

**Náhodně generované interaktivní  
testy v PDF formátu pro výuku  
Matematické analýzy I**

**Randomly generated tests in  
interactive PDF format for the  
teaching of Calculus I**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012

.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Zvláště pak RNDr. Petře Šarmanové, PhD., vedoucí mé bakalářské práce a doc. Mgr. Robertu Maříkovi, Ph.D., tvůrci systému AcroWEB.

## Abstrakt

Cílem této bakalářské práce je vytvořit sbírku interaktivních testů ve formátu PDF pro podporu výuky Matematické analýzy I. Testy slouží k procvičení především teoretických otázek předmětu Matematická analýza I pro VŠB-Technickou univerzitu Ostrava.

Pomocí  $\text{T}_{\text{E}}\text{X}$ ových příkazů a  $\text{AcroT}_{\text{E}}\text{X}$ ových maker, vektorového editoru Inkscape vytvořím sbírku příkladů. Budu ji poté prezentovat pomocí webové stránky.

Předkládaná práce slouží jako manuál, obsahující veškeré informace pro vypracování obdobných testů.

**Klíčová slova:** matematická analýza I,  $\text{AcroT}_{\text{E}}\text{X}$ , Inkscape

## Abstract

The goal of this thesis is to create a collection of interactive tests in PDF format to support the teaching of Calculus. Tests used to practice primarily theoretical questions for Calculus I at VSB-Technical University of Ostrava.

Using  $\text{T}_{\text{E}}\text{X}$  commands and  $\text{AcroT}_{\text{E}}\text{X}$  macro, Inkscape-vector editor I create a collection of examples. Then I will present it at the website.

This report serves as a manual containing all the information to perform similar tests.

**Keywords:** Calculus,  $\text{AcroT}_{\text{E}}\text{X}$ , Inkscape

## Seznam použitých zkratk a symbolů

|                      |   |
|----------------------|---|
| PHP                  | – Personal Home Page                    |
| HTML                 | – Hyper Text Markup Language            |
| AcroT <sub>E</sub> X | – AcroT <sub>E</sub> X eDucation Bundle |
| PDF                  | – Portable Document File                |
| LAMP                 | – Linux Apache MySQL Perl server        |

## Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>                                      | <b>5</b>  |
| <b>2</b> | <b>AcroWEB</b>                                   | <b>6</b>  |
| 2.1      | Instalace . . . . .                              | 6         |
| 2.2      | Databáze . . . . .                               | 8         |
| 2.3      | Perl skript . . . . .                            | 9         |
| 2.4      | PHP skript . . . . .                             | 11        |
| <b>3</b> | <b>AcroT<sub>E</sub>X</b>                        | <b>13</b> |
| 3.1      | Začínáme . . . . .                               | 13        |
| 3.2      | Možnosti AcroT <sub>E</sub> Xu . . . . .         | 14        |
| 3.3      | Přehled základních příkazů . . . . .             | 15        |
| 3.4      | Prostředí quiz . . . . .                         | 15        |
| 3.5      | Modifikace prostředí quiz . . . . .              | 17        |
| <b>4</b> | <b>Vložení obrázků</b>                           | <b>18</b> |
| 4.1      | Rastrová vs. vektorová grafika . . . . .         | 18        |
| 4.2      | Postup vkládání vektorových obrázků . . . . .    | 19        |
| 4.3      | Inkscape . . . . .                               | 20        |
| 4.4      | Ovládání programu Inkscape . . . . .             | 20        |
| 4.5      | Postup tvorby grafu funkce . . . . .             | 21        |
| <b>5</b> | <b>Testy</b>                                     | <b>23</b> |
| <b>6</b> | <b>Webová stránka</b>                            | <b>25</b> |
| <b>7</b> | <b>Síťová část</b>                               | <b>27</b> |
| <b>8</b> | <b>Závěr</b>                                     | <b>30</b> |
| <b>9</b> | <b>Reference</b>                                 | <b>31</b> |
|          | <b>Přílohy</b>                                   | <b>31</b> |
| <b>A</b> | <b>Příloha-Slovníček pojmů</b>                   | <b>32</b> |
| A.1      | PHP . . . . .                                    | 32        |
| A.2      | Perl . . . . .                                   | 32        |
| A.3      | T <sub>E</sub> X, AcroT <sub>E</sub> X . . . . . | 32        |
| A.4      | AcroWEB . . . . .                                | 32        |
| A.5      | Klient . . . . .                                 | 33        |
| A.6      | Server . . . . .                                 | 33        |
| A.7      | Protokol . . . . .                               | 33        |
| A.8      | Internet . . . . .                               | 33        |
| A.9      | WWW . . . . .                                    | 33        |

## Seznam tabulek

|   |  |    |
|---|--|----|
| 1 | Rozdělení příkladů do náročností . . . . . | 9  |
| 2 | Testování počtu přístupů . . . . .         | 29 |

---

## Seznam obrázků

|    |  |    |
|----|--|----|
| 1  | Sekvenční diagram vytvoření PDF souboru . . . . .                    | 7  |
| 2  | Logo T <sub>E</sub> X . . . . .                                      | 13 |
| 3  | Obsah AcroT <sub>E</sub> Xu . . . . .                                | 14 |
| 4  | Základní podoba testu . . . . .                                      | 16 |
| 5  | Úvodní okno programu Inkscape . . . . .                              | 20 |
| 6  | Vykreslení grafu funkce . . . . .                                    | 22 |
| 7  | První stránka testu s úvodními informacemi . . . . .                 | 23 |
| 8  | Druhá stránka test s ovládacími a vyhodnocovacími tlačítky . . . . . | 24 |
| 9  | Rozvržení webové stránky . . . . .                                   | 25 |
| 10 | Úvodní stránka webové aplikace . . . . .                             | 26 |
| 11 | Schéma komunikace . . . . .  | 27 |
| 12 | Správné vyplnění přihlašovacích údajů programu WinSCP . . . . .      | 28 |



---

## Seznam výpisů zdrojového kódu

|   |  |    |
|---|--|----|
| 1 | Příkazy pro ruční vytvoření T <sub>E</sub> X souboru . . . . . | 6  |
| 2 | Základní podoba databáze . . . . .                             | 8  |
| 3 | Původní Perl skript . . . . .                                  | 9  |
| 4 | Část upraveného Perl skriptu . . . . .                         | 11 |
| 5 | Původní PHP skript . . . . .                                   | 11 |
| 6 | Část upraveného PHP skriptu . . . . .                          | 12 |
| 7 | Základní použití prostředí quiz . . . . .                      | 15 |
| 8 | Rozšířené použití prostředí quiz . . . . .                     | 16 |

## 1 Úvod

Mým zadáním je navrhnout sadu testů k procvičování vybraných témat předmětu „Matematická analýza I“.

V prvním kroku se seznámíme s prací doc. Mgr. Roberta Maříka, Ph.D., tvůrce systému AcroWEB. Doc. Mgr. Robert Mařík, Ph.D. vytvořil demostrační sadu kombinující Perl, PHP a jednoduchou databázi. Při tvorbě interaktivních testů využíváme balíček AcroTeX, který nám umožní vkládání java skriptů do PDF dokumentů. Tyto skripty opraví testy pomocí vlastní logiky, a proto nepotřebujeme připojení k internetu. Dále se seznámíme, jak řešení přenést na server a vytvořit webové rozhraní.

V druhém kroku rozšíříme demonstrační verzi generátoru testů o možnost vybírat otázky z různých textových souborů. Díky tomuto můžeme uživateli nabídnout procvičení témat, o které má zájem. Dále přidáme možnost vybírat různý počet otázek a jako největší výhodu možnost vkládat do testů otázky s obrázky.

V posledním kroku vytvoření databáze naplníme databázi otázkami k tématům Reálná čísla; Reálná funkce jedné reálné proměnné; Elementární funkce; Posloupnosti; Limita; Derivace; Základní věty diferenciálního počtu; Průběh funkce, Lokální extrém, Globální extrém; Aproximace funkce polynomem.

Hlavním přínosem je tedy vytvořený generátor testů, který je dostupný ze školní sítě na adrese: <http://srva10t.vsb.cz/acrotex/ramce.htm>

Pro správnou funkčnost vytvořených her je nutné používat snadno dostupný program Adobe Reader. Pouze tento program, od verze 5, dokáže interpretovat i javascriptové funkce PDF dokumentu.

