

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY



DIPLOMOVÁ PRÁCE

MATLAB Web Server aplikace
pro Polynomiální Toolbox



Praha, 2004

Richard Novák

Prohlášení

Prohlašuji, že jsem svou diplomovou (bakalářskou) práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne _____

podpis

Poděkování

Děkuji především vedoucímu diplomové práce Ing. Petru Urbanovi, Ing. Jaroslavu Pekaři a Ing. Jirkovi Roubalovi za cenné rady při řešení této práce. Dále bych rád poděkoval svým rodičům za podporu při studiu na této škole a všem lidem, kteří kolem sebe šíří radost a dobrou náladu.

Abstrakt

Tato práce se zabývá vytvářením internetové aplikace umožňující on-line řešení vybraných problémů s polynomy a polynomiálními maticemi, při použití Polynomiálního Toolboxu v MATLABu. Aplikace využívá možností MATLAB Web Serveru, jež je schopen přijímat data z internetových aplikací, a umožňuje práci s nimi v MATLABu. Po provedení požadovaných výpočtů následně zobrazí výsledky na straně uživatele. Aplikace se skládá ze tří částí. První částí jsou vstupní dokumenty. Zadávají se zde data, jež budou poslány MATLABu. Druhou částí je M soubor přijímající poslané údaje. V M souboru se provádí potřebné operace. Výsledky posílá výstupnímu dokumentu, který je poslední částí aplikace. Zobrazují se v něm výsledky z aplikace.

Abstract

This paper deals with the creating of an internet application, which enables on-line solution of some problems with polynomials and polynomial matrices using Polynomial Toolbox in MATLAB. The application makes use of MATLAB Web Server capabilities, which enables to receive data from internet applications and enables to work with them in MATLAB. After performing the desired computations, it displays results on the user's side. The application consists of three parts. The first part presents input documents. The data are entered here. Afterwards, they are sent to MATLAB. The second part is an M file, which receives the data sent. In the M-file, the necessary operations are performed. It sends the results to an output document, which is the third part of the application. It displays the results from the application.

Obsah

1	Úvod	1
2	Popis MATLAB Web Serveru	3
2.1	Součásti MATLAB Web Serveru	3
2.1.1	Soubor matlabserver.conf	5
2.1.2	Program matweb	5
2.1.3	Soubor matweb.m	5
2.1.4	Soubor matweb.conf	5
2.2	Funkce MATLAB Web Serveru	7
2.3	Vytváření aplikace	9
2.3.1	Vstupní dokument	9
2.3.2	M soubor	10
2.3.3	Výstupní soubor	11
2.3.4	Grafika	11
3	Popis aplikace	12
3.1	Vstupní dokument	12
3.1.1	Menu	13
3.1.2	Zadávání matice	15
3.1.3	Poslání dat	17
3.2	M soubor	17
3.3	Výstupní dokument	19
3.4	Adresářová struktura	19
4	Práce s aplikací	20
4.1	Zadávání matice	21
4.2	Polynomiální rovnice	22
4.3	Kontrola údajů	23

4.4	Příklady použití	25
5	Závěr	31
	Literatura	32

Seznam obrázků

1.1	Struktura aplikace	1
2.1	MATLAB na Webu	4
3.1	Rozdělení plochy aplikace	12
3.2	13
3.3	Zadání matice	16
3.4	Zadání matice	16
3.5	Adresářová struktura	19
4.1	Úvodní stránka aplikace	20
4.2	21
4.3	Zadávání matice bez Javascriptu	22
4.4	Polynomiální rovnice s využitím Javascriptu	23
4.5	24
4.6	Neodpovídající si rozměry matic	25
4.7	Determinant matice - zadání	25
4.8	Determinant matice - výsledek	26
4.9	Polynomiální rovnice - zadání	26
4.10	Polynomiální rovnice - výsledek	27
4.11	Redukovaný tvar matice - zadání	27
4.12	Redukovaný tvar matice - výsledek	28
4.13	Hodnota matice - zadání	28
4.14	Hodnota matice - výsledek	28
4.15	Inverze matice - zadání	29
4.16	Inverze matice - výsledek	29
4.17	Spektrální faktorizace matice - zadání	30
4.18	Spektrální faktorizace matice - výsledek	30

Seznam tabulek

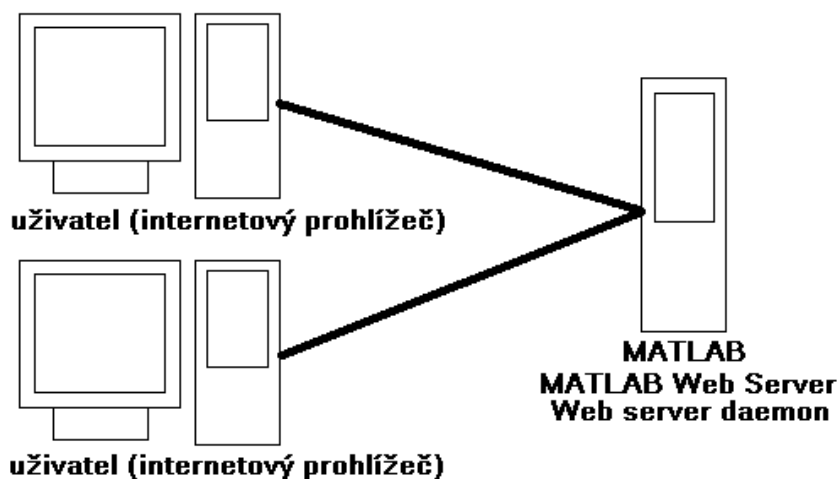
2.1	Základní parametry	5
2.2	Položky matweb.conf	6
2.3	Seznam funkcí	7
2.4	Detailní popis htmlrep	7
2.5	Detailní popis matweb	8
2.6	Detailní popis wscleanup	8
2.7	Detailní popis wsprintjpeg	8
2.8	Detailní popis wssetfield	8

