde běžel program určený pro zpracování té které úlohy. Programy byly určeny pro prostředí MS-DOS (operační systém již mnoho let pro normální účely nepoužívaný) a trpěly obvyklými neduhy programů pro tento operační systém.

Na rozdíl od programů pro operační systém Windows nepoužívaly programy pro MS-DOS standardní ovládací prvky, celá práce se tím znepráhledňovala a uživatel nemohl použít obvyklé postupy na které byl zvyklý z ostatních programů. Při změně jedné zadávané hodnoty bylo nutné začít zadávat všechny hodnoty znovu od začátku.

Dalším nežádoucím znakem programů pro MS-DOS je také neprůhledná spolupráce s počítačovými perifériemi — týká se například tisku z programu.

Konečně MS-DOS již není možné na většině moderních počítačů vůbec spustit. Ačkoli je možné ve Windows spouštět programy v takzvaném emulovaném módu, je tato možnost poměrně omezující z důvodu rozdílného programovacího modelu operačních systémů MS-DOS a Windows. Windows je již od návrhu operační systém, jež umožňuje provozovat více programů současně (takzvaný preemptivní multitasking). Pro zachování robustnosti systému musí Windows důsledně chránit jeden program před druhým — chyba v jednom z programů nesmí ohrozit činnost programu druhého. Proto je zamezen přímý programový přístup k hardware počítače. Toto a mnoho dalších omezení často vyústí v nemožnost některé programy určené pro MS-DOS úplně provozovat.

Pro ilustraci můžeme zmínit celkem obvyklou praxi programů pro MS-DOS — přistupovat přímo k videopaměti zobrazovacího zařízení počítače. Zápisem dat do videopaměti byl údaj zobrazen na obrazovce. Při pokusu o spuštění takového programu v prostředí Windows, program havaruje s chybovým hlášením o pokusu o přímý přístup k hardware, jež není ve Windows povoleno.

Přímý přístup k hardware v prostředí Windows je sice možný, ale pouze prostřednictvím vývoje vlastního ovladače zařízení. Tato činnost zdaleka překračuje obvyklé

„uživatelské“ programování a kromě pokročilých znalostí vnitřních zákonitostí operačního systému je nutné vlastnit i speciální programové vybavení (Windows DDK — Drivers Development Kit).

Abychom učinili vyvážený úsudek, musí být ovšem jmenovány i výhody programů pro MS-DOS, které prostředí Windows odpírá. Patří sem právě již výše zmíněná možnost programů přistupovat přímo k hardware počítače. Jak bylo řečeno, tato vlastnost je nebezpečná, může ohrozit ostatní běžící programy, ale na druhé straně je možné dosáhnout vyšší rychlosti práce s příslušným hardwarovým zdrojem. K výhodám patří i hardwarová nenáročnost MS-DOS programů.

Ze všech výše popsaných důvodů vyvstala nutnost vytvořit software schopný práce v prostředí Microsoft Windows.

Tato práce se zabývá také popisem programového řešení odstraňujícího nedostatky dříve používaných programů.

4.1.1 Poissonovo rozdělení

Úkolem úlohy bylo ověřit platnost Poissonova rozdělení pro částice beta. Aby měřením vzniklo Poissonovo, nikoli Gaussovo rozdělení bylo nutné provést měření s nízkou intenzitou zářiče, nebo měřit ve velmi krátkých časových intervalech. V této úloze se proto pracovalo pouze s přirozeným pozadím. Pro získání statisticky významného vzorku dat ovšem bylo nutné měření mnohokrát opakovat.

V konečném důsledku to znamenalo provádět zápis hodnoty každých pět sekund a měření opakovat čtyřikrát. Takto získané hodnoty musel potom student přepsat do počítače, kde byl spuštěn program určený pro zpracování výsledků z tohoto měření.

Kvůli značné délce měření a nevhodnému programovému vybavení nebylo možné úlohu *Poissonovo rozdělení* provádět jako demonstrační pokus.

Díky této práci bude nadále možno úlohu demonstračně předvádět. Větší rychlosti měření bylo dosaženo použitím radioaktivního zářiče a nový obslužný software zobrazuje změřené hodnoty ihned bez jakékoli nutnosti zásahu obsluhy.

4.1.2 Dolet beta částic

V úloze studenti měřili dolet beta částic v aluminiových plíscích. Zachycené částice byly počítány čítačem, hodnoty bylo nutné po změření přepsat do počítače, kde běžel program pro zpracování výsledků z tohoto měření. Jednalo se celkem o dvacet sedm měřených údajů. Program provedl zpracování a vtištění výsledků.

Na obslužný program můžeme uplatnit všechny výtky jako v předchozím případě. Při měření úlohy bylo potřeba manuálně hledat tečnu křivky změřených hodnot. Kvůli méně citlivému ovládání nebyly výsledky příliš přesné.

Program vyvinutý v této verzi přináší vyšší uživatelský komfort a umožňuje též získání přesnějších výsledků díky citlivějšímu ovládání grafu.

4.1.3 Zpětný rozptyl beta částic

Tato úloha nebyla ve standardním fyzikálním praktiku vůbec zařazena. Byla předváděna v rámci *Letní školy fyziky*.

Nyní lze úlohu snadno zařadit do měřených úloh.

4.2 Poissonovo rozdělení

4.2.1 Teorie úlohy *Poissonovo rozdělení*

Počet impulzů zaregistrovaných Geigerovým–Müllerovým počítačem za určitou dobu při konstantním zdroji záření podléhá statistickým fluktuacím.

Je-li registrovaný počet impulzů velký, je statistické rozdělení, kterému počet částic podléhá, rozdělením Gaussovým. Pro něj je typická rovnost střední a nejpravděpodobnější hodnoty, a vůči této hodnotě je rozdělení symetrické.

Pro malý počet částic ovšem toto rozdělení přechází v rozdělení Poissonovo. Nejpravděpodobnější hodnota tohoto rozdělení je menší než hodnota střední, rozdělení není symetrické.

Pravděpodobnost $P(N)$, že za určitou dobu naměříme počet impulsů N , je dána vztahem

$$P(N) = e^{-p} \cdot \frac{p^N}{N!},$$

kde p je střední hodnota naměřeného počtu impulsů za určitou dobu.

Aby byl střední počet naměřených impulzů malý a aby se dosáhlo statisticky významné četnosti jednotlivých počtů, je nutné, aby záření detekované Geigerovým–Müllerovým počítačem mělo nízkou aktivitu nebo doba měření musí být malá. Typicky se proto nepoužívá radioaktivní zářič, ale pracuje se pouze s pozadím.

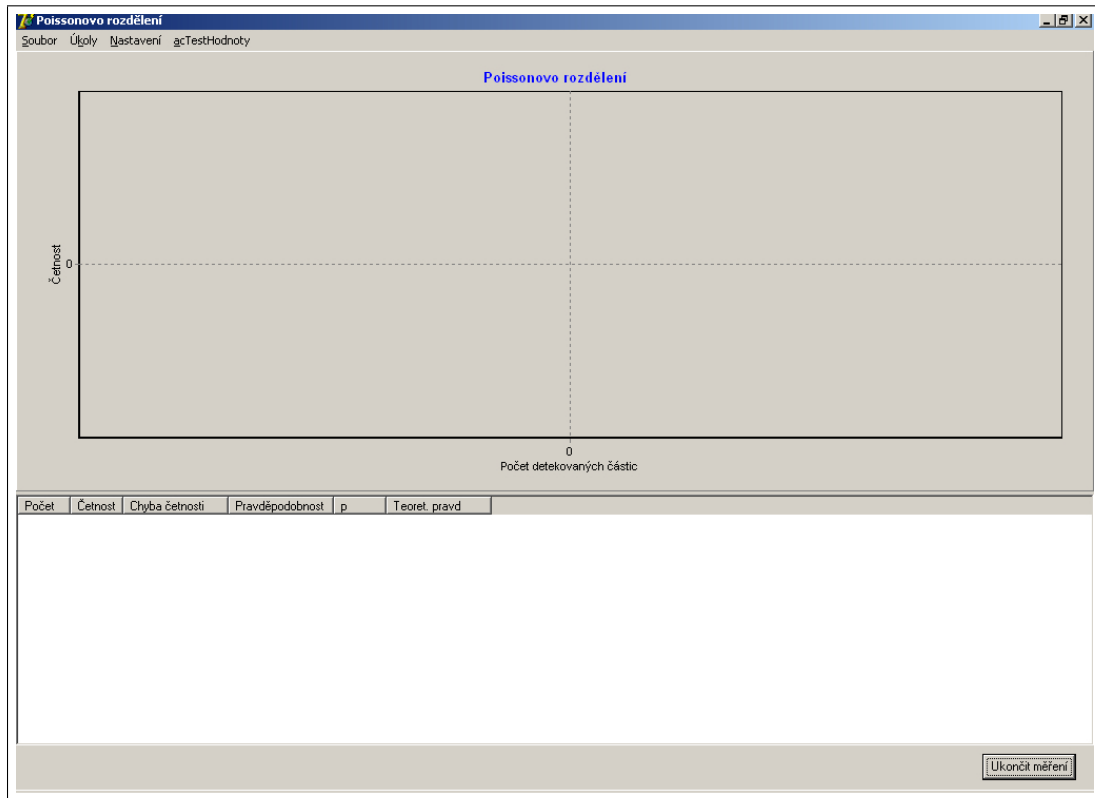
4.2.2 Ovládání programu pro úlohu *Poissonovo rozdělení*

Ovládací program pro úlohu *Poissonovo rozdělení* je umístěn v souboru *Poisson.exe*.

Základní ovládání blíže zmiňuje kapitola 4.5.3. Na tomto místě bude blíže popisováno specifické ovládání pro program sloužící k měření úlohy *Poissonovo rozdělení*.

Hlavní obrazovka ovládacího programu úlohy *Poissonovo rozdělení* je na obrázku 31.

Do horní části hlavní obrazovky bude v průběhu měření dynamicky vykreslován graf, ve spodní části, v bílé tabulce, jsou v průběhu měření zobrazovány hodnoty. Poměr prostoru, který na hlavním okně zabírá graf a tabulka, lze změnit svislým

Obrázek 31: Hlavní obrazovka programu *Poissonovo rozdělení*

tažením horizontální dělicí čáry mezi nimi. To je užitečné při větším množství naměřených hodnot, kdy by se hodnoty v malé tabulce stávaly nepřehledné. Pokud je tabulka větší než jí přidělený prostor, lze v ní rolovat svislým posuvníkem v pravé části okna.

Na vodorovnou osu grafu je vynášen počet impulsů za nastavený časový interval — hodnota N , na svislou osu četnost impulsů M_N a na druhou svislou osu pravděpodobnost $P(N)_{NAM}$ v odpovídajícím měřítku. Graf je sloupcový a zobrazuje naměřenou četnost zachycení určitého počtu částic s vyznačenou pravděpodobnou chybou ϑ_{M_N} a teoretickou závislost $P(N)_{NAM}$ na N — tedy Poissonovo rozdělení. Z definice pravděpodobné chyby je zřejmé, že bude protínat asi polovinu úseček ohrazených chybou ϑ_{M_N} .

Sloupce v spodní tabulce mají následující význam:

Název sloupce	Označení	Význam
Počet	N	Počet částic zachycený v měřeném intervalu
Četnost	M_N	Určuje, kolikrát byl počet částic ze sloupce <i>Počet</i> zachycen
Chyba četnosti	ϑ_{M_N}	Chyba četnosti ze sloupce <i>Četnost</i> dle rovnice 2
Pravděpodobnost	$P(N)_{NAM}$	Experimentálně zjištěná pravděpodobnost naměření počtu částic ze sloupce <i>Počet</i> dle rovnice 3
Teoret. pravd.	$P(N)$	Teoreticky zjištěná pravděpodobnost naměření počtu částic ze sloupce <i>Počet</i> dle rovnice 4

Pravděpodobná chyba četnosti je počítána podle vzorce

$$\vartheta_{M_N} = \frac{2}{3} \cdot \sqrt{M_N}. \quad (2)$$

Naměřená pravděpodobnost $P(N)_{NAM}$ je počítána podle vzorce

$$P(N)_{NAM} = \frac{M_N}{\sum_N M_N}. \quad (3)$$

Teoretická pravděpodobnost P_N je počítána z

$$P(N) = e^{-p} \cdot \frac{p^N}{N!}. \quad (4)$$

p je počítáno ze vztahu pro střední hodnotu

$$p = \frac{\sum_N N M_N}{\sum_N M_N}.$$

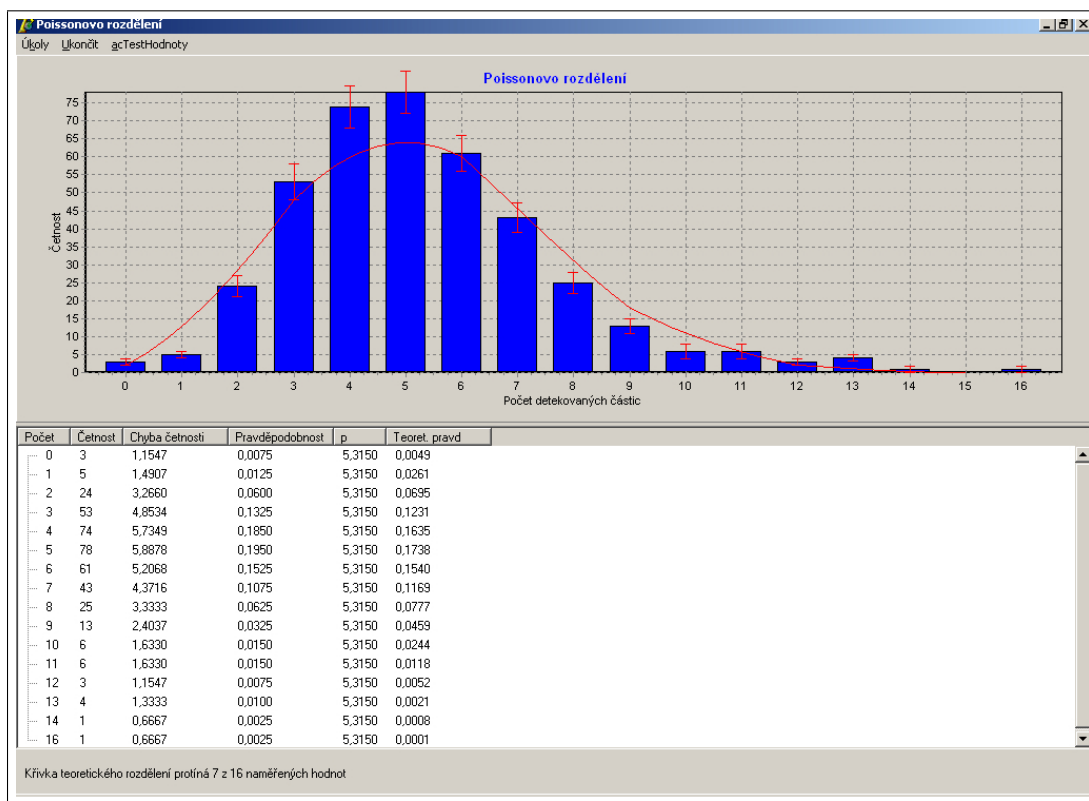
Měření a zpracování probíhá tímto způsobem: Program počítá částice zachycené za určitý interval (nastavitelný uživatelem — typicky pět sekund). Vždy po uplynutí tohoto časového intervalu je získaná hodnota zobrazena v sloupcovém grafu v horní části obrazovky. Příslušně přepočítané hodnoty jsou vypisovány do spodní tabulky.

Po každém překreslení grafu program automaticky přepočítá vhodné měřítko pro vykreslení Poissonova rozdělení do grafu a pod tabulku ve spodní části obrazovky vypíše, kolik úseček ohraničených chybou ϑ_{M_N} křivka rozdělení protíná.

Do grafu závislosti M_N na N je ve správném měřítku vynášena křivka teoretického rozdělení a počítá, kolikrát protнула chybové úsečky. V úvodu bylo řečeno, že by tato hodnota měla být asi polovina celkového počtu měření.

Podmínkou pro změření užitečných hodnot je správné stanovení doby měření.

Měřený interval nesmí být příliš dlouhý. Počet částic zachycených v jednom cyklu nesmí být velký, Poissonovo rozdělení pak přechází v rozdělení Gaussovo. Počet

Obrázek 32: Obrazovka programu *Poissonovo rozdělení* s naměřenými hodnotami

částic kdy k takovému přechodu dochází je přibližně devět.

Ve fyzikálním praktiku nebude k měření použit zářič, pracuje se pouze s pozadím. Jako vhodnou dobu jednoho měření lze v tomto případě doporučit pět sekund.