## 2 AcroWEB

AcroWEB je nástroj k vytvoření testů na webu z databáze otázek. Základní nápad vytvoření AcroWEBu vzešel od doc. Mgr. Roberta Maříka, Ph.D. z Mendelovy univerzity. Než vytvoříme vlastní sadu testů, podíváme se na <http://wood.mendelu.cz/math/acroweb/demo.html>, kde je prentována ukázka.

### 2.1 Instalace

Než budeme moci vytvořit náš první PDF soubor, musíme systém nainstalovat.

Potřebujeme distribuci operačního systému Linux typu LAMP umístěnou nejlépe ve virtuálním stroji. Za zkratkou LAMP se skrývá označení Linux–název operačního systému, Apache–název webového serveru, MySQL–název databázového systému, který ovšem nepoužíváme, protože databáze otázek je uložena čistě v textové podobě a Perl–název programovacího jazyku, kterým spolu s PHP vytváříme PDF soubor.

Instalace ukázky AcroWEBu je velice jednoduchá. Musíme vložit adresář acrowebu (obsahující \*.html, \*.pl, \*.pm, \*.head, \*.tail, \*.txt a \*.tex soubory) do adresáře na http serveru, odzิปovat soubor phpfiles.zip také do tohoto adresáře (vytvoří acroweb.php, demo1.php, demo2.php, template2.php a template.php soubory) a otevřít soubor demo.html v prohlížeči.

Pro potřebu své aplikace jsem využil distribuci MAW, která je používána také na Mendelově univerzitě v Brně.

1. V prostředí námi vybraného operačního systému Linux zadáme příkaz `wget http://user.mendelu.cz/marik/temp/hist`. Tímto aktualizujeme Acro $\TeX$  a nahrajeme potřebné soubory pro odzkoušení do adresáře `/var/www/acroweb`.
2. Soubor si můžeme prohlédnout příkazem `cat hist`.
3. Zjistíme IP adresu virtuálního počítače (pomocí příkazu `ifconfig`) a přihlásíme se na něj. V mém případě byla tato adresa `http://192.168.81.2/acroweb/demo1.php`. Výstupem by mělo jednoduché PDF.

Alternativní formou můžeme vytvořit soubor ručně.

```
cd/var/www/acroweb
perl demo1.pl > /tmp/testtex
cd /tmp
pdflatex test.tex
```

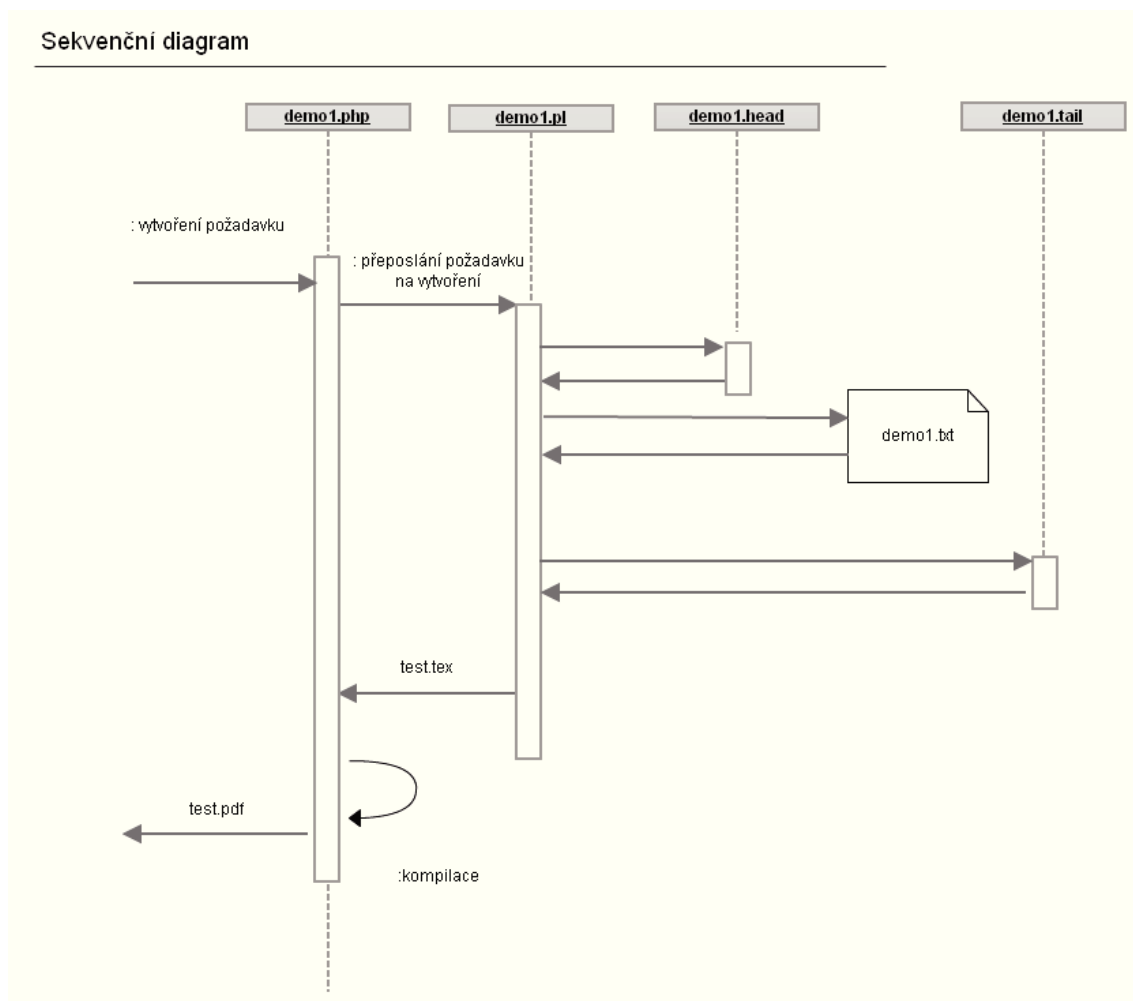
Výpis 1: Příkazy pro ruční vytvoření  $\TeX$  souboru

Výsledkem by měl být opět jednoduchý test.pdf.

Tímto jednoduchým postupem jsme si ověřili, že vytvořený virtuální stroj pracuje správně.

Vytváření PDF souboru vidíme na obrázku 1. Z adresy `http://192.168.81.2/acroweb/demo1.php` přeposíláme do Perl skriptu požadavek na vytvoření T<sub>E</sub>X souboru. Tento soubor složíme z souboru `demo1.head`, textového souboru `demo1.txt` a souboru `demo1.tail`. PHP skriptem jej zkompilujeme a můžeme zobrazit v prohlížeči, nebo uložit pro pozdější použití.

Cílem této práce je nejen ověřit funkčnost již vytvořeného. Zatím umíme vytvořit systém, který obsahuje vybírá otázky z jednoho souboru předem známý počet otázek. Dalším krokem se naučíme, jak jej naplnit vlastními otázkami.



Obrázek 1: Sekvenční diagram vytvoření PDF souboru

## 2.2 Databáze

Nejdůležitější částí této práce a v jistém smyslu mozek práce je databáze otázek [1].

Nejdříve popíšeme základní formát databáze využívající tohoto přístupu. Dále popíšeme vytvořenou databázi otázek, kterou využíváme v této práci.

Základní podoba databáze vypadá následovně:

---

```
První otázka -- na jediném řádku
+správná odpověď
-nesprávná odpověď
-další nesprávná odpověď
-poslední nesprávná odpověď
<jeden prázdný řádek>
Druhá otázka
-nesprávná odpověď číslo 1
-jiná nesprávná odpověď
+správná odpověď
-poslední nesprávná odpověď
<jeden prázdný řádek>
```

---

### Výpis 2: Základní podoba databáze

Zde < jeden prázdný řádek > znamená *blank line*.