Kapitola 1

Úvod

S rostoucím rozvojem internetu se klade stále větší důraz na prezentace výsledků a prací na něm, jakož i vytváření aplikací využívajících internet. Cílem práce je vytvoření internetové aplikace umožňující on-line řešení vybraných problémů s polynomy a polynomiálními maticemi, při využití Polynomiálního Toolboxu [1] v MATLABu [2]. Aplikace využívá možností MATLAB Web Serveru.

MATLAB Web Server [3] přijímá data z internetových aplikací a umožňuje práci s nimi v MATLABu. Výsledky posílá zpět internetové aplikaci, která výsledky zobrazí na straně uživatele. V nejjednodušší konfiguraci je na straně uživatele spuštěn internetový prohlížeč, zatímco na druhém počítači jsou spuštěny MATLAB, MATLAB Web Server a Web server daemon viz obr. 1.1.



Obrázek 1.1: Struktura aplikace

Polynomiální Toolbox je sada funkcí v MATLABu pro práci s polynomy a polynomiálními maticemi orientovaná na návrh řízení a zpracování signálu.

Úloha může najít uplatnění při demonstraci možností Polynomiálního Toolboxu nebo přímo při výuce ve škole.

Práce má následující strukturu. V kapitole 2 popíšeme strukturu MATLAB Web Serveru, jeho funkcí a způsob vytváření aplikací využívajících jej. V kapitole 3 popíšeme vytváření naší aplikace. V kapitole 4 jak pracovat z aplikací a příklady použití. V kapitole 5 zhodnotíme aplikaci.

Kapitola 2

Popis MATLAB Web Serveru

MATLAB Web Server umožňuje vytváření programů v MATLABu, spolupracujících s internetovými aplikacemi. Získané informace posílá MATLABu. MATLAB provede s daty požadované operace a výsledky zobrazí v internetovém prohlížeči.

2.1 Součásti MATLAB Web Serveru

MATLAB Web Server se skládá z několika součástí

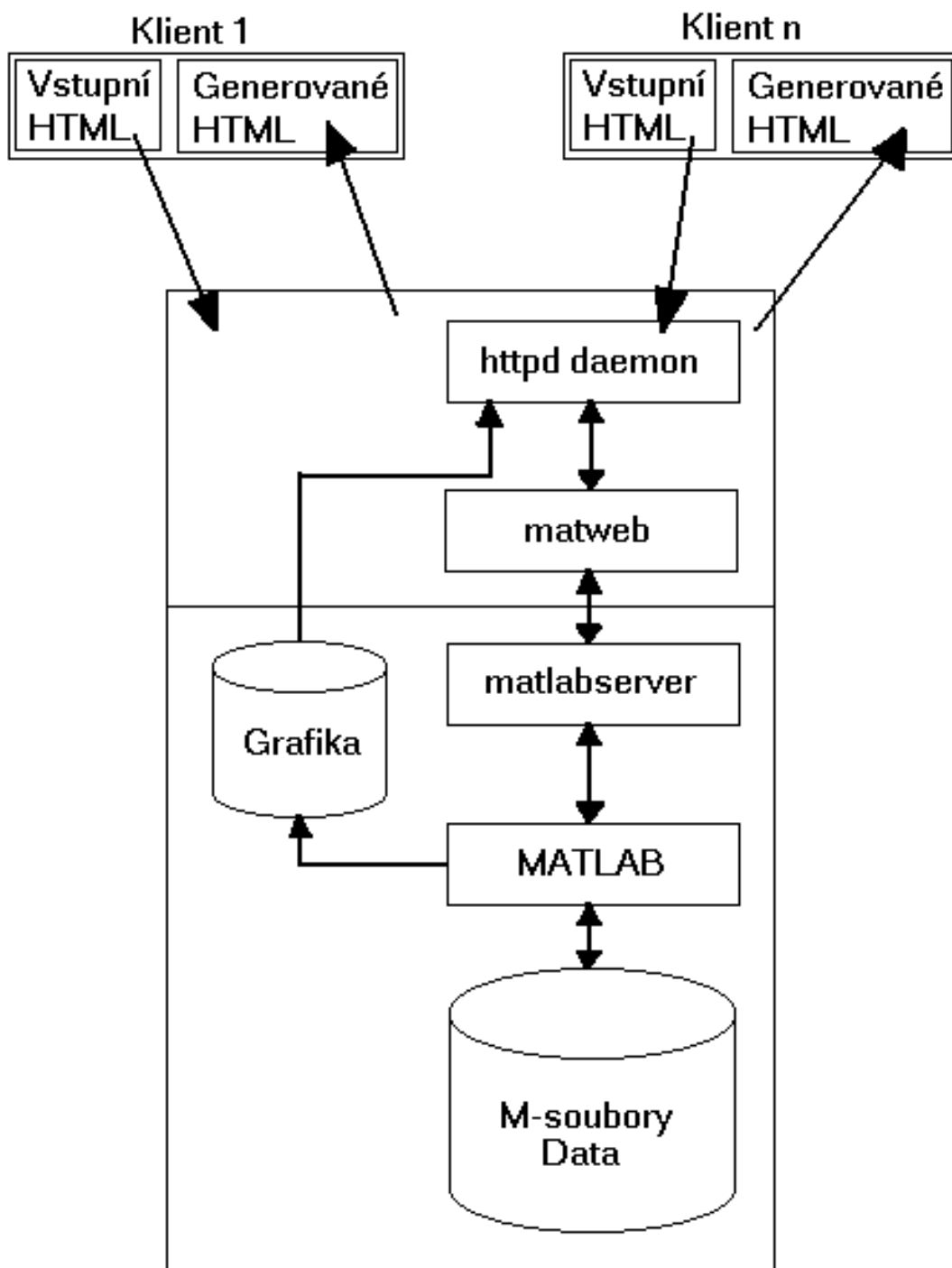
- **matlabserver**: program řídící komunikaci mezi internetovou aplikací a MATLABem.