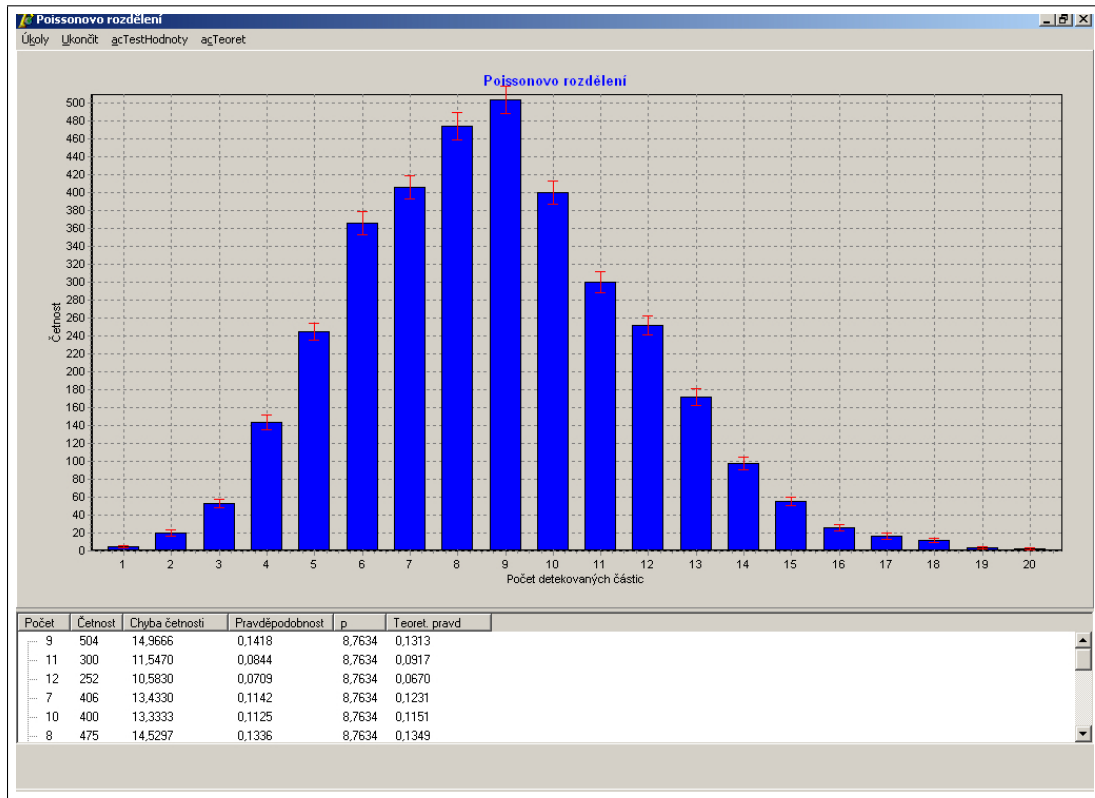
Pokud bude úloha měřena jako *demonstrační*, pak je pro úsporu času užito radioaktivního zářiče. Délka jednoho měření pak může klesat až na desetiny sekundy, v závislosti na aktivitě zářiče a jeho vzdálenosti od sondy Geigerova–Müllerova počítače.

Měření je zahájeno po výběru z hlavního menu **Úkoly-Záznam**. Měření je možné pozastavit položkou v menu **Úkoly-Pozastavit záznam**. Opětovné spuštění lze provést opět z menu **Úkoly-Záznam**. Při pozastavení měření nedojde ke ztrátě hodnot.

Vymazání naměřených hodnot a uvedení programu do stavu, jaký měl těsně po zapnutí, provede možnost **Úkoly-Reset měření**.

Po ukončení měření může obrazovka programu vypadat např. jako obrázek 32.

Modré sloupce zobrazují naměřené četnosti M_N , červené čáry znázorňují chybové úsečky ϑ_{M_N} a zelená křivka je křivka teoretického Poissonova rozdělení. Nejčastěji zachycený počet částic byl pět. Teoretická křivka protíná 7 z 16 chybových úseček, což odpovídá teoretickému předpokladu, že by křivka měla protínat přibližně polovinu úseček. Pokud je úloha prováděna jako demonstrační experiment, je vhodné ukázat, že při delším době jednoho měření přechází statistické rozdělení z Poissonova



Obrázek 33: Gaussovo rozdělení

na Gaussovo. Pro tento důkaz stačí pouze zvětšit dobu měření, nebo přiblížit zářič, a výsledek může vypadat např. jako obrázek 33.

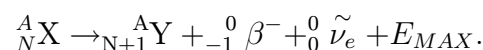
4.3 Dolet částic beta

4.3.1 Teorie úlohy *Dolet částic beta*

Teorie k této úloze byla převzata z [1].

Autor v [1] (Rauner K., Prokšová J., Havel V.: *Fyzikální praktikum II*. Západočeská univerzita, Plzeň 1999) uvádí:

„Schéma radioaktivní přeměny beta je



Po přeměně vznikne prvek posunutý v Mendělejevě tabulce o jedno místo vpravo, má ale stále stejné hmotnostní číslo. Vznikne částice beta (elektron s vysokou energií) a elektronové antineutrino. Energie E_{MAX} se rozděluje mezi elektron a antineutrino. Částice beta tedy může mít jakoukoli energii z intervalu $< 0, E_{MAX}$.

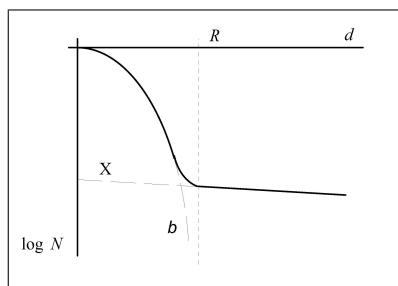
Při průchodu částice látkou (absorbátorem) jsou elektrony pohlcovány. Označme

N_0 počet beta částic dopadajících kolmo na absorbátor tloušťky d a N počet částic beta tímto absorbátorem prošlých. Grafické vyjádření závislosti $\log N$ na d nazýváme **absorpční křivka**.

Tato křivka je složitější než by tomu bylo u absorpční křivky alfa z následujících důvodů:

- částice β mají spojité spektrum v intervalu $< 0, E_{MAX} >$,
- částice β mají malou hmotnost a proto se nepohybují přímočaře,
- kromě ionizace, jež se uplatňuje i u záření α , se výrazně uplatňuje i pružný a nepružný rozptyl a brzdné záření,
- u β částic se značně uplatňují relativistické efekty, jejich hmotnost může dosahovat až pětinasobku hmotnosti klidové.

Použije-li se k měření absorpční křivky izotopu s jednoduchou přeměnou podle výše uvedeného rozpadového vzorce bude mít křivka tvar podle obrázku 34.

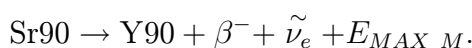


Obrázek 34: Jednoduchá absorpční křivka

Naměřená křivka neklesá k $-\infty$ v místě doletu, ale přechází v zde v přímku s malou strmostí. Absorpční křivku totiž zjišťujeme detektory ionizujícího záření, které detekují nejen částice beta, ale i sekundární projevy vysokoenergetického elektromagnetického záření, které vždy průchod beta částic hmotou provází (brzdné záření X). Dolet částic beta (tloušťka absorbátoru, kterým již neprojde žádná částice) se tedy určuje nepřesně — z místa přechodu absorpční křivky v přímku X. Skutečná absorpční křivka se získá odečtením příspěvku od záření X, jehož průběh se určí extrapolací přímky X k ose $\log N$.

Tato křivka (na obr. 34 označená β) se pak asymptoticky blíží rovnoběžce s osou $\log N$, která protíná osu d v místě R — doletu částic β v daném absorbátoru.

Protože zářiče β s vysokou energií E_{MAX} mají krátké poločasy rozpadu, používá se v laboratorních podmínkách zářiče $\text{Sr90} + \text{Y90}$, jehož schéma dvojité beta přeměny je



$$\text{Y90} \rightarrow \text{Zr90} + \beta^- + \tilde{\nu}_e + E_{MAX\ T}.$$

Poločas prvního rozpadu je 27,2 let, poločas druhého rozpadu je 64 hodin.

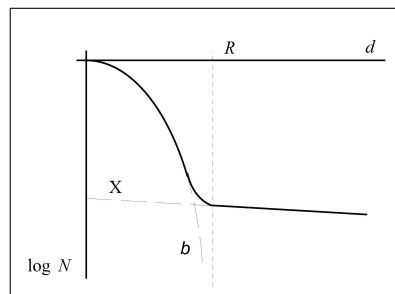
$$E_{MAX\ T} = 1,180 \pm 0,008 \text{ MeV}.$$

$$E_{MAX\ M} = 0,545 \pm 0,005 \text{ MeV}.$$

Indexy M a T rozlišují *Měkkou* a *Tvrdou* složku záření.

Izotop Zr90 je stabilní a spojuje výhodu vysoké energie emitovaných beta částic a dlouhý poločas rozpadu. Nevýhodou je složité spektrum energií β částic, které obsahují dvě složky s energiemi v intervalech $< 0, E_{MAX\ M} >$, $< 0, E_{MAX\ T} >$.

Tvar absorpční křivky je v tomto případě dán superpozicí dvou absorpčních křivek podle obr. 35

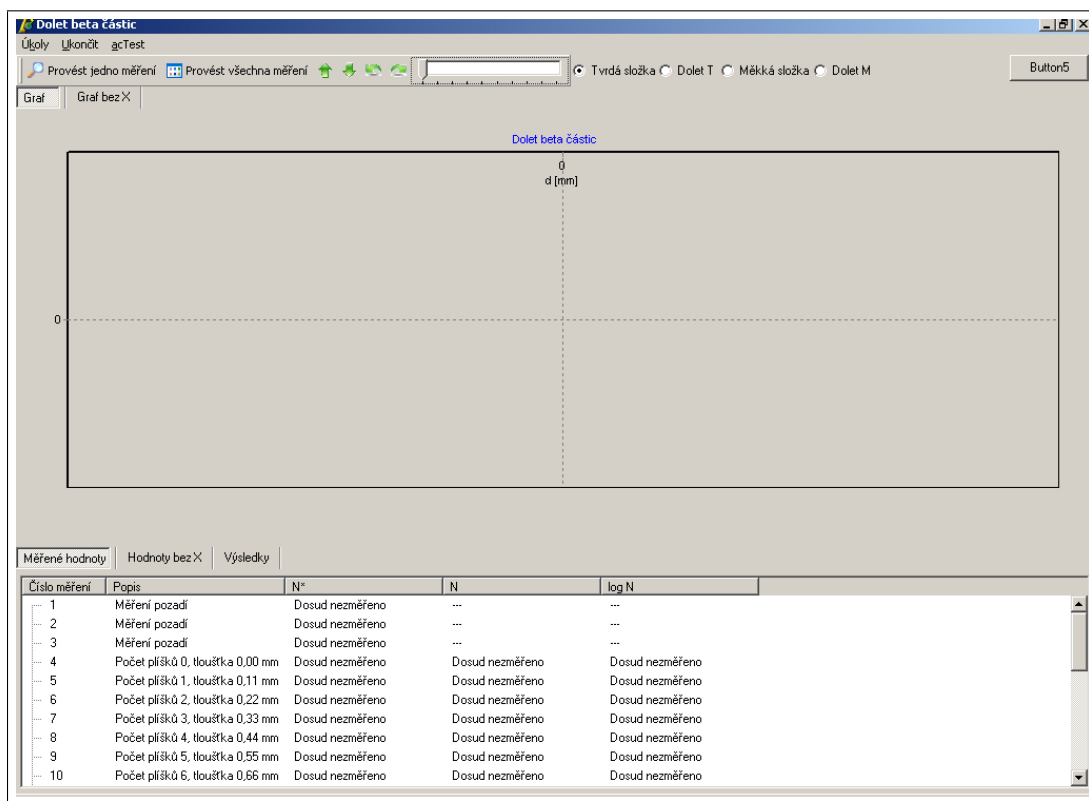


Obrázek 35: Superpozice dvou absorpčních křivek

Z tohoto obrázku je vidět, že dolet β záření s menší energií R_M se určuje ještě obtížněji než dolet záření s vyšší energií R_T , protože provést extrapolaci absorpční křivky β_T k ose $\log N$ je těžké pro neznalost analytického vyjádření této křivky.

Poznámka: Ani extrapolace absorpční přímky záření X není zcela přesná, protože při $d = 0$ žádné záření X nevzniká. Při malých tloušťkách absorbátoru tvoří však záření X pouze 0,1–1 % celkové intenzity, jeho příspěvek je zde zanedbatelný.

V praktickém měření absorpční křivky chybí část pro nejmenší tloušťky d , protože záření β se i při nulové tloušťce absorbátoru pohlcuje a rozptyluje ve vzduchu a na fólii, chránící scintilační krystal detektoru před světlem. K absorpci dochází i v zářiči samotném, a to pro β částice vycházející z určité hloubky zářiče.“

Obrázek 36: Hlavní obrazovka programu *Dolet beta částic*

4.3.2 Ovládání programu pro úlohu *Dolet beta částic*

Ovládací program pro úlohu *Dolet beta částic* je umístěn v souboru *Dolet-Beta.exe*.

Základní ovládání blíže zmiňuje kapitola 4.5.3. Na tomto místě bude blíže popisováno specifické ovládání pro program sloužící měření úlohy *Dolet beta částic*.

Hlavní obrazovka ovládacího programu je na obrázku 36.

Do horní části hlavní obrazovky bude v průběhu měření dynamicky vykreslován graf, ve spodní části, v bílé tabulce, jsou v průběhu měření zobrazovány naměřené hodnoty. Poměr prostoru, který na hlavním okně zabírá graf a tabulka, lze změnit svislým tažením horizontální dělicí čáry mezi nimi. To je užitečné při větším množství naměřených hodnot, kdy by se hodnoty v malé tabulce stávaly nepřehledné. Pokud je tabulka větší než jí přidělený prostor, lze v ní rolovat svislým posuvníkem v pravé části okna.

V horní polovině okna je umístěn ještě druhý graf, ve kterém bude vynesena absorpční křivka záření beta bez příspěvku od záření X. Tento graf lze zobrazit přepnutím tlačítka **Graf** na **Graf bez X**.

Podobně lze přepínat spodní část okna mezi **Měřené hodnoty**, **Hodnoty bez X** a **Výsledky**.

Na vodorovnou osu grafu je vynášena tloušťka plechů — hodnota d , na svislou

osu logaritmus počtu zachycených částic.

Tato hodnota je v grafu vynesena jako křížek, hodnoty jsou poté aproximovány polynomem (více viz. oddíl 4.5.3).

Sloupce ve spodní tabulce pod záložkou **Měřené hodnoty** mají následující význam:

Název sloupce	Význam
Číslo měření	O kolikáté měření se jedná
Popis	Textový popis měřené položky — počet a tloušťka plechů
N_*	Počet zachycených částic, započítáno i pozadí
N	Počet zachycených částic po odečtení pozadí
$\log N$	Dekadický logaritmus počtu zachycených částic po odečtení pozadí

Sloupce ve spodní tabulce pod záložkou **Hodnoty bez X** mají následující význam:

Název sloupce	Význam
Číslo měření	O kolikáté měření se jedná
Popis	Textový popis měřené položky — počet a tloušťka plechů
N	Počet zachycených částic po odečtení pozadí
$\log N$	Dekadický logaritmus počtu zachycených částic po odečtení pozadí
N_x	Příspěvek od záření X vypočtený extrapolací přímky X k ose $\log N$

Pod záložkou **Výsledky** jsou průběžně zobrazovány výsledky měření — Počet částic pozadí, Dolet tvrdé a Dolet měkké složky záření.

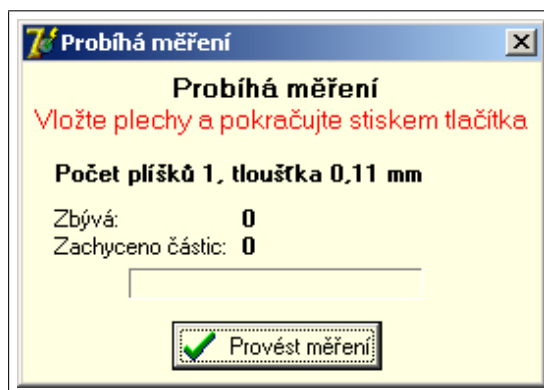
Hodnota pozadí je počítána jako aritmetický průměr mezi hodnotami prvních a posledních třech měření. Po odečtení této hodnoty od hodnoty N_* získáme hodnotu N .

Měření a zpracování probíhá tímto způsobem: Program počítá částice zachycené za určitý interval (nastavitelný uživatelem — typicky sto sekund). Vždy po uplynutí tohoto časového intervalu je získaná hodnota zobrazena v grafu v horní části obrazovky. Příslušně přepočítané hodnoty jsou vypisovány do spodní tabulky.