Pro vytváření textového souboru musíme dodržovat několik základních pravidel:

1. Otázku můžeme napsat libovolně dlouhou, ale nesmíme v ní použít <Enter>, který způsobí komplikace při vytváření otázek.
2. Každou odpověď musíme napsat na novém řádku.
3. Správnou odpověď označíme znakem +, nesprávnou odpověď za znakem -.
4. Nemusíme dávat správnou odpověď na první místo.
5. Teoreticky můžeme použít větší počet odpovědí než je počet požadovaných odpovědí Perl skriptem. V mých textových souborech jsem však dodržoval pravidlo tří nesprávných odpovědí a jedné správné odpovědi.
6. Můžeme použít  $\TeX$  příkazy včetně matematických a další příkazy definovány v hlavičce  $\TeX$  souboru, které jsou definovány v \*.head, během kompilace.
7. Soubor bychom měli ukončit jedním prázdným řádkem.

Otázky jsem rozdělil do devíti témat a navíc jsem je rozlišil podle náročnosti na nenáročné, složité, nebo velmi náročné. Máme tedy celkově 27 textových souborů obsahující otázky.

Otázky k tématům „Průběh funkce“ a „Aproximace funkce polynomem“ jsem vymýšlel sám. Dále jsem soubory doplnil o otázky s obrázky, které jsou nepostradatelné k pochopení základních pojmů matematické analýzy.

|                             |                   |                         |
|-----------------------------|-------------------|-------------------------|
| Reálná čísla                |                   |                         |
| nenáročné–15 otázek         | složité–12 otázek | velmi náročné–12 otázek |
| Funkce                      |                   |                         |
| nenáročné–14 otázek         | složité–12 otázek | velmi náročné–8 otázek  |
| Základní funkce             |                   |                         |
| nenáročné–15 otázek         | složité–13otázek  | velmi náročné–12 otázky |
| Posloupnosti                |                   |                         |
| nenáročné–14 otázek         | složité–10 otázek | velmi náročné–8 otázek  |
| Limita                      |                   |                         |
| nenáročné–8 otázek          | složité–8 otázek  | velmi náročné–7         |
| Derivace                    |                   |                         |
| nenáročné–18 otázek         | složité–17 otázek | velmi náročné–7         |
| Základní věty               |                   |                         |
| nenáročné–8 otázek          | složité–10 otázek | velmi náročné–2         |
| Průběh funkce               |                   |                         |
| nenáročné–7                 | složité–7 otázek  | velmi náročné–8         |
| Aproximace funkce polynomem |                   |                         |
| nenáročné–5                 | složité–3         | velmi náročné–3         |

Tabulka 1: Rozdělení příkladů do náročností

## 2.3 Perl skript

Stejně jako samotný mozek musí být ovládán nervovými vzruchy i samotná databáze by byla zbytečná, kdybychom s ní nemohli dále pracovat.

Úkolem Perl skriptu je vybírat otázky z databáze, předpřipravit je pro kompilaci.

Původní Perl skript vypadá následovně:

---

```

use acroweb_eng;
system("cat_ukazka.head");
print genrandom("test.txt",4,4,0) ;
system("cat_ukazka.tail");

```

---

Výpis 3: Původní Perl skript

Prvním příkazem čteme knihovnu `acroweb_eng`; s makry. Stavíme  $\text{\TeX}$  soubor ze tří částí: z hlavy (soubor označený `*.head`), textových souborů (náhodně generovaných testů) a paty (soubor označený `*.tail`).

Hlava (soubor označený `*.head`) obsahuje úvodní data pro překladač jako je definice druhu dokumentu, definování použitých balíčků, makra a úvodní informace k vytvoření PDF dokumentu.

---

Následuje funkce `genrandom („test.txt“ ,4,4,0)`;

- která čte soubor `test.txt` (první parametr),
- vybírá z něj 4 náhodné otázky (druhý parametr označený nadtržením),
- třetí parametr určuje počet nesprávných odpovědí,
- poslední parametr může mít hodnoty 0, 1, nebo 2. Hodnota 0 znamená: prostě vybereme 4 nesprávné odpovědi, hodnota 1 pracuje podobně vybere také 4 nesprávné odpovědi, ale poslední odpověď je zaměněna za frázi „another answer“. Hodnota 2 je užita pro vyplňovací otázky.

Pata (soubor označený `*.tail`) obsahuje ukončovací část párových příkazů  $\TeX$ u a vytváří jednoduchou tabulku pro vyhodnocení.

Pomocí vzorového skriptu umíme tedy vytvořit právě jeden soubor `test.tex`, který má pevně stanovený počet otázek—4. Z tabulky (1) plyne, že nejvíce otázek má téma „Derivace“ v nenáročné variantě. Neupravíme-li skript, můžeme sice vytvořit tisíce testů, přesněji 3060, ale uživatelé si jistě všimnou, že je jim předkládáno pouze 18 otázek.

Využijeme všech 27 textových souborů a přidáním několika podmínek znásobíme počet různých variant k miliardám. Díky PHP skriptu získáváme proměnnou `firquan`, která určuje počet otázek je číslo od jedné do pěti, a proměnnou `firvar`, která určuje náročnost testu.

---

```

use acroweb_eng;
system("cat_ukazka1.head");
### the multichoice question with open choice (=none of the above)
if (( $\$firquan \geq 1$ ) and ( $\$firquan \leq 5$ ))
{
if ( $\$firvar == 1$ )
{
print genrandom("1A1.txt", $\$firquan$ ,4,0);
}
if ( $\$firvar == 2$ )
{
print genrandom("1B1.txt", $\$firquan$ ,4,0);
}
if ( $\$firvar == 3$ )
{
print genrandom("1C1.txt", $\$firquan$ ,4,0);
}
%\hline
}
system("cat_demo1.tail");

```

---

Výpis 4: Část upraveného Perl skriptu

## 2.4 PHP skript

Posledním krokem je vytvoření jednoduchého rozhraní pro uživatele, aby si mohl vybírat požadované otázky. PHP skript volá Perl skript, od kterého dostává T<sub>E</sub>X soubor, a ten kompiluje a přeměrovává výstup na prohlížeč. Původní PHP skript vypadá následovně:

---

```

<?php
require (" ./acroweb.php");
// create temporary directory
 $\$adresar = \text{tmp}/\text{acrotex}.\text{getmypid}()$ ;
system("mkdir_ $\$adresar$ ");
system("perl_demo1.pl_>&_ $\$adresar$ /test.tex")
// clean temp directory
compile\_acroweb("demo-1", $\$adresar$ );
?>

```

---

Výpis 5: Původní PHP skript

Prvním příkazem čteme soubor acroweb.php, který obsahuje zapouzdřenou funkci `compile_acroweb`. Na druhém řádku odvozujeme jméno pracovního adresáře z počtu běžících procesů a na třetím řádku vytváříme tento adresář. Další řádek volá Perl skript, jehož výstupem je blok testů v T<sub>E</sub>X. Ten je přeměrován na soubor test.tex v pracovním adresáři. Na posledním řádku voláme funkci `compile_acroweb`. Jejím úkolem je vytvořit pomocí pdf<sub>A</sub>T<sub>E</sub>X PDF soubor. Dále napíše jednoduchý log obsahující datum, čas IP adresu a jméno testu ve formě, kde "jméno testu" je první parametr funkce `compile_acroweb`. Tato funkce také odstraní pracovní adresář.



---

Tento skript jsem upravil pro čtení proměnných z .html stránky. Tyto proměnné skript upraví a pošle.

---

```
<?php&
require (". /acroweb.php");
// create temporary directory
$adresar="tmp/acrotex".getmypid();
system("mkdir_.$adresar");
$firquan=$_GET['firquan'];
$firquan=intval ($firquan);
settype($firquan, "integer");
$firvar=$_GET['firvar'];
$firvar = intval ( $firvar );
settype( $firvar , "integer");
system("perl_ -s_pocetotl.pl
-firvar=$firvar _-firquan=$firquan
// clean_temp_directory&
compile_acroweb("demo-1",$adresar);
?>$
```

---

Výpis 6: Část upraveného PHP skriptu

### 3 Acro $\TeX$

Systém  $\TeX$  používáme pro sazbu dokumentů ve vysoké typografické kvalitě, především pro svou preciznost v sazbě matematických a chemických vzorců. Tento systém, vyvinutý v 70. letech 20. století, je volně šiřitelný s možností vytváření dokumentů ve formátu PDF.



Obrázek 2: Logo  $\TeX$

V této práci využíváme i Acro $\TeX$  eEducation Bundle (dále zkráceně AeB) [2]. Můžeme vytvářet testy, kvízy a otázky ve formátu PDF. Navíc tyto testy můžeme i vyhodnotit a opravit.