matlabserver je vícevláknový TCP/IP server. Spouští programy (M soubor) v MATLABu specifikované ve skryté položce jménem **mlmfile** obsažené v HTML dokumentu. **matlabserver** spustí soubor **matweb.m**, což je požadovaný M soubor.

matlabserver může být nastaven na libovolný TCP/IP port nastavený v souboru **matlabserver.conf**. Rovněž zde můžeme nastavit počet současně spuštěných MATLABů.

- **matweb**: TCP/IP klient. Využívá Common Gateway Interface (CGI) k extrahování dat z HTML dokumentu a jejich přenosu programu **matlabserver**.
- **matweb.m**: volaný M soubor, který chce internetová aplikace spustit.
- **matweb.conf**: konfigurační soubor, jehož **matweb** potřebuje ke spojení s programem **matlabserver**. Aplikace musí být evidována v souboru **matweb.conf**.
- **hosts.conf**: pokud je tento soubor vytvořen, pouze počítače uvedené zde se mohou připojit k MATLAB Web Serveru.

Na obr. 2.1 je ukázáno fungování MATLABu s internetovou aplikací.



Obrázek 2.1: MATLAB na Webu

2.1.1 Soubor `matlabserver.conf`

Při spuštění programu `matlabserver` se nastaví jeho vlastnosti podle obsahu `matlabserver.conf`. Základní parametry nastavení, viz tab. 2.1

Tabulka 2.1: Základní parametry

parametr	význam	defaultní hodnota
<code>-p [n]</code>	číslo portu programu <code>matlabserver</code>	8888
<code>-m [n]</code>	počet současně spuštěných MATLABů	1

Mezi parametrem a jeho hodnotou je třeba udělat mezeru. Pokud program `matlabserver` nenalezne soubor `matlabserver.conf` použije defaultní nastavení. Napsáním `matlabserver -h` ve Windows dostaneme výpis všech parametrů editovatelných v `matlabserver.conf`. Soubor `matlabserver.conf` se nachází v adresáři, kde je nainstalován MATLAB Web Server.

2.1.2 Program `matweb`

`matweb` je TCP/IP klient používající Common Gateway Interface (CGI) k získávání dat z HTML formulářů. Přenáší informace programu `matlabserver`, který potom spouští aplikace napsané v M souborech. Pro přístup HTTP serveru musí být kopie programu `matweb` umístěna v adresáři ukazující na alias `/cgi-bin`.

2.1.3 Soubor `matweb.m`

Spouštěný soubor volaný internetovou aplikací. Do formuláře v HTML dokumentu se vloží skryté vstupní pole se jménem `mlmfile` a hodnotou s názvem M souboru, př.

```
<input type="hidden" name="mlmfile" value="my_det">
```

Uvedený příklad spustí soubor `my_det.m` po odeslání formuláře.

2.1.4 Soubor `matweb.conf`

Ke spojení s programem `matlabserver` potřebuje `matweb` informace uložené v souboru `matweb.conf`. Tento soubor musí být umístěn ve stejném adresáři jako `matweb`, tj. v adresáři ukazující na alias `/cgi-bin`.

Příklad `matweb.conf`

```
[my_det]
mlserver=eda1.sh.cvut.cz
mldir=d:/mws/example
```

```
[my_inv]
mlserver=eda1.sh.cvut.cz
mldir=d:/mws/example
```

Konfigurace všech aplikací musí být ve stejném souboru. Každá proměnná je na novém řádku. Ihned za ní následuje znaménko =, které proměnné přiřazuje hodnotu, např. `mlserver=eda1.sh.cvut.cz`. Jméno (M soubor) aplikace napsané v uzavřených závorkách [] je hlavním vstupním bodem pro aplikaci, např. za řádkem `[my_det]` následují všechny související proměnné pro aplikaci `my_det.m`. V tab. 2.2 jsou vypsány všechny položky, jež lze nastavovat v `matweb.conf`.