Při každé hodnotě přidané do grafu je přepočítán aproximační polynom a hodnoty v grafu jsou proloženy odpovídající křivkou.

Podmínkou pro změření užitečných hodnot je správné stanovení doby měření. Tato doba lze nastavit v konfiguračním dialogu, viz. 4.5.3.

Pro zahájení měření uživatel nejprve pomocí myši vybere, který řádek z tabulky chce měřit a stiskne tlačítko **Provést jedno měření**. Zobrazí se okno dle obrázku 37.

Obrázek 37: Měřicí dialog programu *Dolet beta záření*

Černý text informuje, které část je zrovna měřena, a upozorňuje obsluhu na vložení správného počtu vzorků. Vlastní měření započne stiskem tlačítka **Provést měření**. Po uplynutí měřicí nastavené doby je okno uzavřeno a získaná hodnota je vyplněna do tabulky v hlavním okně.

Pro zrychlení celého procesu je možné zvolit tlačítko **Provést všechna měření**. Měřicí dialog z obrázku 37 není po každém měření uzavřen, pouze opakuje výzvu k vkládání vzorků než proběhne měření pro všechny vzorky.

Vymazání naměřených hodnot a uvedení programu do stavu jaký měl těsně po zapnutí provede možnost v hlavním menu **Úkoly-Reset měření**.

Po ukončení měření může obrazovka programu vypadat např. jako obrázek 38.

Od grafu podle obrázku 38 je nutné odečíst příspěvek od záření X. K tomu je nutné nastavit žlutou čárkovanou přímkou do polohy absorpční přímky záření X — podle obrázku 35.

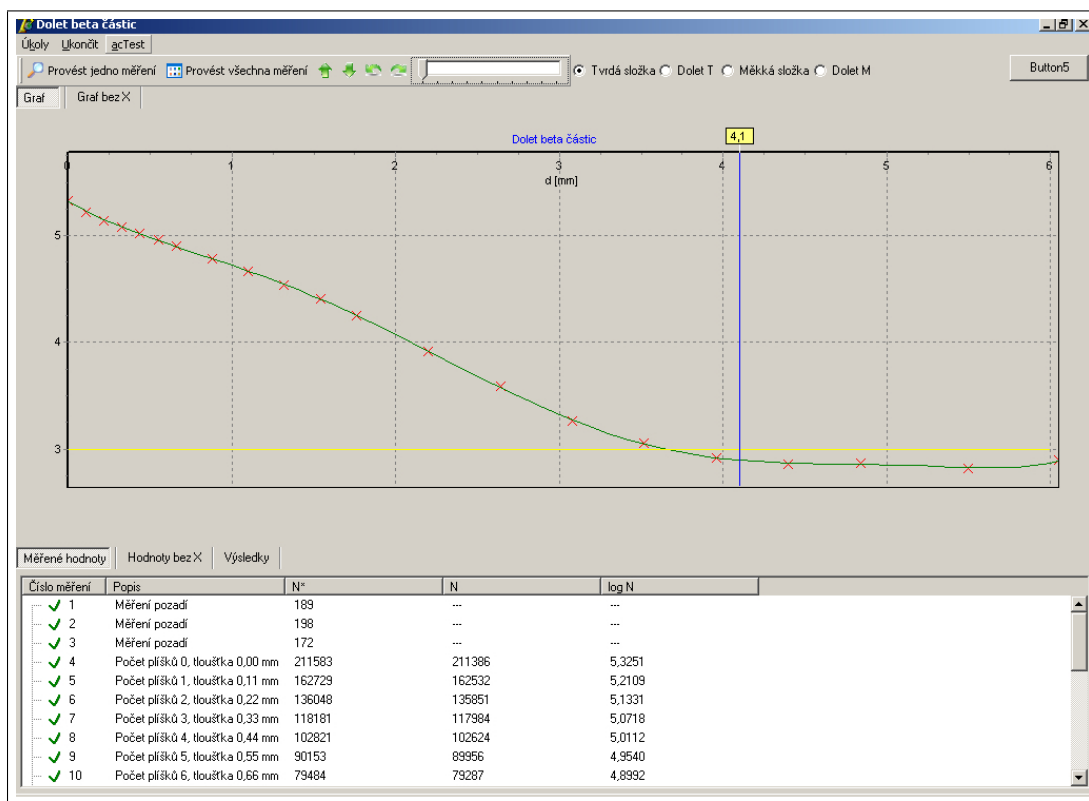
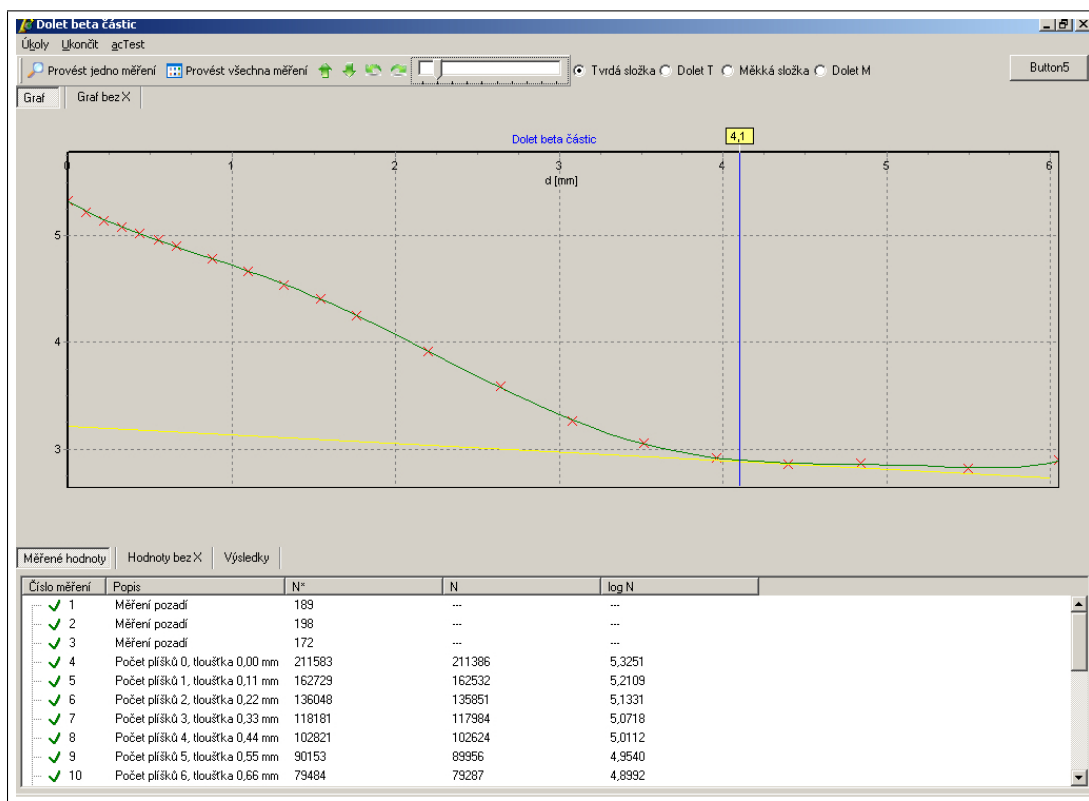
K pohybu žluté přímky jsou určena zelená tlačítka se symbolem šipek a je nutné mít zvolené přepínací tlačítko **Tvrdá složka**. Krok, po kterém se bude křivka posouvat lze upravit posuvníkem mezi přepínacími tlačítky a zelenými šípkami. Čím je posuvník více vlevo, tím citlivěji bude přímka posouvána.

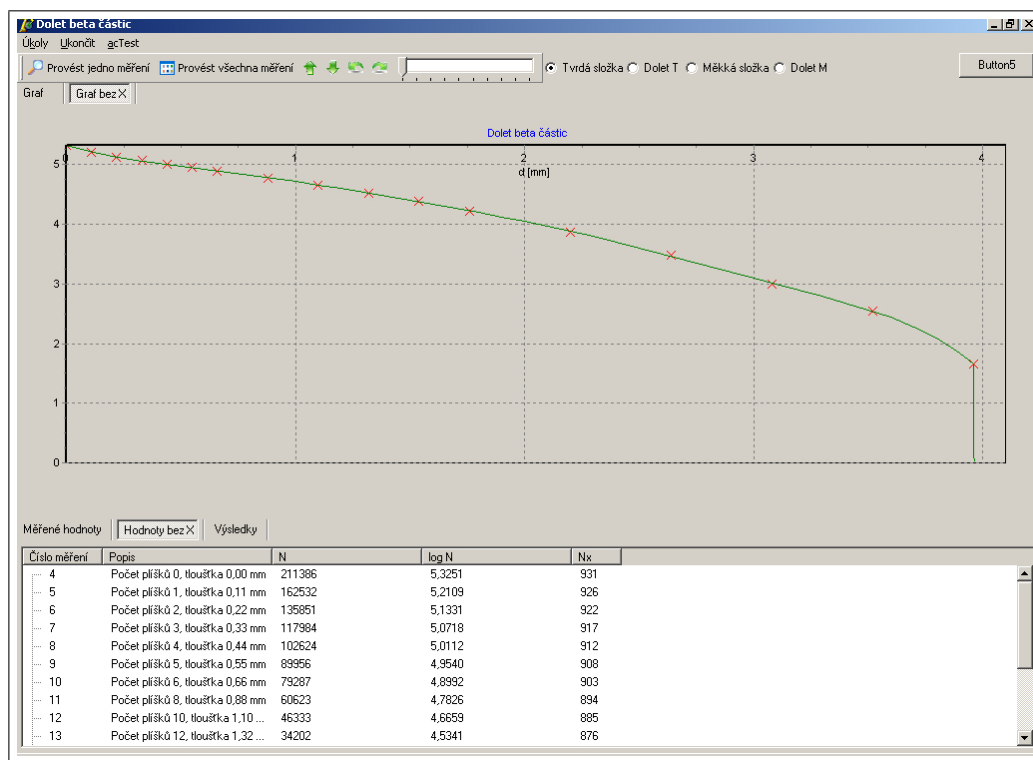
Po nastavení do požadované polohy je nutné stanovit dotek tečny s absorpční křivkou. Ten určuje svislá modrá přímka, lze jí pohybovat při zvoleném přepínacím tlačítku **Dolet T**. Poloha této svislice určuje dolet tvrdé složky záření beta.

Obrazovka s nastavenou tečnou a svislicí vypadá podle obrázku 39.

Se správně nastavenými přímkami lze od hodnot odečíst příspěvek od záření X. To proběhne po výběru položky z menu **Úkoly-Přepočíst graf bez X**. Tímto se vyplní tabulka pod záložkou **Hodnoty bez X** a graf pod záložkou **Graf bez X**. Výsledek může vypadat jako obrázek 40.

Posledním úkolem je nalezení doletu měkké části záření X. To nalezneme obdobně jako při nalézání doletu tvrdé složky. Tečnou, resp. svislicí hýbeme opět šípkami, nyní ovšem v grafu pod záložkou **Hodnoty bez X** a při zapnutém přepínacím tlačítku

Obrázek 38: Obrazovka programu *Dolet beta částic* s naměřenými hodnotamiObrázek 39: Obrazovka programu *Dolet beta částic* s naměřenými hodnotami a nastavenými čarami



Obrázek 40: Obrázovka programu *Dolet beta částic* grafem po odpočtení příspěvku záření X

Měkká složka, resp. Dolet M.

Tím jsou úkoly splněny a získané výsledky jsou dostupné pod záložkou **Výsledky**.

Hodnoty a grafy je možné exportovat výběrem z menu **Úkoly-Export dat**.

V případě exportu dat z tabulky jsou exportovány výsledky z obou tabulek ve spodní části obrazovky (záložka **Měřené hodnoty** i **Hodnoty bez X**).

Pokud uživatel vybere možnost exportu grafu, bude uložen graf, který je v tu chvíli aktivní.

4.4 Zpětný rozptyl částic beta

4.4.1 Teorie úlohy *Zpětný rozptyl částic beta*

Teorie úlohy byla převzata z [2].

Autor v [2] (Rauner K.: *Zpětný rozptyl částic beta*. Školská fyzika **VII**, mimořádné číslo (2001)) uvádí:

„Interakce částic beta při průletu hmotou jsou velmi rozmanité a složité. Je to dáno velkým specifickým nábojem částic beta a výraznými relativistickými efekty, protože kinetická energie částic bývá několik MeV, zatímco klidová energie elektronu

je 0,511 MeV. Energetické spektrum beta částic je navíc spojitě, což je dáno rozdělením energie radioaktivní přeměny mezi elektron a neutrino při přeměně neutronu v proton:

$${}^1_0\text{n} \rightarrow {}^1_1\text{p} + {}^0_{-1}\beta^- + {}^0_0\tilde{\nu}.$$

Hlavními interakcemi jsou:

- pružný rozptyl na elektronovém obalu,
- nepružný rozptyl na elektronovém obalu (ionizace, excitace),
- pružný a nepružný rozptyl na atomovém jádře,
- brzdné záření,
- Čerenkovovo záření.

Hmotnost částice beta je navíc velmi malá, proto se pohybují po složité trajektorii ve tvaru lomené čáry. Popsat kvantitativně závislost úhlu rozptylu na protonovém čísle Z rozptylujícího prvku je proto teoreticky velmi obtížné a v literatuře se s takovým vztahem nesetkáváme. V učebnicích kvantové mechaniky bývá popsán jen rozptyl monoenergetických elektronů nízkých energií. Z rozboru mechanismu rozptylu a z její hlavní příčiny — coulombovského působení elektronového obalu a jádra na elektron — je zřejmé, že za jinak stejných podmínek bude úhel rozptylu záviset na protonovém čísle rozptylujícího prvku. Čím větší bude protonové číslo vzorku, tím více se částice beta budou rozptylovat. Při rozptylu po průchodu vzorkem o tloušťce menší než je dolet částic je nutné brát v úvahu koncentraci rozptylujících atomů, hustotu a tloušťku vzorku. Vyloučení těchto vedlejších parametrů se dosáhne měřením zpětného rozptylu (rozptylu o úhly blízké 180°) na vzorcích, jejichž tloušťka přesahuje dolet částic. I tento rozptyl je tím pravděpodobnější, čím je protonové číslo rozptylujícího materiálu větší. Toho se využívá i při výrobě zářičů — plošný radioaktivní vzorek se podkládá prvkem s velkým Z (olovem) a tak se dosahuje zvýšení aktivity zářiče až o 50%.

Závislosti zpětného rozptylu částic beta se využívá k analýze. Výhodou této metody je jednak to, že je nedestruktivní, jednak to, že umožňuje zjišťovat prvky ve vzorku bez ohledu na chemické vazby. K nevýhodám metody patří hlavně to, že u směsí, sloučenin a slitin poskytuje jen údaj o středním protonovém čísle, ke kvantitativní analýze musí mít vzorek přesně definovaný tvar. Proto se tato metoda v jaderné chemii příliš často nepoužívá, výhodnější je neutronová aktivační analýza či jiné postupy. Vzhledem k fyzikální podstatě se však metoda analýzy pomocí zpětného rozptylu částic beta velmi dobře hodí k zařazení do fyzikálních praktik

na vysokých školách různého typu. Popisovaná úloha byla poprvé měřena na pedagogické fakultě ZČU v Plzni na Letní škole fyziky 2001, od zimního semestru bude vedle dalších úloh součástí fyzikálního praktika zaměřeného na jadernou fyziku a optiku. Měření zpětného rozptylu částic beta se provádí v uspořádání podle obrázku 41.

Rozptylující vzorky mají tvar nízkých válců s průměrem 34 mm a s výškou přesahující dolet částic. Aby byla zachována konstantní vzdálenost od zářiče, podkládají se vzorky vyrovnávacími podložkami. K dispozici jsou vzorky prvků: C, Al, S, Fe, Ni, Cu, Zn, Sn, Hg, Pb, Bi a slitin Ag+Cu, Pb+Sn. Při vlastním měření se zjišťuje počet rozptýlených částic na aparatuře bez vzorku (vzduch, podložka pod vzorkem z akrylového skla), na vybraných vzorcích prvků (Hg většinou studenti nepoužívají) a na zvolené slitině. Doba jednoho měření je 10 sekund, použitý zářič (kontrolní typ J) má aktivitu 100 kBq. K snížení chyby vznikající z časově proměnné funkce aparatury vlivem zahřívání se všechny vzorky měří dvakrát v opačném pořadí. Měřené údaje se zpracují tabelárně a graficky např. v Excelu, zjišťuje se polynomičtá aproximace zjištěné závislosti $N(Z)$ (N je měřený počet zpětně se rozptylujících β -částic, Z protonové číslo vzorku) i inverzní závislosti. Příklady zpracování jsou na obrázku 42.

Z polynomu popisujícího inverzní závislost je pak možné zjistit střední protonové číslo slitiny a určit tak zastoupení obou prvků ve slitině.