V současné době se AeB skládá ze sedmi částí balíků: web, exerquiz, eforms, insdljs, taborer, dljslib a eqExam. Ve své práci nejvíce používáme především balíčky exerquiz a web. Ostatní slouží především k prezentačním účelům.

Uvedu zde ještě několik úvodních poznámek: Acro $\TeX$  podporuje pouze tři cesty výroby PDF souboru:

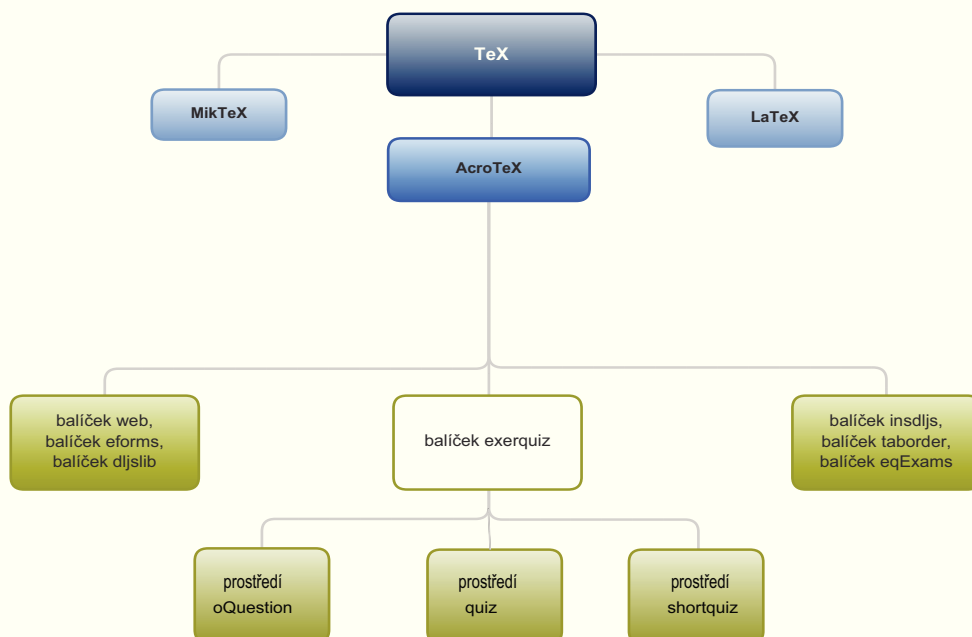
1. Adobe Acrobat Distiller (verze 5.0 nebo vyšší);
2. pdftex;
3. dvipdfm.

Celý systém Acro $\TeX$  je založený na vkládání JavaScriptů do formátu PDF. Jediný prohlížeč, který dokáže tyto JavaScripty zpracovat je velice rozšířený Adobe Reader.

#### 3.1 Začínáme

Distribuce balíčku Acro $\TeX$  se skládá ze dvou částí:

- acrotex\_pack.zip: Tento zip soubor obsahuje programové soubory (web, exerquiz, eforms, insdljs, dljslib, taborder a aeb.js), manuály k Acro $\TeX$ u a eForms manuál.
- acrotex\_exdoc.zip: Tento zip soubor obsahuje všechny další soubory, rozšířenou dokumentaci a bohatou sbírku příkladů.



Obrázek 3: Obsah AcroTeXu

### 3.2 Možnosti AcroTeXu

Pomocí tohoto systému umíme formátovat otázky s odpovědmi do několika typů a dále vytvořit k nim jednoduché uživatelské rozhraní.

- Jednou z možností je výčet s výběrem z nabízených odpovědí. V této variantě uživatel označí jednu správnou odpověď.
- Další možností je zapsání odpovědi do textového pole. Uživatel vepíše do pole text, který se porovnává se správnou odpovědí zadanou autorem ve zdrojovém dokumentu.
- Nejzajímavější pro testování v matematice je možnost zadání výsledku jako výraz. Se zapsaným výrazem je zacházeno jako s matematickým. Kontrola zapsaného matematického výrazu proběhne dosazením náhodných bodů z autorem zadaného intervalu. Náhodné body se dosadí do uživatelem zapsaného výrazu a zároveň do autorovy správné odpovědi. Po získání těchto hodnot systém výsledky porovná a vyhodnotí výsledek při shodě ve všech porovnávaných bodech za správný (přesnost shody je nastavena autorem testu). Zapisovaný výsledek je tedy ověřován z hlediska matematické správnosti nikoli z hlediska shodnosti textu. Tento přístup je

výhodný zejména v případě, pokud je zadávaným výsledkem funkce s jednou nebo i více proměnnými. I složitější výraz, který uživatel může zapsat v různé podobě (neroznásobené jednotlivé členy, jiné pořadí členů,...) je vyhodnocem výše napsaným způsobem.

### 3.3 Přehled základních příkazů

Balíček `exerquiz` umožňuje vytvářet testy, kvízy i kvízy s vyhodnocením třemi způsoby.

- **oQuestion**– prostředí umožňuje vytvoření jednoduché otázky, pro jejíž odpověď je určeno textové, nebo matematické pole, obsahuje vždy jednu otázku
- **shortquiz**– prostředí pro vytvoření krátkého kvízu, ve kterém je uživatel informován o správnosti své odpovědi pomocí vyskakovacího okna
- **quiz** – prostředí pro vytvoření rozsáhlejšího testu, ve kterém dojde k vyhodnocení otázek až po ukončení celého testu

### 3.4 Prostředí quiz

V tomto prostředí můžeme vytvořit dva druhy testů. Napsané ve formuláři, nebo napsané v řádku. Pro vytvoření formulářového testu napíšeme příkaz `\quiztype{f}`, pro vytvoření obdobného testu v podobě seznamu `\quiztype{l}`.

Při nejjednodušším používání prostředí napíšeme pouze tři příkazy. Jeden do souboru `*.head`, druhý do souboru `*.tail`:

---

```
\quiztype{f}
\begin{quiz}{jmeno_testu}
....
....
....
....
\end{quiz}
```

---

#### Výpis 7: Základní použití prostředí quiz

Musíme mít na paměti, že k vytvoření kvízu je skrytě používán JavaScript, proto pro vytvoření kvízu dodržujeme několik pravidel:

1. k pojmenování testu musíme využít základní písmena latinské abecedy (A..Z, a..z), podtržítka (`_`), nebo symbolu dolar (`$`),
2. číslice se nesmí vyskytovat na začátku pojmenování,
3. prostředí `quiz` naplňujeme dvěma typy prostředí `question` a `answers`,
4. prostředí `question` je nadřazeno prostředí `answers`, prostředí `quiz` je nadřazeno oběma prostředí,

5. prostředí `answers` musíme doplnit o počet sloupců. Pokud nastavíme počet sloupců na 1 vytvoříme seznam, jinak se vytvoří prostředí podobné `tabular`. Sloupce oddělujeme ampersandem (`&`), řádky vytvoříme dvojitým zpětným lomítkem (`\\`),
6. příkaz `\Ans` doplňujeme o 0 v případě špatné odpovědi, nebo o 1 v případě správné odpovědi,
7. pomocí příkazu `\ScoreField\currQuiz` vytváříme pole, ve kterém se zobrazují informace o výsledku,
8. k opravování testu napíšeme dva příkazy `\eqButton\currQuiz` volající definované makro.

---

```

\quiztype{f}
\begin{quiz}{derivace}
\begin{questions}

\problem{Rozhodněte, které z následujících tvrzení je pravdivé}
\begin{answers}{1}
\Ans0 ....\\[5 pt]
\Ans1 ....\\[5 pt]
\Ans0 ....\\[5 pt]
\Ans0 ....\\[5 pt]
\end{answers}

\end{questions}
\end{quiz}
\ScoreField\currQuiz

```

---

Výpis 8: Rozšířené použití prostředí `quiz`

2. Rozhodněte, které z následujících tvrzení je pravdivé.

- $\lim a_n = 5 \Rightarrow (\exists n_0 \in \mathbb{N}) (\forall n \in \mathbb{N}, n > n_0) : a_n > 5.$
- $\lim a_n = 5 \Rightarrow (\exists n_0 \in \mathbb{N}) (\forall n \in \mathbb{N}, n > n_0) : a_n < 6.$
- $\lim a_n = 5 \Rightarrow (\exists n_0 \in \mathbb{N}) (\forall n \in \mathbb{N}, n > n_0) : a_n \geq 5.$
- $\lim a_n = 5 \Rightarrow (\exists n_0 \in \mathbb{N}) (\forall n \in \mathbb{N}, n > n_0) : a_n \leq 5.$

Obrázek 4: Základní podoba testu

### 3.5 Modifikace prostředí quiz

Základní podobu můžeme také upravovat. Všechny možné modifikace jsou uvedeny v [2]. Uvedu jen několik základních možností grafické úpravy.