Tabulka 2.2: Položky `matweb.conf`

Proměnná	Význam
<code>[aplikace]</code> (nutná)	Jméno aplikace MATLABu.
<code>mldir</code> (optimální)	Pracovní adresář. Jestliže je specifikován, přidá se automaticky do cest v MATLABu.
<code>mllog</code> (optimální)	Vytváří log soubor, který obsahuje všechny výměny mezi aplikací a MATLABem. Je vhodné zrušit vytváření log souboru, protože má negativní dopad na výkon.
<code>mlserver</code> (nutná)	Jméno serveru, na kterém se spouští program <code>matlabserver</code> .
<code>mlport</code> (optimální)	Číslo portu pro <code>matlabserver</code> . Musí odpovídat hodnotě nastavené v souboru <code>matlabserver.conf</code> .
<code>mltimeout</code> (optimální)	Počet sekund čekání pro <code>matlabserver</code> před vypršením času.

Při vytvoření nové aplikace MATLAB Web Serveru a vložení nových konfiguračních dat do `matweb.conf` je potřeba `matlabserver` restartovat.

2.2 Funkce MATLAB Web Serveru

V této kapitole popíšeme funkce MATLAB Web Serveru. V tab. 2.3 máme funkce a jejich popis. Dále bude v následujících tabulkách 2.4, 2.5, 2.6, 2.7 a 2.8 proveden detailnější popis funkcí.

Tabulka 2.3: Seznam funkcí

Funkce	Význam
<code>htmlrep</code>	Nahradí jména proměnných v HTML dokumentu jejich hodnotami.
<code>matweb</code>	Hlavní vstupní bod MATLAB Web Serveru.
<code>wscleanup</code>	Smaže staré soubory z adresáře.
<code>wsprintjpeg</code>	Vytvoří JPEG soubor.
<code>wssetfield</code>	Přidá novou položku k již existující struktuře.

Tabulka 2.4: Detailní popis `htmlrep`

Význam	Jména proměnných v HTML dokumentu nahradí hodnotami
Syntaxe	<code>outstring = htmlrep(instruct, infile)</code> <code>outstring = htmlrep(instruct, infile, outfile)</code>
Popis	<code>outstring = htmlrep(instruct, infile)</code> nahradí všechny proměnné v souboru <code>infile</code> (HTML dokument) odpovídajícími hodnotami proměnných stejného jména jako ve struktuře <code>instruct</code> . Proměnné mohou být řetězce, matice nebo pole obsahující řetězce a skaláry. Řetězce a skaláry se nahrazují přímou substitucí. Jména proměnných v souboru <code>infile</code> musí být uzavřeny mezi znaky <code>\$ \$</code> , př. <code>\$varname\$</code> . Výstup se vrací v proměnné <code>outstring</code> . <code>outstring = htmlrep(instruct, infile, outfile)</code> navíc zapisuje výstup do HTML dokumentu <code>outfile</code> (vhodné pro testování).

Tabulka 2.5: Detailní popis matweb

Význam	Hlavní vstupní bod MATLAB Web Serveru
Syntaxe	<code>matweb(instruct)</code>
Popis	<p><code>matweb</code> je volaný M soubor uložený v položce <code>mlmfile</code></p> <p><code>instruct</code> obsahuje</p> <ul style="list-style-type: none"> - všechny data z HTML dokumentu - <code>mlmfile</code> obsahuje jméno M souboru, který se má spustit - <code>mldir</code> pracovní adresář specifikovaný v <code>matweb.conf</code> - <code>mlid</code> unikátní identifikátor pro vytváření jmen souborů a udržování souvislosti.

Tabulka 2.6: Detailní popis wscleanup

Význam	Maže staré soubory z adresáře
Syntaxe	<code>deletecount = wscleanup(filespec,timewindow,direc)</code>
Popis	<p>maže všechny soubory odpovídající <code>filespec</code> v adresáři <code>direc</code>, které jsou starší než počet hodin určený v <code>timewindow</code>.</p> <p><code>deletecount</code> je počet aktuálně smazaných souborů.</p>

Tabulka 2.7: Detailní popis wsprintjpeg

Význam	Vytváří JPEG soubor
Syntaxe	<code>status = wsprintjpeg(fig, jpegfilename)</code>
Popis	<p>Vytvoří JPEG soubor se jménem <code>jpegfilename</code>. <code>wsprintjpeg</code> se pokusí vytvořit JPEG soubor použitím příkazu MATLABu na tisk s parametrem <code>-djpeg</code>. Pokud se mu to nepodaří, vytvoří dočasný PCX soubor a potom zavolá funkce <code>imread</code> a <code>imwrite</code> k vytvoření JPEG výstupu.</p>

Tabulka 2.8: Detailní popis wsetfield

Význam	Přidá novou položku k již existující struktuře
Syntaxe	<code>s = wsetfield(s,name1,value1,...)</code>
Popis	<p>Nastaví obsah položky <code>name1</code> na hodnotu <code>value1</code> a vrátí výsledek do změněné struktury <code>s</code>.</p>

2.3 Vytváření aplikace

Vytváření aplikace probíhá v několika krocích. Vytvořením

- vstupního dokumentu a přidáním nových údajů do souboru `matweb.conf`
- M souboru
- výstupního dokumentu

Není nutné dodržet předepsané pořadí vytváření souborů.