U slitiny Ag+Cu bylo tak zjištěno zastoupení stříbra 72,4 %, oficiální údaj je 70 %. Závislost zpětného rozptylu β částic je možné předvádět i demonstračně. Lze k tomu použít Geigerovu–Müllerovu trubici ze soupravy GAMABETA, na kterou se připevní zářič s vyzařovacím úhlem tak, aby částice unikaly směrem od trubice. Přiblížíme-li takto upravený detektor k materiálům s různým protonovým číslem (například Al a Pb), je akusticky indikována výrazná odlišnost zpětného rozptylu. Takto upravený detektor může při demonstračním pokusu dokonce sloužit jako hledač kovů: minci (10 Kč, 20 Kč) lze zahrabat do písku ve fotomisce a detektorem ji bezpečně nalézt.“

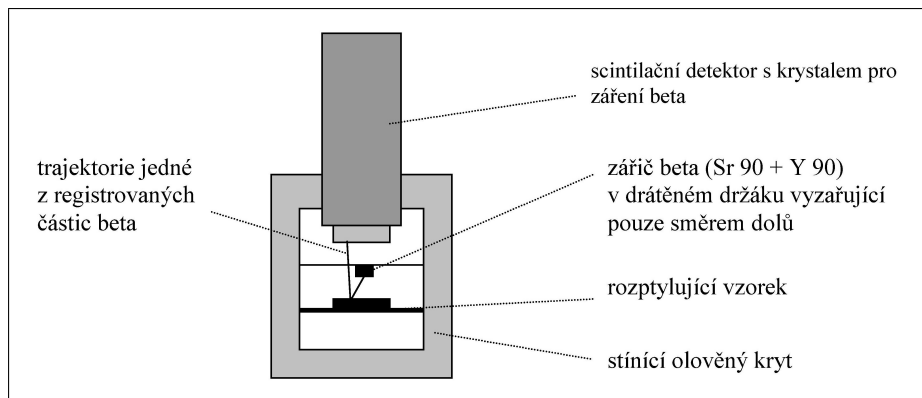
4.4.2 Ovládání programu pro úlohu *Zpětný rozptyl beta záření*

Ovládací program pro úlohu *Zpětný rozptyl beta záření* je umístěn v souboru *CB.exe*.

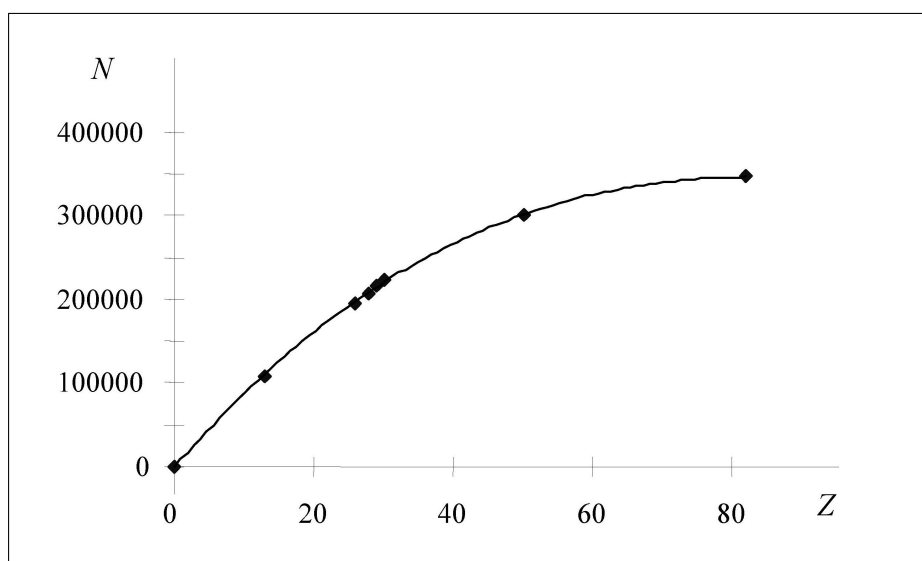
Základní ovládání blíže zmiňuje kapitola 4.5.3. Na tomto místě bude blíže popisováno specifické ovládání pro program sloužící měření úlohy *Zpětný rozptyl beta záření*.

Hlavní obrazovka ovládacího programu je na obrázku 45.

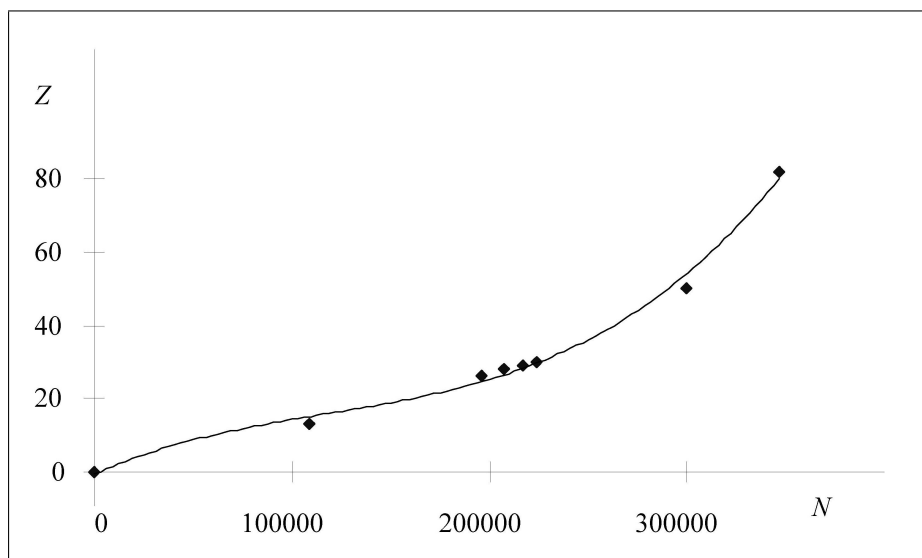
Do horní části hlavní obrazovky bude v průběhu měření dynamicky vykreslován graf, ve spodní části, v bílé tabulce, jsou v průběhu měření zobrazovány hodnoty.



Obrázek 41: Měřicí aparatura úlohy *Zpětný odraz beta částic*



Obrázek 42: Závislost N na Z v úloze *Zpětný rozptyl beta částic*



Obrázek 43: Závislost Z na N v úloze *Zpětný rozptyl beta částic*

Poměr prostoru, který na hlavním okně zabírá graf a tabulka, lze změnit svislým tažením horizontální dělicí čáry mezi nimi. To je užitečné při větším množství naměřených hodnot, kdy by se hodnoty v malé tabulce stávaly nepřehledné. Pokud je tabulka větší než jí přidělený prostor, lze v ní rolovat svislým posuvníkem v pravé části okna.

V horní polovině okna je umístěn ještě druhý graf, ve kterém bude vynesena závislost počtu zachycených částic N na protonovém čísle prvku Z . Graf závislosti Z na N lze zobrazit přepnutím tlačítka **Graf N-Z** na **Graf Z-N**.

Hodnota zachyceného počtu částic je v grafu vynesena jako křížek, hodnoty jsou poté aproximovány polynomem (více viz. oddíl 4.5.3).

Sloupce ve spodní tabulce pod záložkou **Měřené hodnoty** mají následující význam:

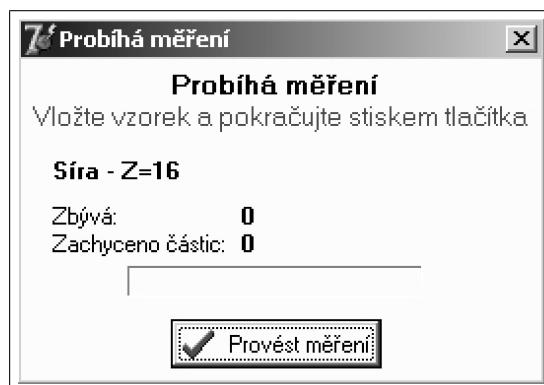
Název sloupce	Význam
Číslo měření	O kolikáté měření se jedná
Popis	Textový popis měřené položky — prvek a protonové číslo
N	Počet zachycených částic

Měření a zpracování probíhá tímto způsobem: Program počítá částice zachycené za určitý interval (nastavitelný uživatelem — typicky sto sekund). Vždy po uplynutí tohoto časového intervalu je získaná hodnota zobrazena v grafu v horní části obrazovky. Příslušně přepočítané hodnoty jsou vypisovány do spodní tabulky.

Při každé do grafu přidané hodnotě je přepočítán aproximační polynom a hodnoty v grafu jsou proloženy odpovídající křivkou.

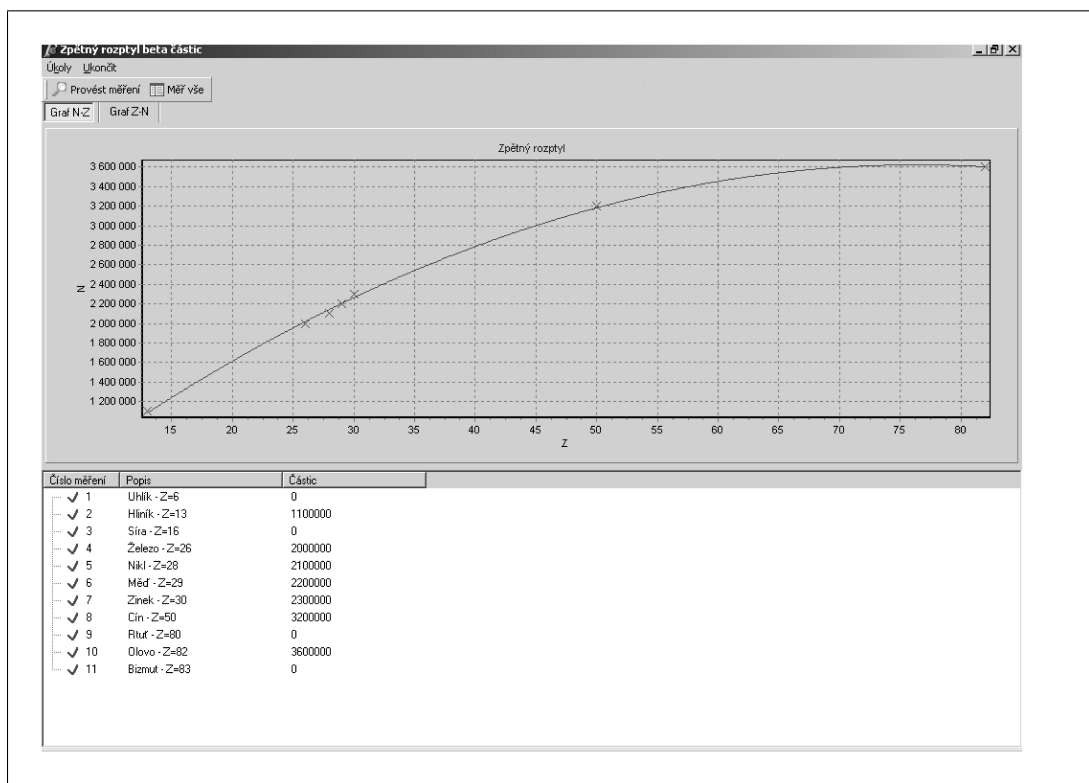
Podmínkou pro změření užitečných hodnot je správné stanovení doby měření. Tato doba lze nastavit v konfiguračním dialogu, viz. 4.5.3

Pro zahájení měření uživatel nejprve pomocí myši vybere který řádek z tabulky chce měřit a stiskne tlačítko **Provést jedno měření**. Zobrazí se okno dle obrázku 37.



Obrázek 44: Měřicí dialog programu *Zpětný rozptyl beta záření*

Vlastní měření započne stiskem tlačítka **Provést měření**. Po uplynutí měřicí



Obrázek 45: Obrazovka programu *Zpětný rozptyl beta částic* s naměřenými hodnotami

nastavené doby je okno uzavřeno a získaná hodnota je vyplněna do tabulky v hlavním okně a vynesena do grafu.

Pro zrychlení celého procesu je možné zvolit tlačítko **Provést všechna měření**. Měřicí dialog z obrázku 44 není po každém měření uzavřen, pouze opakuje výzvu k vkládání plechů než proběhne měření pro všechny plechy.

Vymazání naměřených hodnot a uvedení programu do stavu jaký měl těsně po zapnutí provede možnost v hlavním menu **Úkoly-Reset měření**.

Po ukončení měření může obrazovka programu vypadat např. jako obrázek 45.

Hodnoty a grafy je možné exportovat výběrem z menu **Úkoly-Export dat**.

Pokud uživatel vybere možnost exportu grafu, bude uložen graf, který je v tu chvíli aktivní.

4.5 Technické a programové řešení práce

Úkolem práce bylo zpracování výsledků měření úloh fyzikálního praktika počítačem. Ve všech úlohách je tímto počítačem standardní PC s procesorem Intel a s nainstalovaným systémem Windows. Tato platforma byla zvolena pro svou nízkou cenu a snadnou dostupnost. U většiny uživatelů lze předpokládat alespoň základní zkušenost právě s osobními počítači PC a systémem Windows.

Programy vzniklé v rámci této diplomové práce důsledně dodržují zásady uživa-

telského rozhraní Windows (např. využívání standardních ovládacích prvků — tlačítka, roletové nabídky, ...). Program může úspěšně obsluhovat každý, kdo zvládl základní pracovní postupy v systému Windows.

4.5.1 Technické řešení práce

V úlohách 1 (*Poissonovo rozdělení*), 2 (*Dolet beta částic*) a 3 (*Zpětný rozptyl*) bylo nutné vyřešit komunikaci fyzikálních přístrojů, na nichž bylo měření dosud prováděno, s osobním počítačem. Do počítače dostačuje posílat pouze údaj *zachycena/nezachycena* částice (čistě binární informace), není nutné zaznamenávat velikost signálu. Při vzniku této práce bylo uvažováno několik možných přístupů k tomuto problému. Řešení mělo být provedeno jedním z těchto postupů:

- Připojení výstupu fyzikálního měřicího přístroje přímo na sériový port osobního počítače (pro vstup signál RxD, pro výstup signál TxD.).
- Připojení výstupu přístroje do zvukové karty v osobním počítači. Dále viz oddíl 4.5.1.
- Komunikace s osobním počítačem přes sériový port, oproti řešení 1 je mezi měřicí přístroj a PC vložena „mezivrstva“ — elektronický obvod zajišťující upravení signálu z měřicího přístroje.

Komunikace přes sériový port V tomto scénáři komunikace měřicího přístroje a osobního počítače je výstup z přístroje připojen přímo do sériového portu PC. Obslužný program provádí čtení dat ze sériového portu, jedné přístrojem zachycené částici odpovídá jedna změna logické úrovně na sériovém portu. Získaný signál není nutné nijak vzorkovat.

Na napěťové úrovni sériové komunikace je kladena tato podmínka:

- logická 1 odpovídá napěťové úrovni $-3\text{ V} - -15\text{ V}$
- logická 0 odpovídá napěťové úrovni $+3\text{ V} - +15\text{ V}$

Výhody řešení:

- sériový port je běžnou součástí osobních počítačů,
- snadno realizovatelné,
- nejsou potřeba další pomůcky,
- vysoká rychlost snímání signálu z měřicího přístroje.

Nevýhody řešení:

- Málo „robustní“ řešení — na přístroj je kladena podmínka výstupního napětí v intervalu vhodném pro sériový port. Při možné budoucí záměně měřicího přístroje není zaručena funkčnost obslužného software.
- Potencionální nebezpečí v poškození sériového portu, případně celého počítače při poruše měřicího přístroje.
- Chybí možnost předběžné úpravy signálu před zpracováním v počítači.

Ve konečné verzi nebylo toto řešení vzhledem k vyjmenovaným nevýhodám použito.

Komunikace přes vstup zvukové karty V tomto přístupu ke komunikaci měřicího přístroje s osobním počítačem je výstup z přístroje připojen do vstupu standardní zvukové karty (LINE-IN).

Obslužný program provádí čtení dat ze vstupu zvukové karty. Získaný signál je nutné analyzovat a tím určit počet zachycených částic.

Výhody řešení:

- Vysoká rychlost snímání signálu z měřicího přístroje.