1. Můžeme změnit titulek, velikost tlačítek a orámování tlačítek.
2. Můžeme změnit vzhled zaškrťovacích symbolů.
3. Můžeme upravit titulek, velikost a orámování vypisovacího pole.
4. Můžeme rozšířit obsah vypisovacího pole na více řádků. Po přidělení bodů jednotlivým otázkám zjistit počet bodů, percentil i ohodnotit test známkou dle námi zvolené stupnice.

## 4 Vložení obrázků

Zatím jsem popisoval známé a už implementované postupy. Spojení AcroT<sub>E</sub>Xu a AcroWEBu vytvořilo test obsahující pouze text. Pro testování pouček a definic skvělý nástroj. S pojmem matematická analýza se téměř každému ale vybaví grafy funkcí. Proto jsem se rozhodl, rozšířit stávající řešení o možnost vkládání obrázků. Dříve než však ukážu mé řešení, čtenář může řešit následující otázky:

- Jaký formát obrázku je pro tuto práci nejvhodnější?
- Jak postupovat, abych mohl vložit obrázek k testové otázce?
- Jak se změní T<sub>E</sub>Xovský kód?

### 4.1 Rastrová vs. vektorová grafika

Základní reprezentace obrázků se dělí na formu rastrovou a formu vektorovou.

Rastrová forma je tvořena posloupností informací o každém bodu (pixelu) daného obrazu. Každý bod má určen svou přesnou polohu a barvu v nějakém barevném modelu (např. RGB).

Nevýhody rastrové grafiky:

- velké nároky na zdroje (při vysokém rozlišení a barevné hloubce velikost obrázku dosahuje i jednotek megabytů, v profesionální grafice se běžně operuje i s podklady o desítkách megabytů).
- změna velikosti (zvětšování nebo zmenšování) vede ke zhoršení obrazové kvality obrázku.
- zvětšování obrázku je možné jen v omezené míře, neboť při větším zvětšení je na výsledném obrázku patrný rastr

Už první nevýhoda ukazuje, že rastrový obrázek není vhodný. Jedná se o webovou aplikaci, jejíž odezva by měla být v milisekundách. Jediný kvalitní obrázek by z testu, který má řádově stovky kB, vytvořil zbytečně velký soubor ke stažení.

Druhá a třetí jen potvrzují, že rastrový obrázek není tím pravým řešením.

Na druhé straně vektorová forma je tvořena posloupností příkazů pro vykreslení jednotlivých elementů daného obrazu (přímek, elips, mnohoúhelníků a podobně včetně nejrůznějších parametrů– barva, sklon, typ čar).

Vektorová grafika má proti rastrové grafice některé výhody:

- Výsledná paměťová náročnost obrázku je obvykle mnohem menší než u rastrové grafiky.
- Je možné libovolné zmenšování nebo zvětšování obrázku bez ztráty kvality.
- Je možné pracovat s každým objektem v obrázku odděleně.

Paměťově nenáročný obrázek je tím, co hledáme. Pro úplnost uvedeme i nevýhody vektorové grafiky.

- Oproti rastrové grafice zpravidla složitější pořízení obrázku. V rastrové grafice lze obrázek snadno pořídit pomocí fotoaparátu nebo skeneru.
- Překročí-li složitost grafického objektu určitou mez, začne být vektorová grafika náročnější na operační paměť a procesor než grafika bitmapová.

Tento typ nám poskytne hledaný kvalitní výstup i pro grafy funkcí.

## 4.2 Postup vkládání vektorových obrázků

Tento postup zahrnuje dva kroky, které si vysvětlíme:

Aby byl  $\text{\TeX}$  schopen zacházet s různými druhy obrázků, je potřeba použít správný ovladač. Ovladač se zavádí použitím parametru při vkládání balíku `graphics` (nebo jeho rozšířené verze `graphicx`). Například `\usepackage[dvips]graphics`. Nejpoužívanější ovladače jsou `dvips` (obrázky `.ps` a `.eps`) a `dvipdf` (obrázky `.pdf`). První se používá se standardním  $\text{\LaTeX}$ em, druhý s `pdf\LaTeX`em.

$\text{\TeX}$  musí vědět, který obrázek se má použít. Proto využijeme příkaz pro vkládání obrázků `\includegraphics`, jehož jediným parametrem je jméno daného obrázku.

Postup se tedy skládá ze dvou kroků:

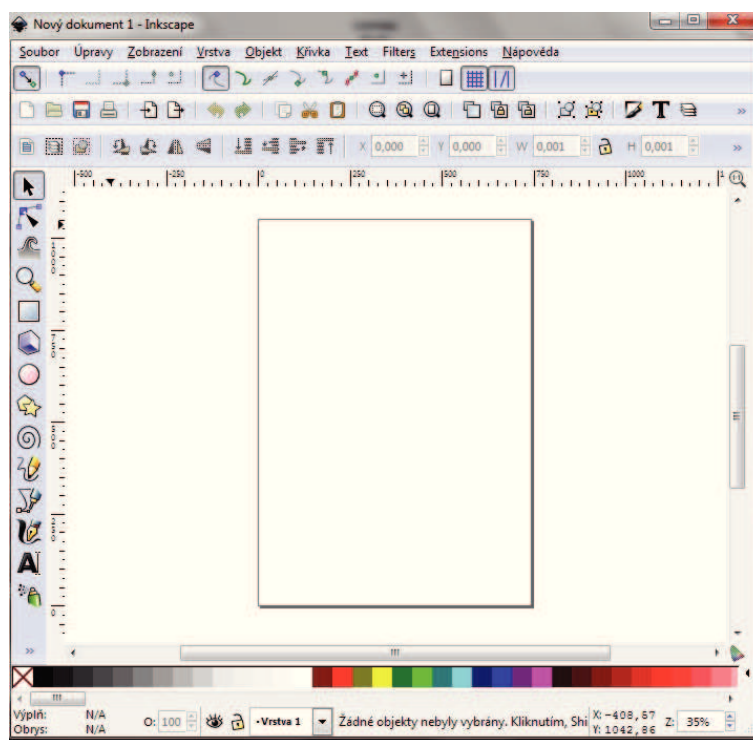
1. Příkaz `\usepackage[dvips]{graphics}` tedy vložíme do souboru `*.head`.
2. Příkaz `\includegraphics{jmeno.pdf}` umístíme do textového souboru, kde danou otázku používám.



## 4.3 Inkscape

Pro vytvoření vektorových obrázků musíme zvolit vhodný editor. Já jsem pracoval s volně šiřitelným programem Inkscape. Obsahuje příjemné uživatelské rozhraní pro vykreslování grafů, které v krátkosti představím.

Základní okno Inkscape obsahuje několik základních ovládacích prvků.



Obrázek 5: Úvodní okno programu Inkscape

## 4.4 Ovládání programu Inkscape

### 4.4.1 Horní lišta

Nejvýše v okně programu se nachází horní lišta, která obsahuje na levé straně logo programu s názvem obrázku (po otevření nového implicitně Nový dokument). Na pravé straně se nachází standardní tlačítka pro práci s okny—Minimalizovat, Maximalizovat a Zavřít.

### 4.4.2 Lišta menu

Druhý řádek slouží k ovládání programu. Nabízí se v něm několik roletových nabídek, které se po klepnutí myší rozbalí.

### 4.4.3 Rychlé volby

Na třetím vodorovné řádce se nachází ikony nejpoužívanějších funkcí, například: Nový soubor, uložit soubor, Zpět a Vpřed a mnoho dalších.

### 4.4.4 Panel voleb

Poslední vodorovný řádek se mění podle vybraného pracovního nástroje. V tomto řádku nastavuji potřebné vlastnosti a parametry objektů. V pravém horním rohu vidím právě používané barvy, barvu výplně a obrysu objektů. Pokud chci zvolit jiné, v dolní části okna můžu pomocí posouvátka vybrat z široké škály barev.