2.3.1 Vstupní dokument

Vstupním dokumentem rozumíme HTML nebo XML.

V dokumentu nejprve vytvoříme formulář podle

```
<form action="/cgi-bin/matweb.exe" method="post">
```

Dále vytvoříme skryté vstupní pole s názvem `mlmfile` a hodnotou, obsahující jméno M souboru, který budeme chtít spustit po odeslání formuláře

```
<input type="hidden" name="mlmfile" value="muj_m_soubor">
```

Do formuláře pak můžeme přidávat další prvky, např. různé typy vstupního pole (tag `input`), výběrové pole (tag `select`), velké vstupní pole (tag `textarea`) apod. Důležité je, abychom u těchto dalších prvků uváděli atribut `name`. Potom budeme schopni přistupovat k jejich hodnotám i v našem M souboru.

Nakonec přidáme tlačítko k odeslání formuláře

```
<input type="submit" value="Submit">
```

Do souboru `matweb.conf` přepíšeme jméno našeho M souboru a nutné údaje. Minimálně to musí být jméno počítače, kde poběží program `matlabserver`. Ještě je vhodné zadat pracovní adresář, ve kterém je M soubor, př.

```
[muj_m_soubor]  
mlserver=eda1.sh.cvut.cz  
mldir=d:/mws/example
```

Soubor `matweb.conf` se spolu s programem `matweb` musí nacházet ve stejném adresáři ukazující na alias `/cgi-bin`.

2.3.2 M soubor

V M souboru můžeme používat normálně všechny příkazy MATLABu. Navíc je třeba použít několik příkazů, abychom byli schopni přijmout data ze vstupního souboru, a výsledky vrátit do výstupního dokumentu. Každý řádek musí být ukončen středníkem, tj. je potlačen veškerý výpis na obrazovku. Jinak by byl HTML výstup znehodnocen.

Na začátku M souboru napíšeme

```
function retstr = muj_m_soubor(instruct, outfile)
```

Inicializujeme návratový řetězec

```
retstr = char('');
```

Nastavíme pracovní adresář

```
cd(instruct.mldir);
```

Získáme proměnné ze vstupního souboru, které jsou uloženy ve struktuře `instruct`. Jména proměnných v této struktuře odpovídají atributům `name` zadaným ve vstupním dokumentu

```
moje_promenna = instruct.moje_promenna;
```

Protože ve struktuře `instruct` jsou položky uloženy ve formě řetězců, musíme řetězce převést na číselné hodnoty, pokud s nimi chceme dále počítat, např.

```
moje_promenna = str2num(instruct.moje_promenna);
```

Lze samozřejmě použít i jiné funkce pro převod, záleží co potřebujeme.

Dále provedeme potřebné operace.

Vložíme proměnné, které chceme vypsát ve výstupním dokumentu do výstupní struktury

```
oustruct.moje_promenna = moje_promenna;
```

Nakonec zavoláme funkci `htmlrep`, ale ještě před tím uložíme do proměnné `outfile` jméno výstupního HTML dokumentu pomocí funkce `which`

```
outfile = which('det_out.html');  
retstr = htmlrep(outstruct,outfile);
```

2.3.3 Výstupní soubor

Proměnné, které chceme vypsát, uzavřeme mezi znaky \$ \$, př.

```
Determinant matice je $moje_promenna$
```

Lze rovněž dynamicky vytvářet HTML tabulky a výběrová pole z matic nebo polí prvků obsahující řetězce a skaláry. Používá se k tomu atribut `autogenerate`, př.

```
<table border="1" cellspacing="1" autogenerate="$matrix$"
  <tr>
    <td align="right">
    </td>
  </tr>
</table>
```

```
<select name="namelist" SIZE="1" autogenerate=$mylist$
  <option>
  </option>
</select>
```

2.3.4 Grafika

Pokud chceme uložit nějaký soubor a poté jej použít ve výstupním dokumentu, např. obrázek, je vhodné k pojmenování tohoto souboru využít položku `mlid` ze struktury `instruct`. Nejprve získáme položku `mlid` pomocí funkce `getfield` a potom jí použijeme k vytvoření názvu souboru

```
mlid = getfield(instruct,'mlid');
outscruct.name_file = sprintf('%spicture.jpeg',mlid);
```

Vlastní obrázek vytvoříme funkcí `wsprintjpeg`.

Kapitola 3

Popis aplikace

V předchozí kapitole jsme probrali obecný postup vytváření aplikace MATLAB Web Serveru. Zde popíšeme konkrétní způsob vytváření naší aplikace.

3.1 Vstupní dokument

Stránky jsou napsány v XHTML []. Dále jsou využívány CSS styly a Javascript. Byla zachována základní funkčnost i při vypnutém Javascriptu.