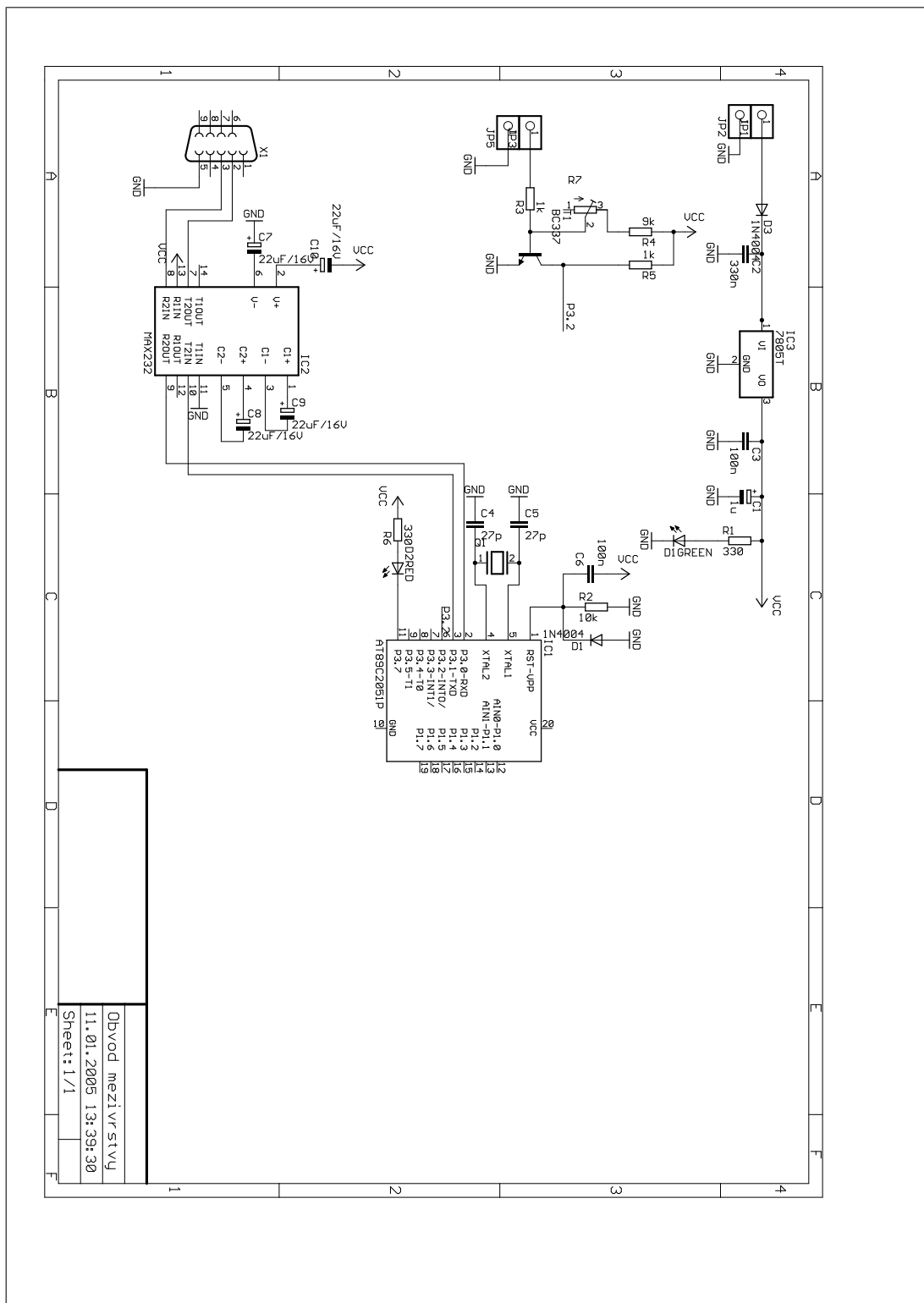
Nevýhody řešení:

- Nutnost analyzování (vzorkování) přijatých dat. Vzorkování dat může být časově náročné — po tuto dobu není možné zpracovávat další vstup z měřicího přístroje.
- Potencionální nebezpečí v poškození zvukové karty, případně celého počítače při poruše měřicího přístroje.
- Závislost na konfiguraci software v osobním počítači (např. nastavení ovladačů hlasitosti zvukové karty).
- Nutnost použití počítače se zvukovou kartou (u PC v laboratoři nemusí být zvuková karta standardní součástí).

Přístup nebyl použit pro svou závislost na dodatečném hardware počítače (zvuková karta) a nutnost nesnadné úpravy přečtených dat.

Komunikace přes sériový port s použitím „mezivrstvy“ Důvodem vzniku tohoto přístupu byla nutnost odstranění nevýhod 1. řešení (viz. oddíl 4.5.1).

Mezi fyzikální přístroj a PC je vložen další elektronický obvod („mezivrstva“). Úkolem tohoto obvodu je přizpůsobovat signál z měřicího přístroje na signál vhodný pro zpracování v PC, v tomto případě pro sériový port.



Obrázek 46: Schéma obvodu mezivrstvy

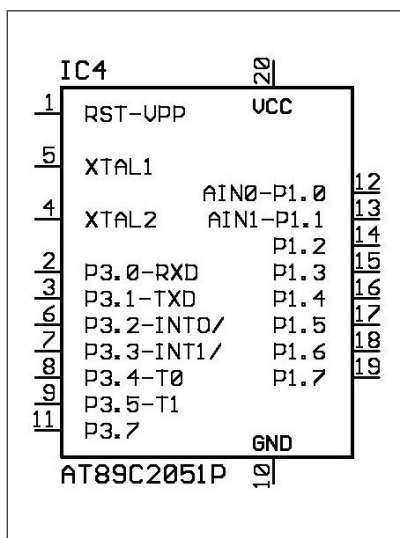
Schéma mezivrstvy je na obrázku 46. Celá mezivrstva je založena na mikrokontroléru ATMEL AT89C2051. Obvod je detailně popsán v [3], zde bude uvedena pouze základní charakteristika mikrokontroléru.

Základní charakteristika mikrokontroléru AT89C2051 Mikrokontrolér svou konstrukcí vychází z jednočipového mikropočítače 8051. Stejně jako μ procesor 8051 má AT89C2051 Hardwariskou architekturu — paměť programu a dat je oddělena. Na údaje v datové paměti nelze nahlížet jako na instrukce. Tato architektura významným způsobem zvyšuje bezpečnost běhu programu (chyba programátora nemůže přepsat údaje v datové paměti). Dále vede ke zvýšení výkonnosti mikrokontroléru.

- vestavěná Flash paměť pro uložení programu 2kB. Výrobce zaručený počet přeprogramování je 1000,
- datová paměť RAM o kapacitě 128 B,
- napájecí napětí 2,7-6 V,
- hodinový kmitočet 0-24 MHz,
- 15 vstupně/výstupních linek,
- vestavěný analogový komparátor,
- programovatelný sériový kanál využitelný např. pro komunikaci s PC,
- 6 zdrojů přerušení,
- dva režimy snížené spotřeby (IDLE mode a Power-down mode).

Sériová komunikace Uvnitř mikrokontroléru je vestavěn plně duplexní sériový kanál (vysílání i příjem může probíhat současně). Vstup je opatřen vyrovnávacím registrem, který umožňuje odebírat data se zpožděním až jednoho znaku. Mikrokontrolér není vybaven příznaky pro detekci chyby (chyba parity apod.). V této práci je mezi PC a obvodem AT89C2051 použito komunikace s těmito parametry:

- Rychlost 14400 B/s,
- osmibitová komunikace,
- nepoužívá se parita,
- použit jeden stop bit.



Obrázek 47: Vývody mikrokontroléru AT89C2051 na pouzdru DIP 20

Popis zapojení obvodu mezivrstvy Obvod mezivrstvy je zapojen podle obrázku 46. Mikrokontrolér (označen jako AT89C2051) pro svou činnost vyžaduje externí hodinový signál, který synchronizuje činnost celého mikrokontroléru. Ten je generován krystalem Q_1 , k němuž jsou dále připojeny kondenzátory C_4 a C_5 . Zdroj externího hodinového signálu je přiveden na vstupy XTAL1 a XTAL2. Vhodným výběrem krystalu lze ovlivňovat hodinovou frekvenci mikrokontroléru. V použitém zapojení pracuje mikrokontrolér s frekvencí 24 MHz. Délka jednoho strojového cyklu (zpracování jedné jednoduché instrukce) je 12x větší než hodinový cyklus. Pro frekvenci 24 MHz tedy trvá zpracování jednoduché instrukce $0,5\mu s$.

Mikrokontrolér je napájen napětím 5 V. To je získáno stabilizátorem napětí — integrovaným obvodem 7805. Na jeho vstup je přivedeno napájecí napětí 9 V ze síťového adaptéru. Na výstupu stabilizátoru je napětí 5 V. To je přivedeno na napájecí vstup VCC mikrokontroléru. Je jím napájena také zelená LED D_1 GREEN; pokud svítí, je napájení celého obvodu v pořádku.

Mikrokontrolér AT89C2051 je vybaven dvěma vstupy pro vnější přerušení (vstupy P3.2 — INT0 a P3.3 — INT1). Na existenci tohoto vstupu je založena funkce celé mezivrstvy. Výstup měřicího přístroje je připojen ke vstupu P3.2. Přivedením sestupné hrany na tento vstup je mikrokontrolérem generováno přerušení. Při každém detekovaném přerušení je programově inkrementován čítač.

Hodnota tohoto čítače je odesílána přes sériový port do osobního počítače. Mikroprocesor může komunikovat s osobním počítačem ve dvou režimech, blíže kapitola 4.5.2. Pokud je počítač vytížen zpracováním požadavku jiného programu a nemůže obsluhovat sériový port, nejsou impulsy z měřicího přístroje ignorovány, ale jsou detekovány obvodem ATMEL. Po uvolnění výpočetního výkonu PC je obvodu ATMEL odeslán požadavek a mikrokontrolér odešle do PC počet detekovaných impulsů. Po

každém obslouženém požadavku je programový čítač vynulován.

Obvod ATMEL komunikuje s osobním počítačem přes sériový port. Pro tento úkol je mikrokontrolér vybaven vstupy P3.0 — RxD a P3.1 — TxD. Signál vstupů P3.0 a P3.1 nelze přivést přímo na sériový port osobního počítače, jeho napěťovou úroveň je nutné upravit obvodem MAX323. Pro účely této práce není nutné funkci obvodu MAX323 podrobně rozebírat. Vstup obvodu MAX323 R1IN a výstup T2OUT jsou připojeny k sériovému portu osobního počítače.

Podle [3] je pro detekci přerušení obvodem ATMEL nutné přivést na vstup P3.2 (INT0) signál se sestupnou hranou. Pro splnění tohoto požadavku je mezi výstup měřicího přístroje a vstup obvodu ATMEL zařazen tranzistor NPN, v zapojení se společným emitorem — jako spínač. Při nulovém výstupu z přístroje je na bázi tranzistoru napětí cca 0,6 V. Tranzistor není saturován a na vstupu INT0 mikrokontroléru je napájecí napětí 5 V. Hodnotu napětí lze korigovat trimrem. Jednoduše lze tedy upravit citlivost celého zapojení, případně přizpůsobit obvod pro připojení různých přístrojů. Napětí z výstupu přístroje zvýší napětí na bázi tranzistoru nad hodnotu 0,7 V. Tranzistor se otevře a uzemní vstup P3.2 obvodu ATMEL; je vyvoláno přerušení.

Výhody řešení:

- Vlastní sběr dat je proveden obvodem mezivrstvy, počítač data pouze zobrazuje.
- Vysoká rychlost snímání signálu z měřicího přístroje.
- Snadno rozšiřitelné řešení pro různé typy měřicích přístrojů.
- Počítač pracuje pouze s binární informací *zachycena/nezachycena* částice.

Nevýhody řešení:

- Obtížnější realizace.

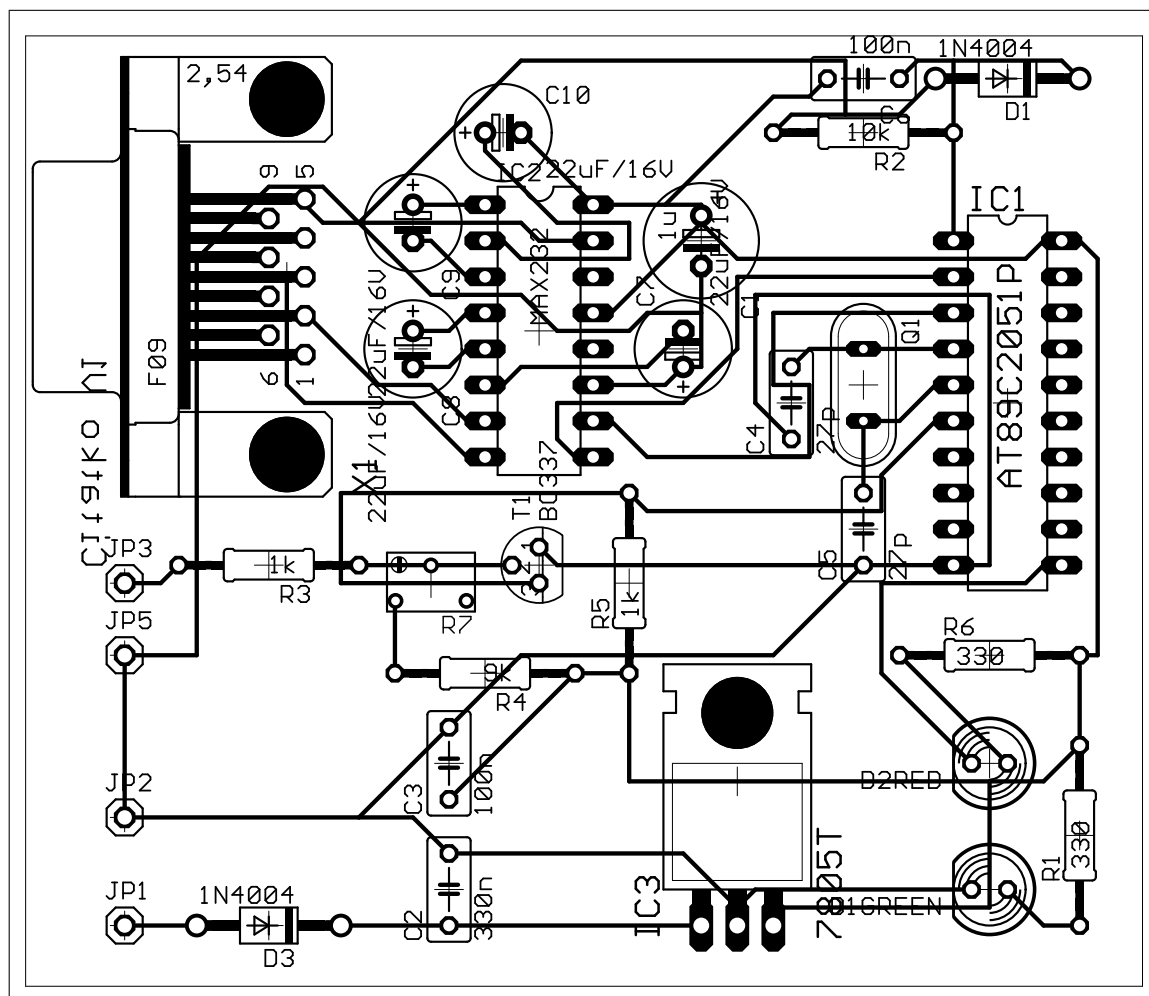
V průběhu testování vykazovalo toto řešení nejvyšší spolehlivost ze všech jmenovaných a bylo použito pro finální verzi práce.

Zapojení celého obvodu mezivrstvy bylo provedeno na plošném spoji viz obr. 48. Po osazení a otestování byl obvod umístěn do plastového pouzdra.

4.5.2 Programové řešení práce

Programové řešení je složeno ze dvou částí:

- tvorba programu pro mikrokontrolér ATMEL AT89C2051,
- tvorba front-end — grafické obslužné aplikace.



Obrázek 48: Plošný obvod mezivrstvy

Program pro mikrokontrolér ATMEL AT89C2051

Program byl vytvo-

řen v programovacím jazyce *assembler*.

Pro každé přerušení je v paměti mikroprocesoru pevně vyhražena adresa, od které začne mikroprocesor zpracovávat instrukce po příchodu přerušení (např. pro námi využívané přerušení INT0, je to adresa 03h).

Data z obvodu mezivrstvy lze získat dvěma způsoby:

- Dotazem počítače,
- Data předává mikroprocesor ATMEL za použití vnitřního časovače.

V prvním způsobu komunikace počítač v periodicky se opakujících intervalech žádá mikroprocesor o předání dat.

Mimo dobu zpracování obslužné rutiny přerušení program čeká v nekonečné smyčce. Přerušení je vyvoláno příchodem sestupné hrany na vstup P3.2 — INT0. Po příchodu přerušení začne mikrokontrolér zpracovávat rutinu obsluhy přerušení umístěné na adrese 03h. V této rutině je provedeno inkrementování čítače impulsů.

Jelikož MS-WINDOWS nejsou jednoúlohovým operačním systémem, je obtížné zaručit přesně se opakující dotazy. Tímto způsobem nelze pokládat více než cca 100 dotazů za sekundu. Protože zachycené částice jsou uvnitř mikroprocesoru ATMEL ukládány do proměnné typu byte, jejíž maximální hodnota je 255, je možné tímto způsobem snímat s obvodem mezivrstvy signály s frekvencí cca do 25 KHz. Pro vyšší frekvence je nutné použít druhý typ komunikace.