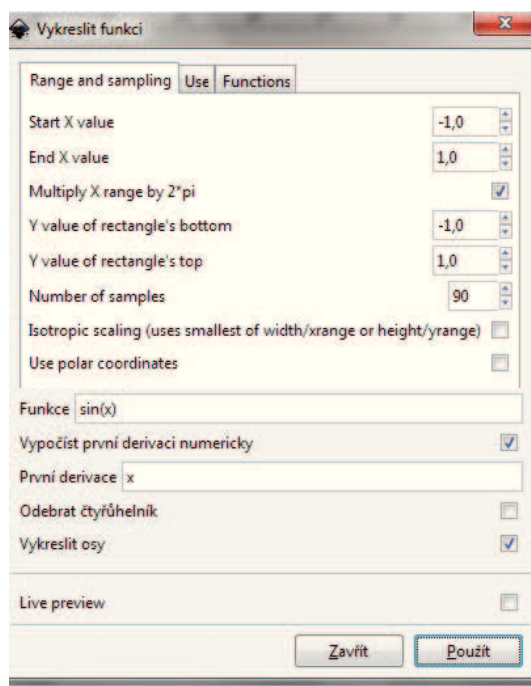
### 4.4.5 Panel nástrojů

Pro kreslení využívám panel se sedmnácti ikonami, který se nachází na levé straně. Nástroje můžu ovládat i pomocí klávesových zkratk. Mnou nejpoužívanější byly:

- F1–Výběr a transformace objektů
- F4–Tvorba obdélníků a čtverců
- Shift+F7–Vyplnění ohraničených ploch
- F8–Tvorba a úprava textových objektů

## 4.5 Postup tvorby grafu funkce

1. Pro vytvoření jednoduchého grafu otevřu nový soubor. Inkscape obsahuje velké množství předdefinovaných kreslicích pláten. Já jsem používal kreslicí prostředí pojmenované A4, web\_banner\_728x90 a  $\LaTeX$ .Beamer.
2. Poté vytvořím obdélník, do kterého budu graf funkce kreslit.
3. Pro vykreslení grafu funkce vyberu z horní části obrazovky nabídku Efekty->Vykreslit->Vykreslit funkce. Touto volbou se otevře dialogové okno s nabídkou nastavení vlastností.
4. Dialogové okno (zobrazené na obrázku) obsahuje velice příjemné uživatelské rozhraní, které není třeba popisovat. Pro editaci pak můžu použít panel nástrojů. Především možnost popsat osy grafu funkce vypadá jako velice užitečná. Další možností můžeme v menu Křivka vybrat funkci Rozdělit na části, které umožní rozdělit graf na několik částí.



Obrázek 6: Vykreslení grafu funkce

## 5 Testy

V této kapitole uvedeme pohled uživatele a popíšeme si, jak vypadá PDF soubor. Popíšeme si, jak pracovat s tímto dokumentem.

Po otevření PDF souboru uvidíme první stranu. Tato stránka neobsahuje žádné interaktivní prvky, ale obsahuje neméně důležité informace. Nejdříve nás zaujme nadpis, který nám ukazuje, pro který předmět byl tento test vytvořen. Dále vidíme jednoduchý návod, jak můžeme pracovat s tímto testem. V pravém dolním rohu pak čteme informace o autorském právu a také datum sdělující, kdy byl test vytvořen.

### Diferenciální počet funkcí jedné proměnné

Tomáš Majda z originálu  
pana Roberta Mařika

1. Před začátkem testu klikni na tlačítko "Zacatek testu".
2. Vyber vždy jednu správnou odpověď.
3. Pro ukončení testu klikni na tlačítko "Konec testu".
4. Test je poté vyhodnocen a počet správných odpovědí je vyznačen v poli.
5. Správné odpovědi se zobrazí po kliknutí na tlačítko "Opravit".
6. Test začíná na následující straně.

Copyright © 2012  
Poslední změna 3. května 2012

Obrázek 7: První stránka testu s úvodními informacemi

Na druhé stránce už uvidíme samotný test. Nejvýše je umístěno zahajovací tlačítko „Zacatek testu“. Dále nalezneme část vyhrazenou otázkám. Na jedné stránce uvidíme zpravidla dvě otázky a pod každou otázkou čtyři možné správné odpovědi. K listování v testu používáme jednoduché a dvojité šipky. Jednoduché šipky nám ukáží předcházející nebo následující stránku testu, dvojité nás posunou na začátek nebo konec dokumentu.

Na druhé stránce vidíme i poslední části souboru, kterou používáme k ukončení a vyhodnocení námi zvolených odpovědí. Pro ukončení testu klikneme na tlačítko „Konec testu“. Okamžitě uvidíme výsledek v červeně ohraničeném poli. Chceme-li se dozvědět správné odpovědi, musíme kliknout na tlačítko „Opravit“. Při opravování testu uvidíme své zaškrtnutí zeleně, jestliže jsme odpověděli správně. Pokud jsme vybrali špatnou

odpověď, naše zaškrtnutí přepíše červený křížek a zelený bod nám ukáže správnou odpověď.

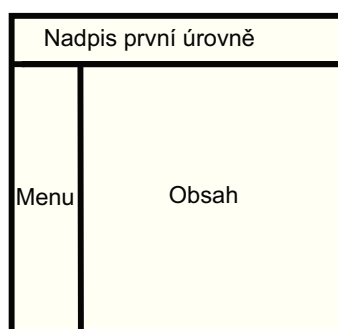
The image shows a quiz interface. At the top right, the number '2' is displayed. Below it, there is a button labeled 'Zacatek kvizu'. The main content is a question: '1. Rozhodněte, které z následujících tvrzení je pravdivé. (6)'. There are four radio button options: 'Každá geometrická posloupnost je konvergentní.', 'Každá omezená aritmetická posloupnost je konvergentní.', 'Každá geometrická posloupnost je divergentní.', and 'Každá omezená geometrická posloupnost je divergentní.'. At the bottom left, there is a button labeled 'Konec kvizu'. At the bottom right, there is a button labeled 'Opravit' which is highlighted with a red border. At the very bottom, there are four navigation buttons: two double arrows pointing left and right, and two single arrows pointing left and right.

Obrázek 8: Druhá stránka test s ovládacími a vyhodnocovacími tlačítky

## 6 Webová stránka

Webová stránka slouží jako snadno ovladatelné rozhraní. Uživatel si sám určí, která témata ho zajímají. Vybere si počet otázek a obtížnost, pomocí PHP proměnných se mu pak zobrazí výsledek.

K vytvoření webových stránek jsem použil HTML 1.0, jehož základní značky (tagy) jsem psal do programu PSpad. Nejobtížnější v této fázi byl návrh grafické podoby. Nejdříve jsem si ručně navrhnul podobu stránky. Poté ji napsal v HTML a nakonec porovnal navržený koncept a se skutečnou podobou v prohlížeči. Uvědomil jsem si, že některé části se budou opakovat a je tedy zbytečné je v kódu stále opakovat. Rozdělil jsem tedy stránku pomocí rámců na tři části–Hlavička a Menu–budou obsahovat z pohledu uživatele jednotící prvky, část Obsah se bude měnit. Uživatel může vybrat ze stránek Úvod, Aplikace, Návod, Návštevni kniha a O Autorech–se mění v závislosti na přání uživatele.



Obrázek 9: Rozvržení webové stránky

### 6.0.1 Úvod

Obsahuje název Interaktivní testy pro matematickou analýzu I a jednoduchý motivační obrázek.

### 6.0.2 Aplikace

Tato stránka je nejdůležitější. Obsahuje rozhraní pro výběr počtu otázek a výběr náročnosti. Interaktivita je zajištěna pomocí navazujícího PHP skriptu.

### 6.0.3 Návod

Tato stránka ukazuje, jak se aplikace ovládá.

## 6.0.4 Návštěvní kniha

Na této stránce mohou uživatelé zanechat své vzkazy pro autory.

## 6.0.5 O Autorech

Poslední stránka představuje autory: Roberta Maříka, Petru Šarmanovou a Tomáše Majdu.