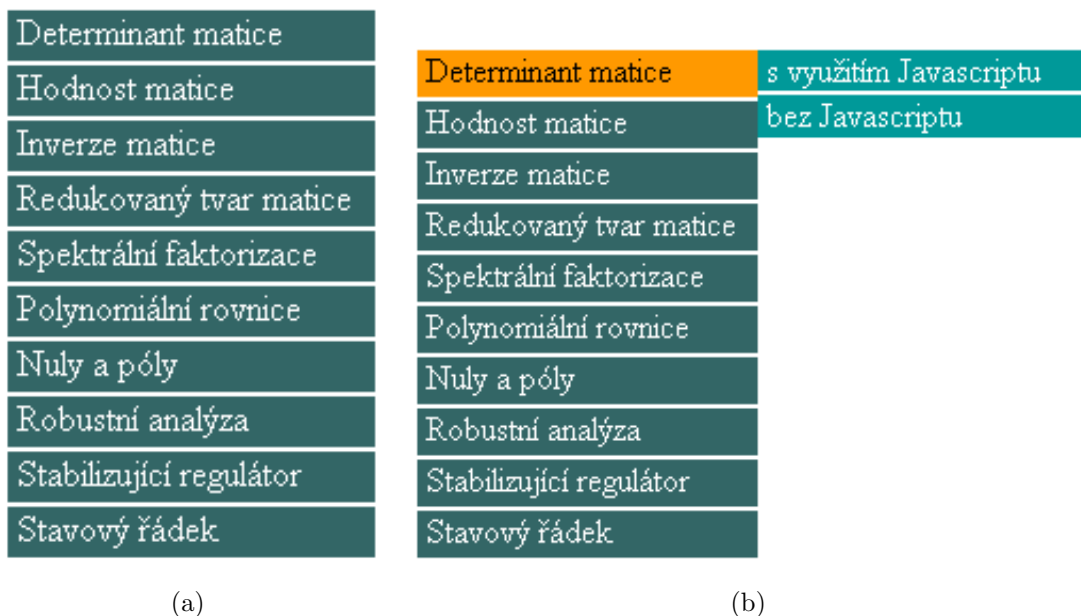
Na obr. 3.1 vidíme rozdělení plochy aplikace. První oblastí je horní pruh táhnoucí se přes celou šířku plochy (na obr. 3.1 oblast č.1). Obsahuje odkaz na úvodní stránku a může zobrazovat další důležité informace. Další oblastí je pruh po levé straně (na obr. 3.1 oblast č.2), který obsahuje nabídku možností (menu). Zbytek plochy je poslední oblast (na obr. 3.1 oblast č.3). Zadávají se zde údaje pro jednotlivé nabídky a posílají se MATLABu. Tato část slouží i k zobrazování výsledků.



Obrázek 3.1: Rozdělení plochy aplikace

3.1.1 Menu

Na obr. 3.2(a) je základní nabídka. Po najetí na vybranou položku se zobrazí nabídka pro tuto položku, viz obr. 3.2(b).



Obrázek 3.2: Menu: (a) Základní nabídka; (b) Podnabídka

K vytvoření nabídky je použito CSS stylů a XHTML. Ve všech internetových prohlížečích bude fungovat i s vypnutým Javascriptem. Výjimku tvoří Internet Explorer. Zde totiž funguje pseudotřída `:hover` pouze pro odkazy (tag `a`). U jiných prohlížečů funguje tato pseudotřída pro kterýkoli prvek, např v Mozille. U Internet Exploreru si lze pomoci CSS vlastností `behavior`, která určuje vazbu skriptů na události prvku.

Do zápisu stylu uvedeme adresu na externí soubor s příponou `htc`

```
selektor {behavior: url('soubor.htc')}
```

Soubory s koncovkou `htc` používá pouze Internet Explorer. Mozilla využívá soubory s koncovkou `xbl`. Navíc se syntaxe souborů XBL a HTC trochu liší. V našem případě je to výhoda, protože potřebujeme vytvořit náhradu pseudotřídy `:hover` i pro jiné elementy než je odkaz (tag `a`) a to jen v Internet Exploreru. Uvedeme příklad našeho HTC souboru

```
<component>
  <attach event="onmouseover" onevent="hov-on()">
  <attach event="onmouseout" onevent="hov-off()">
```

```
<script>
  function hov-on()
  {
    element.style.overflow="visible";
  }
  function hov-off()
  {
    element.style.overflow="hidden";
  }
</script>
</compoment>
```

Při události `onmouseover` nastavíme CSS vlastnost elementu `overflow` na hodnotu `visible`. Událost `onmouseover` nastává, když myší najedeme na prvek. Při události `onmouseout` nastavíme CSS vlastnost elementu `overflow` na `hidden`. Když kurzorem myši odjedeme z prvku, nastává událost `onmouseout`.

Protože se již jedná o použití Javascriptu, nebude v Internet Exploreru menu fungovat, pokud bude zakázáno používání Javascriptu. Pro tuto možnost je připraven jiný CSS styl, kdy budou podnabídky zobrazeny stále a ne jen při pohybu kurzoru myši nad položkou menu.