V druhém způsobu komunikace je časování odesílání počtu částic řízeno mikroprocesorem ATMEL, nikoli osobním počítačem. Nastavení frekvence mikroprocesoru se provádí zasláním jednoho bajtu s hodnotou 5 do mikroprocesoru. Další dva přijaté bajty použije mikroprocesor k nastavení vnitřního časovače. Při každém doběhlém časovači je na sériový port odeslána hodnota odpovídající aktuálnímu počtu zachycených částic.

Vyvoláním přerušení je indikováno také čtení (resp. zápis) na sériový port.

Po vyvolání tohoto přerušení je zpracována obslužná rutina na adrese 023h. Zde je nutné detekovat, zda bylo přerušení vyvoláno jako reakce na zápis nebo čtení na sériovém portu. V případě že mikrokontrolér přijal data, je nutné porovnat získaný bajt dat dle níže uvedené tabulky a provést očekávanou reakci (odeslání dat, reset, ...).

Popis komunikačního protokolu

Pro správnou funkčnost komunikace přes sériový port bylo nutné specifikovat *komunikační protokol*. Ten specifikuje formát požadavku, jež je nutný zaslat mikrokontroléru pro provedení určité činnosti. Funkci příkazu plní jeden byte dat s významem dle následující tabulky.

Hodnota příkazu	Význam	Reakce mikrokontroléru
01h	Žádost o test	V případě správné funkčnosti odešle mikrokontrolér ihned jeden byte s hodnotou 02h.
03h	Žádost o data	Mikroprocesor odešle do PC aktuální hodnotu čítače a vynuluje jej.
05h	Nastavení čítače	Oznamuje mikroprocesoru, že následující dva přijaté byte mají být použity k nastavení čítače pro obvodem ATMEL řízenou komunikaci.
05h	Žádost Reset	Dojde k vynulování vnitřního programového čítače.
06h	Žádost o Reset čítače	Dojde k vynulování čítače používaného pro obvodem ATMEL řízenou komunikaci.

Obslužný program Grafická aplikace umožňující zpracování naměřených výsledků byla vytvořena v programovacím jazyce (resp. vývojovém prostředí) Delphi. Hlavním znakem tohoto prostředí je vlastnost RAD (*Rapid Application Development*). Tvorba programů je velice rychlá a intuitivní.

Dosud bylo zpracování výsledků prováděno programy určenými pro operační systém MS-DOS. Výhody a nevýhody přechodu řešení na platformu Windows diskutuje kapitola 4.1

Komunikaci se sériovým portem obstarává knihovna Synaser, k dispozici volně na <http://www.ararat.cz/synapse/>. Knihovna zapouzdřuje přístup k sériovému portu zveřejněním třídy TBlockSerial.

Následující část kódu nastaví parametry komunikace se sériovým portem na

rychlost 14400 Bd

datové slovo osm bitů

parita bez parity

stopbity 1

SoftFlow vypnuto

HardFlow vypnuto

a odešle na port COM1 1 byte s hodnotou 8.

```
var
  bs: TBlockSerial;
begin
  bs := TBlockSerial.Create;
  bs.Connect('COM1');
  bs.Config(14400, 8, 'N', 0, False, False);
  bs.SendByte(8);
  bs.Free;
end;
```

Pro reálné použití je samozřejmě nutné doplnit předchozí část kódu o ošetření chybových stavů komunikace (např. chyba při konfiguraci sériového portu). Pro přečtení jednoho byte dat ze sériového portu by bylo použito volání funkce `bs.RecvByte`.

O tabulkové zobrazení dat je v programech použita komponenta `TVirtualStringTree` z knihovny `VirtualTrees` volně dostupné na adrese <http://www.delphi-gems.com/>.

Kromě knihovny `Synaser` a `Virtualtrees` byly v práci použity pouze standardní komponenty obsažené ve vývojovém prostředí Delphi.

Vytvořený komunikační program pro svůj běh vyžaduje osobní počítač PC s nainstalovaným systémem Win98 a vyšší.

Požadavky na počítač použitelný pro měření všech popsanych úloh jsou:

- osobní počítač PC,
- procesor 80486 a vyšší,
- schopnost komunikace přes sériový port,
- nainstalovaný operační systém Microsoft Windows 98 a vyšší,
- cca 1,5MB na pevném disku na každou úlohu.

4.5.3 Základy práce s obslužným software

Program přímo komunikující s uživatelem se pro různé typy úloh (*Poissonovo rozdělení*, *Dolet beta částic*, ...) liší. V různých verzích (ve verzích určených pro měření různého typu úloh) tohoto software lze ovšem nalézt prvky společné všem těmto verzím. Právě popisu identických částí software bude věnována tato část práce.

Data získaná z obvodu mezivrstvy je nutné jistým způsobem interpretovat a upravit pro další zpracování.

S použitím popsaných zásad bylo při tvorbě této práce rozhodnuto, že úkolem vytvořeného software nebude *vytvořit* referát z měření, ale pouze *pomoci jej vytvořit*.

Hodnoty získané v programu bylo nutné nějakým způsobem exportovat z programu, umožnit další práci s hodnotami. Problémům exportu se blíže věnuje následující kapitola.

Export dat z programu Program umožňuje exportovat naměřené výsledky do externího souboru. Pro export dat z programu bylo nutné určit nejvýhodnější formát souboru. Vzhledem k tomu, že se vždy jedná o export dat uspořádaných do tabulky, bylo potřeba nalézt formát, se kterým je schopen pracovat nejrozšířenější tabulkový editor — Microsoft Excel. Byly uvažovány tyto možnosti:

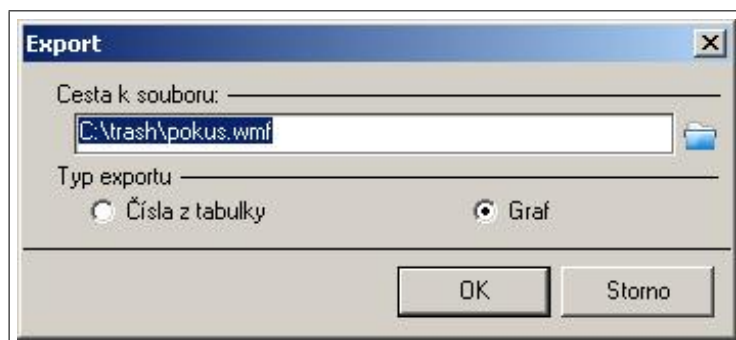
1. Přímá komunikace s programem Excel
2. Export do formátu XLS
3. Export do obecného formátu určeného pro reprezentaci tabulkově uspořádaných dat

První možnost byla téměř okamžitě zavrhnuta z důvodu nutnosti instalace programu MS Excel na počítač, na kterém bude prováděno měření, ačkoli by tato možnost nabízela možnost plně automatického exportu, tedy vysoký uživatelský komfort.

Výše popsanou nevýhodu by odstranila druhá možnost, tedy export do formátu XLS — do nativního (primárního) formátu aplikace MS Excel. Nevýhodou tohoto řešení je nesnadná práce s tímto formátem z prostředí Delphi. Tato možnost tedy také nebyla shledána vyhovující.

Nedostatky obou předchozích řešení uspokojivě odstranil třetí přístup k problému, export dat do obecného formátu určeného pro reprezentaci tabulkově uspořádaných dat. Tento formát je CSV (Comma separated values). Spojuje výhodu podpory tabulkového kalkulátoru Excel se snadným generováním tohoto formátu z prostředí Delphi. Výhodou je též podpora formátu ze strany ostatních výrobců tabulkových procesorů (např. tabulkový program kancelářského balíku Open Office). Formát lze úspěšně komprimovat (zmenšit velikost použitím jistých matematických algoritmů), což usnadňuje přenos naměřených hodnot na jiný počítač pro další zpracování. Student si např. může snadno odnést naměřené hodnoty domů a použít je pro tvorbu výsledného protokolu o měření.

Velmi užitečnou funkcí programu je též možnost exportu v programu vytvořeného grafu. Export grafu probíhá do formátu „Enhanced Meta File — přípona *wmf*“. Výhodou tohoto formátu je naprostá nezávislost na použitém zobrazovacím zařízení. Soubor vlastně obsahuje pokyny pro zobrazovací zařízení jak vykreslit obraz v něm obsažený. Tím, že formát *wmf* není bitmapový, nedochází při změně



Obrázek 49: Dialog pro export dat

proporcí k deformaci obrázku. Formát je podporován širokým spektrem aplikací. Se soubory formátu *wmf* umí pracovat i aplikace Microsoft Word, exportovaný graf lze snadno vložit do dokumentu a tak použít při tvorbě výsledného referátu o měření.

Export dat (číselných hodnot nebo grafu) je přístupný v dialogu pro export dat viz. obr. 49.

Ve spodní části okna dialogu lze přepnutím zaškrtnutých polí vybrat typ exportu — Čísla z tabulky nebo graf. Výběr souboru, do kterého budou data exportována, je proveden po stisknutí tlačítka se symbolem modré složky. Po výběru typu exportu a cesty k souboru lze zahájit export stiskem tlačítka OK. Tlačítko Storno zavře dialog bez provedení exportu.

Při dalším použití dialogu pro export dat se do okna „Cesta k souboru“ přednastaví naposled použitý soubor. To urychlí příští práci s dialogem.

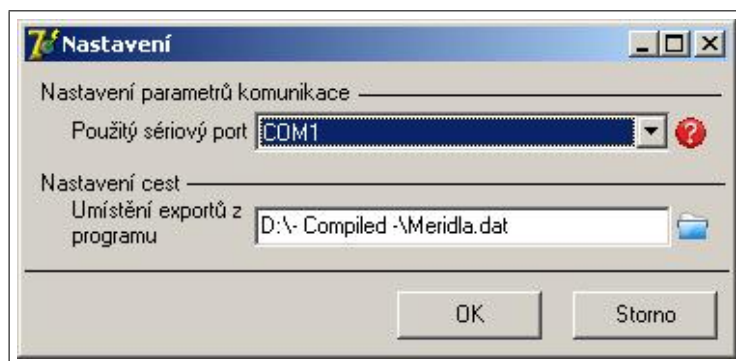
Shrnutí: Při tvorbě výsledného referátu z měření lze výhodně použít možnosti exportu dat z obslužného software. Předpokládejme zpracování referátu v programech kancelářského balíku MS Office. Naměřené hodnoty po exportu do formátu *csv* lze otevřít v programu Excel a poté převést do programu Word.

Exportované grafy lze vložit do programu MS Word přímě. Obslužný software umožňuje spolupráci s nejčastěji používanými kancelářskými programy, čímž velice usnadňuje tvorbu výsledné reprezentace dat.

Konfigurace programu Pro správnou funkčnost software je nutné správně nastavit parametry konfigurace pro komunikaci s obvodem mezivrstvy.

K těmto účelům je ve všech verzích programu vytvořen konfigurační dialog (viz obr. 50) určený k nastavení těchto parametrů.

Obslužný program musí znát číslo sériového portu, ke kterému je obvod mezivrstvy připojen. Tento parametr má tvar COMXXX, kde XXX je celé kladné číslo označující sériový port. Tento údaj lze vyčíst na skříni PC, kam je připojen komunikační kabel vyvedený z obvodu mezivrstvy. Pokud je obvod mezivrstvy k PC připojen prostřednictvím COM-USB konvertoru, je nutné správné číslo portu vy-



Obrázek 50: Konfigurační dialog

hledat ve Správci zařízení systému Windows.

Po zjištění správného čísla sériového portu je nutné jej zadat do konfiguračního dialogu výběrem z rolovací nabídky v horní části okna. Správnost nastavení lze jednoduše ověřit stiskem tlačítka označeného červeným otazníkem. Konfigurační dialog se pokusí poslat na určené číslo sériového portu žádost o odezvu (více viz. kapitola 4.5.2). Po dokončení této akce je uživatel informován o správnosti nastavení sériového portu. Pokud test komunikace neproběhne v pořádku, je možné jej opakovat s rozdílným nastavením čísla sériového portu. Dokud neproběhne test úspěšně, nelze pokračovat v práci s programem.

Soubory s exportovanými daty jsou implicitně umisťovány v adresáři určeném v konfiguračním dialogu (obr. 50) parametrem „umístění exportů z programu“. Editací pole vpravo od názvu parametru zobrazuje současné nastavení této cesty. Cesta může být změněna stiskem modrého tlačítka se symbolem otevřené složky. Po stisknutí je zobrazen dialog pro výběr adresáře. Při vlastním exportu však je možné umístit soubor do libovolného umístění na disku, tento parametr pouze urychluje výběr požadovaného adresáře.

Konfigurační soubor Parametry popsané v sekci 4.5.3 je nutné ukládat pro použití při dalším spuštění programu. V programech pro operační systém Windows se doporučuje provádět uschování konfiguračních parametrů jedním z těchto způsobů:

1. Uschování parametrů v systémové registrační databázi
2. Použití ini souborů

Systémová registrační databáze (známější spíše pod zkrácením označení „registr“, je hierarchicky uspořádaná databáze určená pro centralizované ukládání konfiguračních údajů. Největší nevýhodou tohoto řešení je **nepřenositelnost**. Obslužný software není potřeba při přenosu na jiný počítač instalovat, je nutné pouze složku s programem přkopírovat do požadovaného umístění. Tímto způsobem by se samozřejmě

údaje zanesené do systémové registrační databáze na cílový počítač nepřenesly. Nevýhodu by bylo nutné překonat speciálním instalačním (případně re-instalačním) programem. Program by se tímto zásahem neúčelně komplikoval, elegantnější je obejít tuto nevýhodu použitím ini souborů.

Ini soubor je obyčejný textový soubor typicky umístěný ve stejné složce jako spustitelný soubor programu. Nejjednodušší ini soubor může mít takovýto tvar

```
[Sekce] Klic=Hodnota
```

Parametr „sekce“ pouze upřesňuje umístění konfiguračního parametru a usnadňuje orientaci ve větších ini souborech. „Klic“ určuje jméno parametru a „Hodnota“ určuje položku identifikovanou klíčem „Klic“. Například pro uložení hodnoty **COM1** pro položku *komunikační port* a hodnoty **c:\Exports** pro položku *cesty k adresáři s exporty* vytvoříme následující ini soubor:

```
[Konfigurace]
CommPort=COM1
ExportPath=c:\Exports\
```

Obrovskou výhodou použití ini souborů je také snadný přístup k uloženým hodnotám, prostřednictvím třídy **TIniFile** standardně dostupné z prostředí Delphi.

Pro vytvoření výše uvedeného ini souboru postačí provést jednoduchý kus kódu:

```
var
  IniFile: TIniFile;
begin
  IniFile := TIniFile.Create('konfigurace.ini');
  IniFile.WriteString('Konfigurace', 'CommPort', 'COM1');
  IniFile.WriteString('Konfigurace', 'ExportPath', 'c:\Exports\');
  IniFile.Free;
end;
```

Pro načtení hodnoty u položky „CommPort“ lze jednoduše použít tento úsek kódu

```
var
  IniFile: TIniFile;
  Hodnota: String;
begin
  IniFile := TIniFile.Create('konfigurace.ini');
  Hodnota := IniFile.ReadString('Konfigurace', 'CommPort', 'COM1');
  IniFile.Free;
end;
```

Úsek kódu předpokládá existenci souboru pojmenovaného „konfigurace.ini“. Poslední parametr funkce `TIniFile.ReadString` určuje implicitní hodnotu vrácenou z funkce. Pokud funkce `TIniFile.ReadString` nalezne v souboru položku určenou druhým parametrem, vrátí implicitní hodnotu dle třetího parametru.

Překopírováním souboru ini umožňuje snadno přenést konfiguraci programu na jiný počítač, případně spravovat větší počet ini souborů pro různé scénáře nasazení programu.

Položky společné všem obslužným programům Tato část popisuje provedení úkolů, které jsou společné všem obslužným programům (programu pro *Poissonovo rozdělení*, *Dolet Beta částic* i *Zpětného rozptylu*).

Konfigurace Před použitím programu je třeba provést konfiguraci programu, konkrétně sériového portu, ke kterému je připojen obvod mezivrstvy. Více o konfiguraci programu v kapitole 4.5.3. Správnost nastavené konfigurace lze snadno ověřit volbou položky **Úkoly-Test Komunikace**, případně přímo tlačítkem na konfiguračním dialogu. Dialog zobrazený po otestování komunikace oznámí „Komunikace v pořádku“ nebo „Problém při komunikaci“. Při neúspěšném testu ověřte konfiguraci sériového portu a kabel propojující obvod mezivrstvy s osobním počítačem.