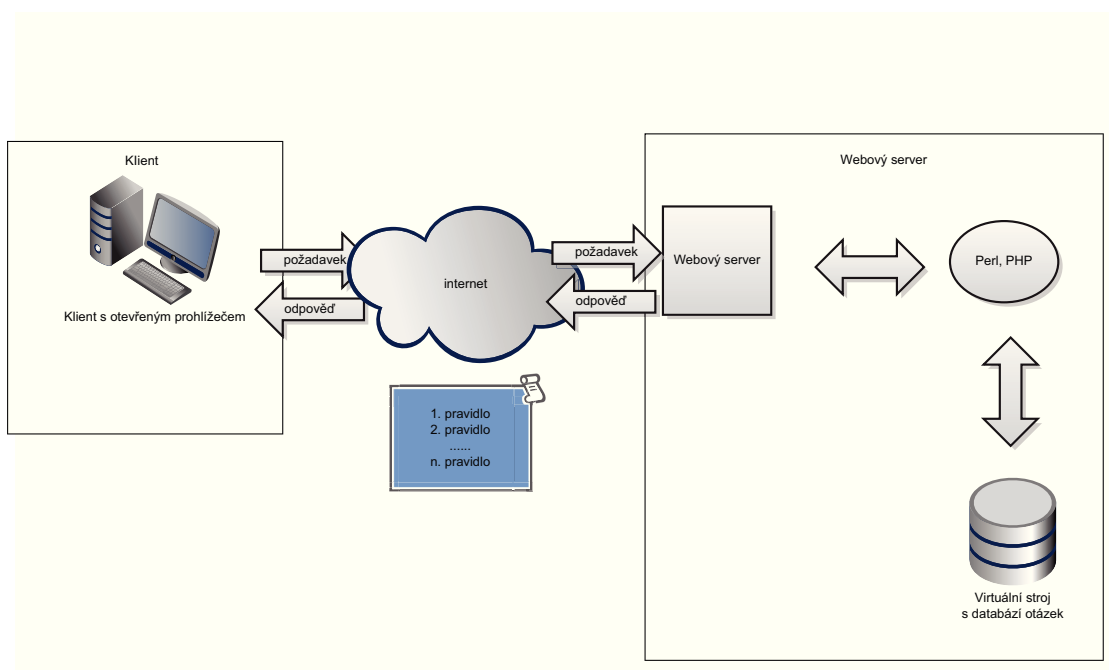
| Výber a označ témata                             | Výber a označ počet otázek daného tématu  | Výber a označ náročnost daného tématu  |
|--|---|--|
| Reálná čísla                                     | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Reálná funkce jedné reálné proměnné              | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Elementární funkce                               | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Posloupnosti                                     | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Limity   | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Derivace   | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Základní věty diferenciálního počtu              | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Průběh funkce, Lokální extrémy, Globální extrémy | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |
| Aproximace funkce polynomem                      | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 | <input type="radio"/> nenáročné <input type="radio"/> složité <input type="radio"/> velice náročné |

Obrázek 10: Úvodní stránka webové aplikace

## 7 Síťová část

Nyní bych mohl prezentovat vytvořený program v čistě offline podobě. Mohl bych implementovat kód pro operační systémy Windows, pro systémy s UNIXovým jádrem a pro úplnost bych mohl zahrnout i systémy používané v zařízeních firem Apple a Kindle. Vytvoření několika verzí pro různé systémy by bylo časově velice náročné. Přihlédnou-li i k faktu, že se tyto systémy neustále mění, může toto uvažování připomínat slavnou úlohu Achilles a želva.

Na druhou stranu všechna zařízení, od osobních počítačů a laptopů po PDA a chytré telefony, se všechna zařízení umí připojit k internetu a obsahují jako základní programové vybavení webový prohlížeč. Hlavně z tohoto důvodu jsem se rozhodnul zvolit cestu, kterou ukazuje obrázek.



Obrázek 11: Schéma komunikace

Pod pojmem klient budu rozumět uživatel, který je připojen pomocí prohlížeče k internetové adrese webové aplikace. I když je na obrázku znázorněn osobní počítač, uživatel může použít i laptop, chytrý mobilní telefon, nebo tablet.

Na druhé straně komunikačního schématu je vidět server. Právě do něj jsem umístil pomocí virtuálního stroje (virtual box) server typu LAMP.



Ke zjištění hardwarové vybavenosti použiju program putty. Po přihlášení k serveru a zadání příkazu `cat /proc/cpuinfo` vyčtu, že server řídí procesor firmy Intel(R), přesněji řečeno se jedná o Intel(R) Xeon(R) X5670 @2.93GHz.

### 7.0.6 Spouštění, aktualizace obsahu, sledování provozu serveru

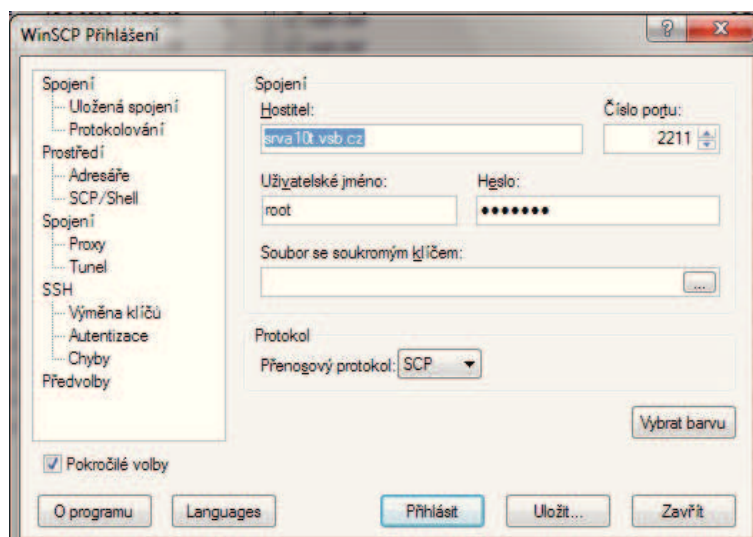
- **Spouštění**

Během nastarování serveru je zahájena činnost skriptu, který spustí program VirtualBox. Po přihlášení k virtualizovanému operačnímu systému Debian se zahájí činnost webového serveru Apache a poté je tento systém dostupný z adresy `srva10t.vsb.cz/acrotex/ramce.htm`.

- **Aktualizace testů**

je díky programu WinSCP velice snadná.

1. Stačí otevřít tento program.
2. Vybrat záložku spojení a zadat tyto údaje: Hostitel: `srva10t.vsb.cz`; port 2211; uživatel: `root`; přenosový protokol: SCP.
3. V prostředí podobném souborovým manažerům vybrat složku `/var/www` a zkopírovat aktualizací soubory do této složky.
4. Pro kontrolu můžeme načíst adresu aktualizace.



Obrázek 12: Správné vyplnění přihlašovacích údajů programu WinSCP

- **Testování počtu přístupu**

Další krokem je otestování počtu přístupu k serveru. Apache Benchmark (ab) je tím nejvhodnějším nástrojem ke zjištění, kolik požadavků za sekundu je schopen Apache obsloužit.

Tento program nalezneme v každé standardní distribuci systému Linux. Kompletní návod k jeho obsluze najdeme v online manuálu[6]. K ovládání programu používám obdobu příkazové řádky, program terminal.

Pomocí příkazu `ab -kc 30 -n 20 http://...` tak vytvořím dávku (parametr `-kc`) 30 spojení a každé generuje (parametr `-n`) 20 požadavků na adrese `http://...`

Následující tabulkou ukazujeme rychlost vykonávání dávky postupně pro 5, 10 a 15 otázek při 20 spojení a 20 požadavcích. V lichých sloupečcích vidíme procentuální vykonání instrukce, v sudých pak dobu vykonání.

|      |         |      |         |      |         |
|------|---------|------|---------|------|---------|
| 50%  | 2608 ms | 50%  | 3021 ms | 50%  | 2823 ms |
| 66%  | 3428 ms | 66%  | 3833 ms | 66%  | 3412 ms |
| 75%  | 3478 ms | 75%  | 3856 ms | 75%  | 3486 ms |
| 90%  | 3501 ms | 90%  | 3897 ms | 90%  | 3537 ms |
| 100% | 3512 ms | 100% | 3900 ms | 100% | 3538 ms |

Tabulka 2: Testování počtu přístupů

## 8 Závěr

Největším přínosem této práce je funkční webová aplikace. Naplnili jsme ji zhruba třemi stovkami otázek. Tuto aplikaci můžeme nabídnout studentům k procvičování svých znalostí z diferenciálního počtu funkce jedné proměnné.