Základem menu je odrážkový seznam (tag `ul`). V jeho položkách (tag `li`) se nachází další odrážkové seznamy, v jehož položkách už jsou odkazy, př.

```
<ul id="menu">
  <li class="menuitem" id="mi01">
    <div class="item">Determinant matice</div>
    <ul class="submenu">
      <li><a href="det_in_js.html">s využitím Javascriptu</a></li>
      <li><a href="det_in_nojs.html">bez Javascriptu</a></li>
    </ul>
  </li>
  <li class="menuitem" id="mi02">
    <div class="item">Hodnost matice</div>
    <ul class="submenu">
      <li><a href="rank_in_js.html">s využitím Javascriptu</a></li>
      <li><a href="rank_in_nojs.html">bez Javascriptu</a></li>
    </ul>
  </li>
</ul>
```

```
</li>  
</ul>
```

Vhodným nastavením CSS vlastností dosáhneme požadované funkčnosti menu. První úroveň `li` (třída `menuitem`) jsou vždy viditelné absolutně pozicované bloky. Druhá úroveň `ul` (třída `submenu`) je už pozicovaná relativně. Pro skrytí nebo zobrazení využíváme CSS vlastnosti `overflow`, která určuje, jak se bude zacházet z obsahem, jež vyteče z rozměru prvku. Prvek však musí mít nastavené rozměry, tedy alespoň `width` nebo `height`. Pro hodnotu `visible` může dojít k přetékání, pro hodnotu `hidden` nikoliv, tj. nebude zobrazeno, co se nevejde.

Vlastnost `overflow` budeme nastavovat pro třídu `menuitem`. Na počátku je hodnota `overflow hidden`. Šířka této třídy je nastavena na maximum. Protože třídu `submenu` posouváme až za maximální šířku, nebude zobrazena. Pomocí pseudotřídy `:hover` nastavíme `overflow` pro třídu `menuitem` na `visible` při přejetí kurzoru nad požadovanou položkou.

3.1.2 Zadávání matice

Při zadávání matice máme dvě možnosti. První je zadání matice stejně jako v MATLABu. V tomto případě není třeba používat Javascript. V druhém případě se matice zadává po jednotlivých prvcích, kdy se na počátku zadá rozměr matice. Zde už používáme Javascript. Na obr. 3.3 vidíme příklad zadávání.

V některých případech má řešení smysl pouze pro čtvercové matice. V tomto případě je možno editovat jeden rozměr, druhý se nastavuje podle prvního.

Každý prvek matice je tvořen jedním vstupním polem (tag `input`). K vytváření matice používáme Document Object model (DOM [4]). Jedná se o objektovou reprezentaci stránky. DOM API umožňuje pomocí vlastností a metod pracovat s celým obsahem dokumentu.

Determinant matice

Rozměr matice x

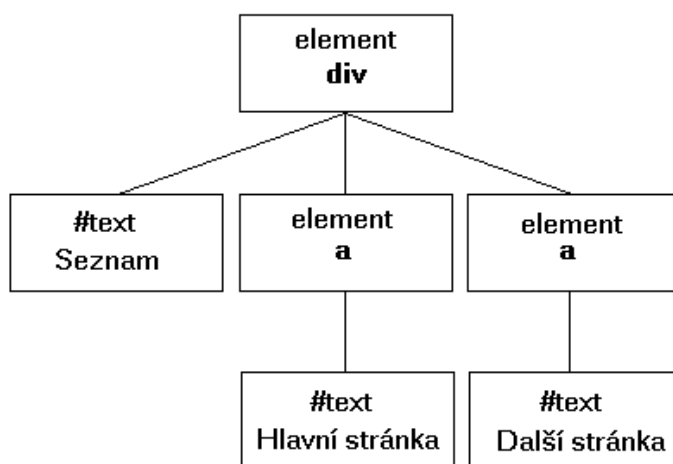
Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Obrázek 3.3: Zadání matice

Prohlížeč, který DOM podporuje, při načítání stránky vytváří stromovou hierarchii prvků, podle pořadí zápisu jednotlivých elementů. Na obr. 3.4 je grafická reprezentace následujícího příkladu

```
<div id="top">
  Seznam
  <a href="index.html">Hlavní stránka</a>
  <a href="next.html">Další stránka</a>
</div>
```



Obrázek 3.4: Zadání matice

Každý element je z hlediska DOM reprezentován uzlem (node). Ve specifikaci DOM existuje k práci s jednotlivými uzly tzv. uzlové rozhraní, což je soubor vlastností patřící danému uzlu.

Při zadávání matice máme na výběr, zda ukládat hodnoty předchozí matice, či nikoli. Uložené hodnoty se pak zpětně obnoví po vytvoření nové matice. Toto je vhodné třeba pokud už máme zadanou matici a chceme jí jenom změnit rozměry. Nemusíme pak znovu zdlouhavě vyplňovat jednotlivé prvky matice. Po zadání nových rozměrů matice, provedeme smazání celé matice, tj. všech vstupní polí. Ještě před smazáním provedeme uložení hodnot předchozí matice, pokud si to uživatel zvolil. Poté vytvoříme novou matici podle zadaných rozměrů a buď nahrajeme původní hodnoty nebo bude každý prvek nulový.