Export dat Další funkcí společnou všem programům je možnost exportu dat. Tato funkce byla podrobněji diskutována v části 4.5.3. Na tomto místě je nutno dodat pouze to, že export lze vyvolat v menu přes **Úkoly-Export dat**. Zvolením této položky zobrazíme dialog zachycený na obrázku 49.

Konfigurace Konfigurační dialog je přístupný z menu **Úkoly-Nastavení**.

Ukončení Poslední společnou položkou příslušející všem programům je ukončení programu dostupné přímo z hlavního menu volbou **Ukončit**.

Grafy Ve všech měřených úlohách je potřeba změřené hodnoty zobrazit v grafu.

Po základní práci s grafy dobře poslouží s Delphi standardně dodávaná komponenta `TChart`. Práce s ní je velmi snadná.

V grafech lze výhodně užít možnost zvětšování či zmenšování měřítka grafu (tzv. Zoom). Stačí kdekoli v grafu stisknout a podržet levé tlačítko myši a táhnutím označit oblast, které se potom zvětší na velikost původního grafu. Oblast musí být označována tažením zleva doprava. Naopak tažením myši zprava doleva je provedeno zmenšení grafu na původní velikost.

Celá oblast grafu lze libovolně posouvat, stačí stisknout a podržet pravé tlačítko myši kdekoli v oblasti grafu a myší posunout výřez do požadované oblasti v grafu.

Bohužel komponenta **TChart** neobsahuje všechnu funkčnost, která je pro naše účely potřebná.

Jedná se o neschopnost

- vykreslovat hladkou spojnicí bodů (obdoba XY bodového grafu známého z programu Microsoft Excel),
- provádět interpolaci a aproximaci změřených bodů a zobrazovat rovnici použité křivky.

Jelikož grafy musejí všechny z výše uvedených vlastností splňovat, musela být tato funkčnost doplněna přímo do obslužného programu.

K tvorbě hladké spojnice bodů je nutné najít matematický popis křivky, na kterou klademe podmínku na průchod *všemi* zadanými body.

Označme naměřené body $\{x_0, y_0\}, \{x_1, y_1\}, \dots \{x_i, y_i\}$. Body budou spojeny interpolačním polynomem n -tého řádu

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$$

Musí pro něj platit

$$P_n(x_0) = y_0, P_n(x_1) = y_1, P_n(x_2) = y_2, \dots P_n(x_i) = y_i$$

Slovně lze říci, že hledáme matematickou funkci (interpolační polynom) které prochází všemi změřenými body. Dodejme, že z tohoto důvodu (požadavku aby funkce procházela všemi body) není použití interpolace ve fyzice typické, každý naměřený údaj je vždy zatížen nějakou chybou a proto je častěji používána aproximace (popsaná v dalším textu), jelikož ta průchod všemi zadanými body nepožaduje.

Získáme soustavu $n + 1$ lineárních rovnic, hledáme koeficienty interpolačního polynomu $a_0, a_1, \dots a_n$. Soustavu lineárních rovnic je vhodné řešit numerickou metodou GEM — Gaussova eliminační metoda.

Gaussova eliminační metoda Soustava lineárních rovnic, přesněji řečeno jejich koeficientů, musí být přepsána do matice. Pokud je rovnic stejný počet jako neznámých, vznikne matice s počtem sloupců o jedničku větším než počtem řádků.

Pro ilustraci si představme následující soustavu rovnic

$$\begin{array}{rcl} (1) & x_1 + 2x_2 + 3x_3 & = 4 \\ (2) & 2x_1 + x_2 & = 4 \\ (3) & x_1 + 4x_2 + 2x_3 & = 3 \end{array}$$

Po zápisu do maticové podoby má soustava tento tvar

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 4 \\ 1 & 4 & 2 & 3 \end{array} \right)$$

Prvním krokem je úprava matice na takzvaný redukovaný stupňovitý tvar. V takové matici budou na diagonále samé jedničky. Toho lze samozřejmě dosáhnout pouze u čtvercových matic.

K úpravě na tento tvar užíváme elementární řádkové operace, což jsou:

- záměna libovolných dvou řádků,
- násobení kteréhokoli řádku nenulovým číslem (skalárem),
- sečtení řádků.

Pro naši soustavu bude úprava na redukovaný stupňovitý tvar následující:

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 4 \\ 1 & 4 & 2 & 3 \end{array} \right) \sim \left(\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 0 & -3 & -6 & -4 \\ 0 & 2 & -1 & -1 \end{array} \right) \sim \left(\begin{array}{ccc|c} 1 & 0 & 4 & 5 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{15}{2} & -\frac{11}{2} \end{array} \right) \sim$$

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{31}{15} \\ 0 & 1 & 0 & -\frac{2}{15} \\ 0 & 0 & 1 & \frac{11}{15} \end{array} \right)$$

Po úpravě matice na tento tvar jsou řešení vyseparována v pravé části matice, pro náš případ je tedy řešením soustavy rovnic $x_1 = \frac{31}{15}$, $x_2 = -\frac{2}{15}$, $x_3 = \frac{11}{15}$.

Právě popsaná metoda je velmi snadno implementovatelná na počítači, a proto byla použita v této práci pro hledání koeficientů interpolačního polynomu.

Aproximace Hlavním rozdílem mezi *aproximací* a výše popsanou *interpolací* je v tom, že při použití aproximačních metod **nepožadujeme**, aby graf výsledné funkce procházel změřenými body.

Pro výpočet aproximačního polynomu je použita metoda *nejmenších čtverců*.

Aproximační polynom v tomto případě není omezen podmínkou na průchod všemi zadanými body, ale součet čtverců rozdílu změřené hodnoty a hodnoty polynomu na konečné množině bodů musí být nejmenší.

Výsledný aproximační polynom má stejně jako u interpolačního polynomu podobu

$$P_m(x) = a_n x^m + a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + \dots + a_1 x + a_0$$

Aproximační polynom se volí pro naměřené hodnoty, protože jsou jistě zatíženy nějakou chybou a proto by nebylo účelné požadovat po křivce která je bude spojovat aby procházela všemi těmito body.

Pro implmenetaci metody Nejmenších čtverců byla použita na internetu volně dostupná knihovna `CurveFit.pas` vytvořená Davidem J. Taylorem.

5 Počítač jako prostředek motivace ve výuce fyziky

5.1 Pojem motivace

Kterákoliv lidská činnost by se jen stěží obešla bez motivace. Chceme-li poznat příčiny jakéhokoli jednání člověka, hledáme, co jej motivovalo. V psychologii je pojem motivace chápán většinou jako *hybná síla* ovlivňující chování. O tom, že se jedná o hybnou sílu, vypovídá i původ slova motivace — z latinského *moveo* — hýbat se. Motivace aktivizuje lidskou činnost, dává jí konkrétní směr a cíl. Je to základní podmínka cílevědomé aktivity jedince v každé oblasti.

Motivaci obvykle dělíme na:

Vnější: vyvstává obvykle z okolního prostředí jedince. Tyto zásahy zevnějšku někdy nazýváme jako *incentivy*. Jako příklad by bylo možno uvést, že se žák něčemu učí, protože to po něm chtějí rodiče. Dospělý pak může mít takovou motivaci tehdy, pokud se musí něco učit kvůli výkonu svého povolání apod.

Vnitřní: v tomto případě motivace většinou vyvstává z neuspokojené potřeby. Motivačním činitelem mohou být i instinkty nebo pudy. Právě biologické a sociální potřeby mohou být velmi silným motivačním činitelem.

Ve vzdělávacím procesu pracujeme obvykle s kombinací obou motivací. Vnější motivace může převládnout u žáků, kteří jsou do učení nuceni z vnějšku. Pokud potřeba vzdělání a poznání vychází z vnitřních potřeb žáka, je tato motivace většinou trvalejší a silnější. Naším pedagogickým cílem by mělo být správně s motivací zacházet a docílit toho, aby se další studium oboru stalo pro žáka silným motivem, který bude vycházet z jeho nitra. Pro správně motivované jedince je pak přijetí výkladu snazší a látka je lépe osvojena.

Motivace učební činnosti žáků Osobnost učitele a jeho chování v hodině je velmi důležitým motivačním faktorem (ačkoli samozřejmě ne jediným). Učitel se obvykle chová tak, aby navodil během hodiny vhodné podmínky pro učení jednak výběrem vhodných výukových metod (frontální experiment, názorné pomůcky, problémové vyučování, hra) ale i nepřímo, tím, jak se žáky komunikuje, způsobem odměn, trestů, apod. Učitele je tak možno zařadit mezi silné motivační činitele. Jak kladné, tak někdy bohužel i záporné.

Při obvyklém, tradičně používaném způsobu výuky je preferována tzv. *sociální vztahová norma*, kdy je výkon žáků hodnocen a srovnáván navzájem mezi sebou.

To může být pro některé jedince silně frustrující a může se stát zdrojem negativní motivace. Student pak prožívá pocity strachu nebo ohrožení, jeho výkon je tím zpravidla ještě snížen a ocitne se v začarovaném kruhu.

Vhodnějším a v moderní výuce více využívanou, je pak takzvaná *individuální vztahová norma*, kdy je současný výkon žáka hodnocen s jeho výkonem v minulosti, žebříček hodnocení a jeho stupně nejsou nastavovány podle ostatních studentů. I slabší žáci pak mají možnost okusit chuť úspěchu (např. zlepšením ze čtyřky na trojku, případně lichotivějším slovním hodnocením). Bez uspokojování výkonových potřeb nemůže žák ve škole existovat a příznaky ztráty motivaci by se měl každý pedagog snažit zachytit a snažit se proti nim bojovat.

Jednou z možností jak zvýšit motivaci a „atraktivitu“ předmětu může být zařazení počítače do výuky.

Počítač a zvyšování motivace žáků Asi každý se alespoň jednou za dobu své školní docházky na hodině nudil. Příčinu nudy lze nacházet ve frustrované poznávací potřebě, jejímž základem je vrozený orientačně pátrací reflex. Jeho neuspokojení unylým, monotónním výkladem vede žáka většinou k jedné z následujících činností:

Stažení se do sebe: dochází k vysazení pozornosti, žák je „duchem nepřítomný“.

Agresivní chování: agrese je běžně obrácena proti věcem (psaní po lavici, trhání sešitů), nebo spolužákům.

Rozptýlená aktivita: žák hledá náhradní aktivitu, vyrušuje a strhává na sebe i pozornost dalších.

Právě rozrušením stále stejným tempem plynoucí hodiny můžeme proti těmto nešvarům bojovat. Jak už bylo řešeno výše, k tomuto účelu může dobře posloužit právě experiment, ale i zařazení počítače do výuky.

Pomineme-li už jmenované oblasti využití počítačů ve výuce pro zvýšení motivace jej můžeme využít pro:

Výukové programy: V současné době trh s počítačovými programy nabízí široké spektrum výukových aplikací, pro všechny předměty, pro různé stupně škol. Touto kategorií se zde nebudeme podrobněji zabývat, analýza nabízených produktů a porovnání jejich kvality by jistě vydalo na celou monotematickou práci. Využití ve výuce však může být velmi prospěšné, může motivovat i slabší žáky a umožnit jim ve výuce postupovat svým tempem.

Programy pro testování: Také těchto komerčních programů lze najít na trhu celou řadu. Mezi výhody můžeme jmenovat, že testování (ověřování znalostí) nemusí být, zvláště pro slabší, tolik stresující, jako například ústní zkoušení

před celou třídou. Lze zde volit individuální přístup ke každému jednotlivci, nehodnotíme jej dle sociální vztahové normy, ale dle individuální. Všechny výsledky testů mohou být v počítači uloženy, je možné sledovat časovou křivku výkonnosti jednotlivých žáků a zachytit případné výchylky ve výkonu žáků. Tato časová informace také může posloužit jako prostředek stálé motivace (žák může sledovat konstantní zlepšování výkonu, apod.).

Demonstrační počítačové programy: předchozím textu už jsme o těchto programech hovořili, můžeme sem zařadit i počítačové modely a simulace vyvinuté v rámci této práce. Programy umožňují ukázat jinak nepřístupné fyzikální jevy (štěpnou reakci, kvantové vlastnosti mikrosvěta, ...), případně i suplovat chybějící materiální pokusy. Nemohou však „fyzicky“ prováděné experimenty zcela nahradit. Student by ztratil kontakt s fyzikální realitou a nevnímal pak předkládané fyzikální vztahy a zákonitosti jako reálně existující vlastnosti okolního světa, ale jako jakousi počítačovou hříčku bez spojení s okolní realitou.

Jednou z dalších možností pro zkvalitnění výuky a motivaci studentů je zařazení takzvaných *didaktických her*.

Didaktické hry Jedná se o metodu, která je v poslední době v souvislosti se snahou o zařazování alternativních výukových metod do výuky stále častěji zařazována. Hra a přirozená soutěživost žáků je velmi silně motivujícím stimulem, jež během hry umožňuje značnou aktivizaci jejich kognitivního potenciálu. Touto formou je tak možné přenést se přes náročné partie výuky snáze, než použitím „konvenčních metod“.

Využití této metody má i pozitivní dopady na ostatní formy rozvoje osobnosti než jen ty, na které je hra přímo zaměřena. Při hře v týmu se zlepšuje schopnost spolupráce na společném úkolu, zodpovědnost za svůj podíl práce na kolektivním cíli, komunikaci mezi členy skupiny a v neposlední době i na schopnosti vyrovnat se s nezdarem, zvyšovat frustrační toleranci.

Jako příklad takovéto počítačové aplikace, která uvádí do praxe výuky fyziky (a nejen její) výše uvedené zákonitosti didaktické hry jsou *fyzikální piškvorky*.

5.2 Didaktická hra — fyzikální piškvorky

Jedná se o autorem této práce vyvinutou počítačovou aplikaci, která již našla uplatnění na mnoha akcích Katedry obecné fyziky Pedagogické fakulty Západočeské univerzity v Plzni. Je používána při pravidelné výuce předmětu *Fyzika pro přírodní vědy*, byla součástí prezentace katedry na konferencích v Německu, Chorvatsku i součástí veřejné fyzikální soutěže pořádáné v rámci akce *Věda na ulici*. Na zahra-

ničních konferencích byly samozřejmě použity databáze s otázkami přeloženými do anglického a německého jazyka.

Aplikace je vyvinuta v jazyce C#, pro platformu Microsoft .NET. Je schopna běhu na počítačích s operačním systémem Windows, i dalších systémech, pro které existuje implementace platformy .NET.

Základní obrazovku aplikace ukazuje obrázek 51 a vychází z obecně známé hry, piškvorek. Učitel rozdělí třídu na dvě skupiny, jež střídavě volí otázky z hracího pole. V případě správně zodpovězené otázky toto pole připadá správně odpovídající skupině. Pole je označeno jejím symbolem (kříže nebo kolečka). Po spojení čtyř polí stejným symbolem vyhrává skupina které se to povedlo.

Aplikace vybírá otázky z databáze. Tato databáze má formát editovatelný aplikací Microsoft Access (součást kancelářského balíku MS Office) a tak je možné, vytvořením různých databází s otázkami obměňovat obtížnost soutěže, tematické zaměření, ale je možné aplikaci použít i v jiných předmětech.

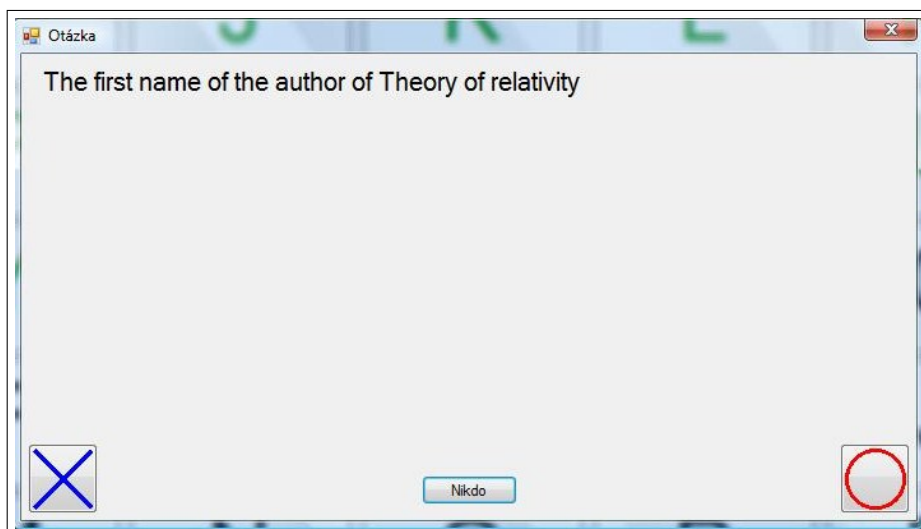
Dialog s otázkou a výběrem správně odpovídající skupiny ukazuje obrázek 52 a stav aplikace po spojení čtyř polí obrázek 53.