V nastupující éře interaktivních tabulí v učebnách a tabletů v rukách studentů můžeme tento systém považovat za další možnost oživení výuky. Nemusíme dělat rozdíl, zda používáme stolní počítač, nebo tablet. Obě zařízení můžeme použít k procvičování, vybavíme-li je libovolným webovým prohlížečem a čtecím programem Adobe Reader 5.0, nebo vyšší verzí. Při práci s vygenerovaným dokumentem ani nemusíme být připojeni k internetu, protože veškeré funkce obsahuje PDF dokument.

Není vhodné používat tento systém jako formu zápočtové práce. Studenti si mohou sami test opravit a zobrazit správné odpovědi vestavěným tlačítkem. Proto je vhodnější pro distanční formu výuky, kdy se student poučuje ze svých chyb. Není také dobré testy vytisknout a používat otázky šablonovitě. Nejen, že se promíchávají otázky, ale i odpovědi na otázku.

Mezi nabyté znalosti považuju i naučení se práce s vektorovým editorem Inkscape a s typografickým systémem  $\text{\TeX}$ .

Při vytváření obrázků jsem se sice setkal s drobnými problémy, ale žádné nebyly neřešitelné. Vytvářel jsem nejen grafy funkcí, ale i další různorodé obrázky. Při zobrazování se ukázala výhoda vektorové grafiky v preciznosti vykreslení.

Tuto práci můžeme chápat i jako manuál k obdobným pracím. A to nejen pro výuku matematických kurzů, ale i všude tam, kde již existuje databáze otázek v textové formě.

Tomáš Majda

## 9 Reference

- [1] Bouchala, Kovář, Šarmanová, Vodstrčil *Otázky z matematické analýzy pro IT* Ostrava: VŠB-Technická univerzita Ostrava, 2007.
- [2] Story *The AcroT<sub>E</sub>XeDucation Bundle (AeB)* 2010. Dostupné z www: [http://www.math.uakron.edu/?dpstory/acrotex/aeb\\_man.pdf](http://www.math.uakron.edu/?dpstory/acrotex/aeb_man.pdf).
- [3] Robert Mařík *Acroweb collection* 2007. Dostupné z www: <http://ctan.mackichan.com/support/acroweb/acroweb.pdf>
- [4] Robert Mařík, Roman Plch, Petra Šarmanová *Tvorba interaktivních testů pomocí systému AcroT<sub>E</sub>X* 2010. Dostupné z www: <http://mi21.vsb.cz/tvorba-interaktivnich-testu-acrotex-1>
- [5] Jalová Nela *Testy z integrálního počtu funkcí více proměnných* [online], bakalářská práce MU Brno, 2008 [cit. 2010-04-12]. Dostupné z www: [http://is.muni.cz/th/175885/prif\\_b/bakalarka4.pdf?lang=en](http://is.muni.cz/th/175885/prif_b/bakalarka4.pdf?lang=en).
- [6] *Apache-manuál* www: <http://d.apache.org/docs/2.0/programs/ab.html> citováno 4. května 2012
- [7] Jan Pavlas *Interaktivní hry a testy pro výuku předmětu Matematická analýza I* Ostrava: VŠB-Technická univerzita Ostrava, 2010
- [8] Doc. RNDr. Jaromír Kuben, CSc., RNDr. Petra Šarmanová, Ph.D. *Interaktivní hry a testy pro výuku předmětu Matematická analýza I* Ostrava: VŠB-Technická univerzita Ostrava, 2006
- [9] HTML - Jak psát web, návod na html stránky [online], duben 2012. Dostupné z www: [www.jakpsatweb.cz/html/](http://www.jakpsatweb.cz/html/)
- [10] Jak se správně ptát [online], duben 2012. Dostupné z www: [www.hash.cz/inferno/otazky.html](http://www.hash.cz/inferno/otazky.html)
- [11] Mařík Robert-Home page [online], duben 2012. Dostupné z www: <http://user.mendelu.cz/?marik/>.

## A Příloha-Slovníček pojmů

### A.1 PHP

PHP („PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, či XHTML.

PHP je nejrozšířenějším skriptovacím jazykem pro web díky jednoduchosti použití a bohaté zásobě funkcí. Největší výhodou, kterou využívám i v této práci, je, že v kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP, Perl nebo Python.

### A.2 Perl

Perl je interpretovaný programovací jazyk vytvořený Larry Wallem v roce 1987. Perl umožňuje psát krátké programy jednoduše a rychle a přitom nebrání v psaní těch složitých. Perl je vhodný k řešení malých i velkých problémů. Schopnosti a nástroje, které se používají u velkých projektů, lze použít i v krátkých skriptech.

Já ho používám pro vytvoření skriptu typu CGI. Common Gateway Interface (zkráceně CGI) je protokol pro propojení externích aplikací s webovým serverem. To serveru umožňuje delegovat požadavek od klienta na externí aplikaci, která dle požadavku vrátí výstup. Taková aplikace typicky zpracuje nějaký skript ve webové stránce a webovému serveru navrátí statickou stránku, která je následně poslána klientovi jako výstup jeho požadavku.

### A.3 T<sub>E</sub>X, AcroT<sub>E</sub>X

Systém T<sub>E</sub>X slouží pro sazbu dokumentů ve vysoké typografické kvalitě, především pro svou preciznost v sazbě matematických a chemických vzorců. Tento systém, vyvinutý v 70. letech 20. století, je volně šiřitelný s možností vytváření dokumentů ve formátu PDF.

AcroT<sub>E</sub>X eDucation Bundle je kolekce maker, která umožňuje vytváření hypertextových testů, kvízů a otázek ve formátu PDF. Součástí balíčku jsou také prostředky pro vyhodnocení a opravu odpovědí.

Pro více detailů o AcroT<sub>E</sub>X u se můžete podívat na [urlhttp://www.acrotex.net](http://www.acrotex.net).

### A.4 AcroWEB

Balík AcroWEB je kolekcí PHP a Perl skriptů a L<sup>A</sup>T<sub>E</sub>Xšablon, které

- čtou čistý text z databáze matematických otázek, na které se může odpovědět výběrem z nabízených možností, nebo matematickým výrazem
- vybere předem vybraný počet otázek z předem napsané databáze
- vytvoří L<sup>A</sup>T<sub>E</sub>X soubor, který je kompilován pdfL<sup>A</sup>T<sub>E</sub>Xem a poslán na prohlížeč

## A.5 Klient

je v obecném významu aplikace, nebo systém, který přistupuje ke vzdálené službě umístěné na serveru. V této práci pod pojmem klient rozumíme http stránku a php skript.

## A.6 Server

je počítač, na kterém je realizována služba. V této práci to bude jednak hardware pracující server a jednak softwarové řešení– virtuální stroj umístěný v serverovně.

## A.7 Protokol

je způsob komunikace mezi dvěma počítači. Určují syntaxi a význam jednotlivých zpráv při komunikaci.

## A.8 Internet

je celosvětový systém navzájem propojených počítačových sítí , ve kterých mezi sebou počítače komunikují pomocí rodiny protokolů TCP/IP. Společným cílem všech lidí využívajících Internet je bezproblémová komunikace (výměna dat).

Nejnámější službou poskytovanou v rámci internetu je WWW (kombinace textu, grafiky a multimédií propojených hypertextovými odkazy) a e-mail (elektronická pošta), avšak nalezneme v něm i desítky dalších.

## A.9 WWW

World Wide Web (WWW, také pouze zkráceně web), je označení pro aplikace internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů.

Dokumenty umístěné na počítačových serverech jsou adresovány pomocí URL, jehož součástí je i doména a jméno počítače. Název naprosté většiny těchto serverů začíná zkratkou www, i když je možné používat libovolné jméno vyhovující pravidlům URL.

Autorem webu je Tim Berners-Lee, který jej vytvořil při svém působení v CERNu. Navrhl jazyk HTML a protokol HTTP, napsal první webový prohlížeč WorldWideWeb a koncem roku 1990 spustil první webový server na světě info.cern.ch. V říjnu roku 1994 založil World Wide Web Consortium (W3C), které dohlíží na další vývoj webu.