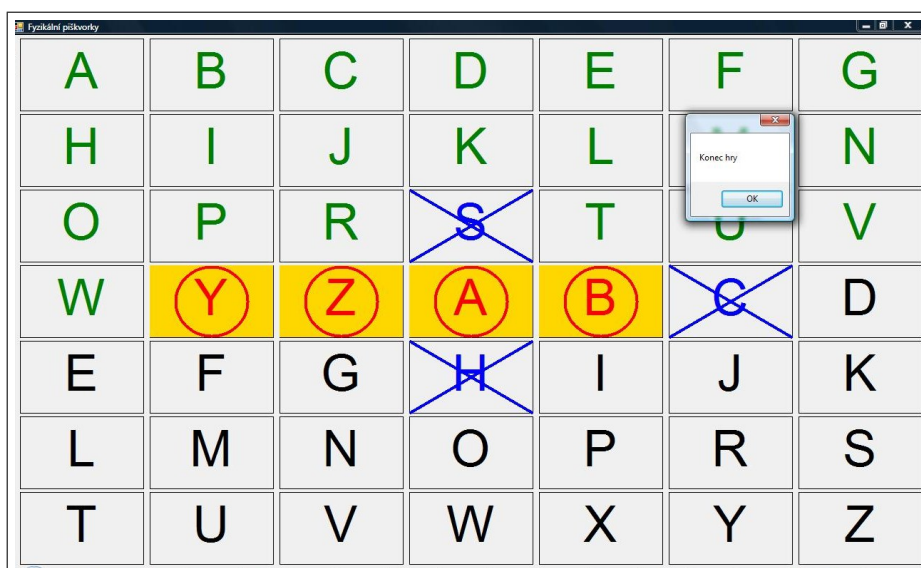
Obrázek 51: Aplikace fyzikální piškvorky, základní obrazovka

Zadávaní otázek v aplikaci MS Access ukazuje obrázek 54.

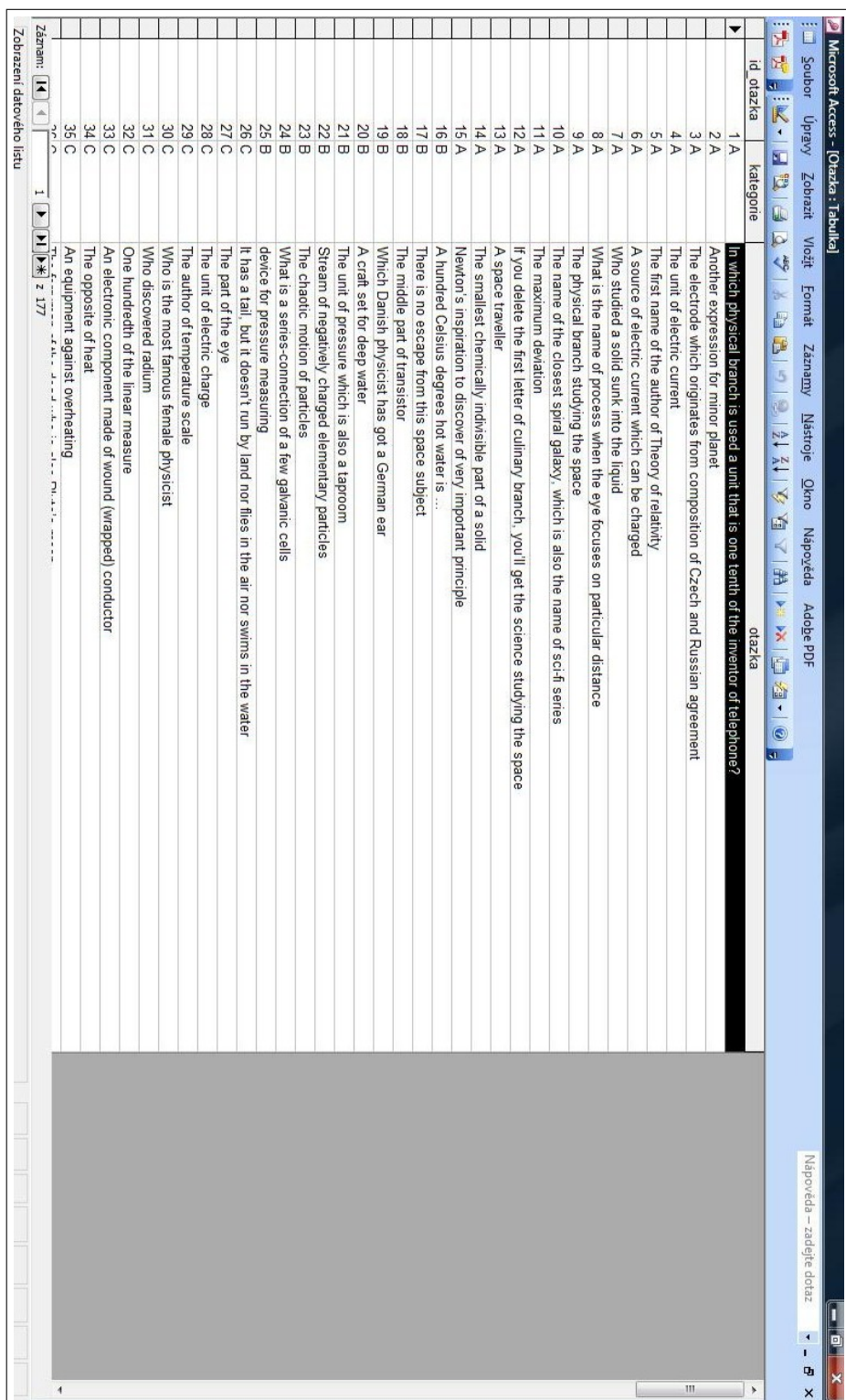
Se správně sestavenou databází otázek být může tato aplikace platným pomocníkem pro zvyšování motivace žáků ve vyučování.



Obrázek 52: Aplikace fyzikální piškvorky, výběr otázky



Obrázek 53: Aplikace fyzikální piškvorky, konec hry



Obrázek 54: Aplikace fyzikální piškvorky, zadávání otázek

6 Závěr

Tato práce hodnotí vybrané způsoby a možnosti počítačové podpory fyzikálního vyučování.

Kapitola 2 hovoří o tvorbě fyzikálních modelů a simulací. Hodnoceny byly postupy pro tvorbu těchto aplikací v prostředí *Codegear Delphi*. V tomto vývojovém prostředí byly vytvořeny a popsány fyzikální modely vybraných příkladů z teoretické mechaniky a jeden z atomové a jaderné fyziky. Tyto aplikace mohou být bez dalších úprav použity pro oživení výuky fyziky. Některé z těchto aplikací už dokonce ve výuce použity byly.

Se znalostmi popsaných detailů o tvorbě fyzikálních simulací a použitím předpřipravené „kostry“ programu není obtížné vytvoření aplikace s fyzikálním obsahem i pro zájemce s jen základní znalostí programování. „Kostru“ programu tak mohou použít například programátoři, kteří dříve fyzikální aplikace programovali v jazyce Pascal nebo PC FAND a přechodu na tvorbu aplikací pro MS Windows se z důvodu vyšší obtížnosti vyhýbali.

Druhá část kapitoly 2 vyhodnotila vhodnost prostředí *Adobe Flash*. I v tomto prostředí byly vytvořeny kompletní fyzikální aplikace.

Popisem velmi úspěšného a rozšířeného programu *Wolfram Mathematica*, se zabývá kapitola 3. Hodnoceny jsou i didaktické podmínky použití počítačové podpory matematického řešení fyzikálních problémů.

Úkolem kapitoly 4 této práce bylo vyvinout technické a softwarové prostředky umožňující měření úloh fyzikálního praktika, konkrétně úloh *Poissonovo rozdělení*, *Dolet beta částic* a *Zpětný rozptyl částic beta*.

Tyto prostředky měly nahradit stávající zastaralé programové vybavení určené k měření úloh a zmodernizovat uživatelské rozhraní pro provoz programů v prostředí MS-Windows.

Práce umožní do výuky zavést úlohu, která dosud nebyl měřena — *Zpětný rozptyl částic beta*.

Úlohu *Poissonovo rozdělení* lze nyní provádět demonstračně, což dříve kvůli vysoké době měření nebylo možné. Lze toho dosáhnout díky vysoké rychlosti detekce částic, tudíž možností použít pro měření radioaktivního zářiče. Statistické rozdělení četnosti lze ověřit mnohem rychleji.

Využití elektronického obvodu (v práci označován jako obvod mezivrstvy) není omezeno na výše zmíněné tři úlohy, lze k němu připojit libovolný přístroj s diskrétním výstupem a vhodným tvarem výstupních impulsů, více kapitola 4.5.1. Není tedy omezeno na oblast atomové a jaderné fyziky, ale vlastně ani fyziky jako celku. Narozdíl od ostatních řešení existujících v praxi, která často nabízejí více speciálně zaměřené produkty, obvod popisovaný v této práci nabízí velkou obecnost a varia-

bilitu. Logika zpracování dat je přemístěna z hardwarové části do části softwarové.

Nové obslužné programy lze vyvíjet pro jakýkoli operační systém v libovolném programovacím jazyce schopném komunikace po sériové lince. Komunikační protokol mezi obvodem mezivrstvy a osobním počítačem je popsán v kapitole 4.5.2.

K dalšímu rozvoji práce by pravděpodobně patřil vývoj nové verze obvodu mezivrstvy umožňující i zpracování spojitých signálů. Obvod by byl vybaven A/D převodníkem a komunikační protokol rozšířen o další příkazy.

Jinou možnou cestou budoucího vývoje by mohla být úprava obvodu mezivrstvy pro samostatnou práci. Všechny měřené hodnoty by uchovával obvod ve své paměti a byly by do osobního počítače přeneseny až najednou po skončení celého měření. Tento přístup by nevyžadoval připojení k osobnímu počítači po celou dobu měření. Obvod by pak bylo patrně nutné vybavit nějakým zobrazovacím prvkem (displejem) a ovládacími prvky (start měření, konec měření, reset měření, ...).

Za pomoci údajů z této práce není obtížné zhotovit větší množství obvodů mezivrstvy a použít je současně ve všech popisovaných úlohách.

Další možností zkvalitnění výuky a oblastí uplatnění počítačů ve vyučovacím procesu je jejich použití jako prostředku motivace studentů. O této oblasti hovoří kapitola 5. Motivace může být podpořena použitím *didaktických her*. Kapitola popisuje vytvořenou aplikaci fyzikální piškvorky, která může se správně sestavenou databází otázek zájem studentů o fyziku zvýšit.

Věřím, že programové a hardwarové vybavení i zkušenosti s jejich vývojem popsané v této práci významně přispějí ke zkvalitnění jak měření různých úloh fyzikálního praktika, tak výuku teoretickou.

Seznam obrázků

1	Příklad sestavení matematického modelu reálné fyzikální situace — kyvadlo	14
2	Demonstrační aplikace kyvadla	26
3	Nákres řešení dvojkyvadla	27
4	Demonstrační aplikace dvojkyvadla	29
5	Demonstrační aplikace dvojkyvadla, trajektorie hmotných bodů . . .	30
6	Nákres řešení kyvadla na kruhu	31
7	Demonstrační aplikace kyvadla na kružnici	32
8	Demonstrační aplikace kyvadla na kružnici	33
9	Kyvadlo s vodorovně pohyblivým závěsem, start	33
10	Kyvadlo s vodorovně pohyblivým závěsem, běžící simulace	34
11	Ohyb částic na štěrbině, start	35
12	Ohyb částic na štěrbině, 36536 emitovaných částic, šířka štěrbiny 10 nm	35
13	Ohyb částic na štěrbině, 236204 emitovaných částic, šířka štěrbiny 20 nm	36
14	Šikmý vrh míčku v Adobe Flash	40
15	Planckův vyzařovací zákon, teplota AČT 5000 K	42
16	Planckův vyzařovací zákon, posun maxima vyzařování s chladnutím AČT	43
17	Obrazovka programu Mathematica se základními mat. operacemi . .	45
18	Demonstrace funkce Plot	48
19	Funkce $\sin(x) + \sin(5 \cdot x)$	49
20	Funkce $\sin(x) + \sin(3 \cdot x)$	49
21	Pole vektoru intenzity elektrostatického pole	50
22	Vírové vektorové pole	51
23	Sloupcový graf	52
24	Lissajoussův obr. pro poměr frekvencí 1:3 a fázový posuv $\pi/2$	54
25	Lissajoussův obr. pro poměr frekvencí 1:2 a fázový posuv $\pi/3$	55
26	Hustota pravděpodobnosti výskytu částice v jednorozměrné potenci- álové jámě šířky 100 pm, kvantové číslo 1	57
27	Hustota pravděpodobnosti výskytu částice v jednorozměrné potenci- álové jámě šířky 100 pm, kvantové číslo 5	57
28	Hustota pravděpodobnosti výskytu částice v jednorozměrné potenci- álové jámě šířky 100 pm, kvantové číslo 100	58
29	Intenzita elektrického pole v okolí jednotkového náboje ve vakuu . . .	59
30	Potenciál elektrického pole v okolí jednotkového náboje ve vakuu . . .	59
31	Hlavní obrazovka programu <i>Poissonovo rozdělení</i>	63

32	Obrazovka programu <i>Poissonovo rozdělení</i> s naměřenými hodnotami .	65
33	Gaussovo rozdělení	66
34	Jednoduchá absorpční křivka	67
35	Superpozice dvou absorpčních křivek	68
36	Hlavní obrazovka programu <i>Dolet beta částic</i>	69
37	Měřicí dialog programu <i>Dolet beta záření</i>	71
38	Obrazovka programu <i>Dolet beta částic</i> s naměřenými hodnotami . . .	72
39	Obrazovka programu <i>Dolet beta částic</i> s naměřenými hodnotami a nastavenými čarami	72
40	Obrazovka programu <i>Dolet beta částic</i> grafem po odpočtení příspěvku záření X	73
41	Měřicí aparatura úlohy <i>Zpětný odraz beta částic</i>	76
42	Závislost N na Z v úloze <i>Zpětný rozptyl beta částic</i>	76
43	Závislost Z na N v úloze <i>Zpětný rozptyl beta částic</i>	76
44	Měřicí dialog programu <i>Zpětný rozptyl beta záření</i>	77
45	Obrazovka programu <i>Zpětný rozptyl beta částic</i> s naměřenými hod- notami	78
46	Schéma obvodu mezivrstvy	81
47	Vývody mikrokontroléru AT89C2051 na pouzdru DIP 20	83
48	Plošný obvod mezivrstvy	85
49	Dialog pro export dat	90
50	Konfigurační dialog	91
51	Aplikace fyzikální piškvorky, základní obrazovka	100
52	Aplikace fyzikální piškvorky, výběr otázky	101
53	Aplikace fyzikální piškvorky, konec hry	101
54	Aplikace fyzikální piškvorky, zadávání otázek	102

Reference

- [1] Havel V., Rauner K., Prokšová J.: *Fyzikální praktikum II*. Západočeská univerzita, Plzeň 1999.
- [2] Rauner K.: *Zpětný rozptyl částic beta*. Školská fyzika **VII**, mimořádné číslo (2001)
- [3] Matoušek D.: *Práce s mikrokontroléry AT89C2051*. GM Electronic, Praha 2000.
- [4] Kašpar E.: *Didaktika fyziky*. Severografia n.p., Liberec 1978.
- [5] Mechlová E.: *Didaktika Fyziky I*. Pedagogická fakulta, Ostrava 1983.
- [6] Fenclová J.: *Didaktické myšlení a jednání učitele fyziky*. Státní pedagogické nakladatelství, Praha 1984.
- [7] Fenclová J.: *Úvod do teorie a metodologie didaktiky fyziky*. Státní pedagogické nakladatelství, Praha 1982.
- [8] Richter J.: *Windows Pro pokročilé a experty*. Computer Press, Praha 1997.
- [9] Janás J.: *Kapitoly z didaktiky fyziky*. Masarykova univerzita, Brno 1996.
- [10] Korbel Z.: *Praktikum jaderné fyziky I*. Státní pedagogické nakladatelství, Praha 1971.
- [11] Beiser A., *Úvod do moderní fyziky*. Academia, Praha 1975.
- [12] Rauner K., *Disertační práce*. Plzeň 1993.
- [13] Vachek J., Lepil O.: *Modelování a modely ve vyučování fyzice*. Státní pedagogické nakladatelství, Praha 1980.
- [14] Rektorys K.: *Přehled užití matematiky*. SNTL, Praha 1973.
- [15] Komenský J. A.: *Vějička moudrosti neboli umění moudře svá díla opět probírat*. SNTL, Praha 1969.
- [16] Hameka H. F.: *Quantum mechanics*. A Wiley–interscience publication, New York 1981.