3.1.3 Poslání dat

Ještě před posláním údajů MATLABu provedeme alespoň základní kontrolu zadaných dat. Při kliknutí na odesílací tlačítko spustíme skript, který provede kontrolu. Pokud dojde k nějaké chybě vypíšeme chybovou hlášku pomocí funkce `alert`. Informujeme o typu chyby, a na jakém místě k ní došlo. Kontrolní funkce vrátí hodnotu `false`, což znamená, že nedojde k odeslání formuláře.

Kontrolujeme, zda jsou zadány všechny prvky matice, jestli nějaký prvek neobsahuje nepovolené znaky.

3.2 M soubor

Zde je třeba dát pozor nato, aby každý řádek M souboru byl ukončen středníkem, a aby žádná použité funkce nic nevypisovala na obrazovku. Jinak by totiž výstup nebyl korektní.

Funkce `pinit`, která inicializuje Polynomiální Toolbox, při spuštění vypisuje informaci

```
Polynomial Toolbox 3.0 initialized. To get started, type one of
these: helpwin or poldesk. For product information, visit
www.polyx.com or www.polyx.cz.
```

Musíme tedy upravit tuto funkci. Zakomentujeme řádky, kde dochází k výpisu informace.

M soubory pro vstupní dokumenty bez nebo s Javascriptem se liší pouze ve zpracování přijatých dat. Při nevyužití Javascriptu jsou matice zadány ve velkém

vstupním poli (tag `textarea`) v celku. K získání jejich hodnoty postačí prosté přiřazení a funkce `eval`, která provede řetězec jako výraz nebo výrok

```
P = eval(h.textmatrix);
```

Při využití Javascriptu musíme matici rozdělenou na jednotlivé prvky zase sloučit do jednoho celku

```
for x=0:m_row-1,
    for y=0:m_col-1,
        P(x+1,y+1)=str2num(eval(sprintf('h.p_cell%d%d',x,y)));
    end
end
```

Každý prvek matice má svoje jméno. Funkcí `sprintf` vytvořím jméno prvku. Funkcí `eval` říkáme, že chceme hodnotu proměnné, kterou převedeme z řetězce na číslo pomocí `str2num`. Další postup je pak společný pro oba případy.

Využíváme konstrukci `try ... catch ... end`, která slouží k zachytávání vyjímek, k nimž může dojít při provádění kódu mezi `try` a `catch`. Následuje zkrácený výpis z jednoho M souboru

```
function HTMLout=my_det_nojs(h)
pinit;
try
    P=eval(h.textmatrix);
    detP=det(P);
    sr.matrix=rchar(P);
    sr.result=rchar(detP);
    outfiletemp=which('det_out.html');
catch
    n=lasterr;
    sr.result=n;
    outfiletemp=which('out_err.html');
end
HTMLout=htmlrep(sr,outfiletemp);
```

Funkce `rchar` převádí polynomiální objekt na řetězec. Pokud dojde k chybě mezi `try` a `catch`, vykoná se oblast kódu mezi `catch` a `end`. V našem případě si zjistíme poslední chybu pomocí `lasterr` a vypíšeme jí do výstupního dokumentu, který se zobrazí uživateli.

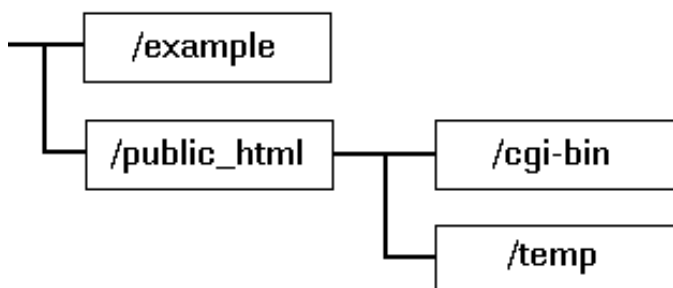
3.3 Výstupní dokument

Výstupní dokument je při zobrazování jakoby spouštěn ze stejného adresáře jako program `matweb`, tj. z adresáře `/cgi-bin`. Tuto skutečnost je třeba brát na zřetel při vytváření odkazů nebo zadávání cest v dokumentu.

K zobrazování výsledků z MATLABu využíváme dva způsoby. Jeden využívá konverzi polynomiálního objektu na řetězec pomocí funkce `rchar`. V dokumentu je pak zobrazován v tabulce, která má nastavený atribut `autogenerate` na jméno proměnné uzavřené mezi znaky `$ $`. Počet sloupců a řádků tabulky se nastaví automaticky podle rozměrů řetězce. Druhý způsob využívá konverzi polynomiálního objektu do jazyka MathML pomocí funkce `pol2mml`.

3.4 Adresářová struktura

Na obr. 3.5 vidíme adresářovou strukturu aplikace. Adresář `example` obsahuje M soubory a výstupní dokumenty, které slouží k zobrazení výsledků. V adresáři `public_html` se nachází vstupní dokumenty. Adresář `cgi-bin` obsahuje program `matweb` a soubor `matweb.conf`. Do adresáře `temp` se ukládají dočasné soubory vytvořené v MATLABu, jako např. obrázky apod.

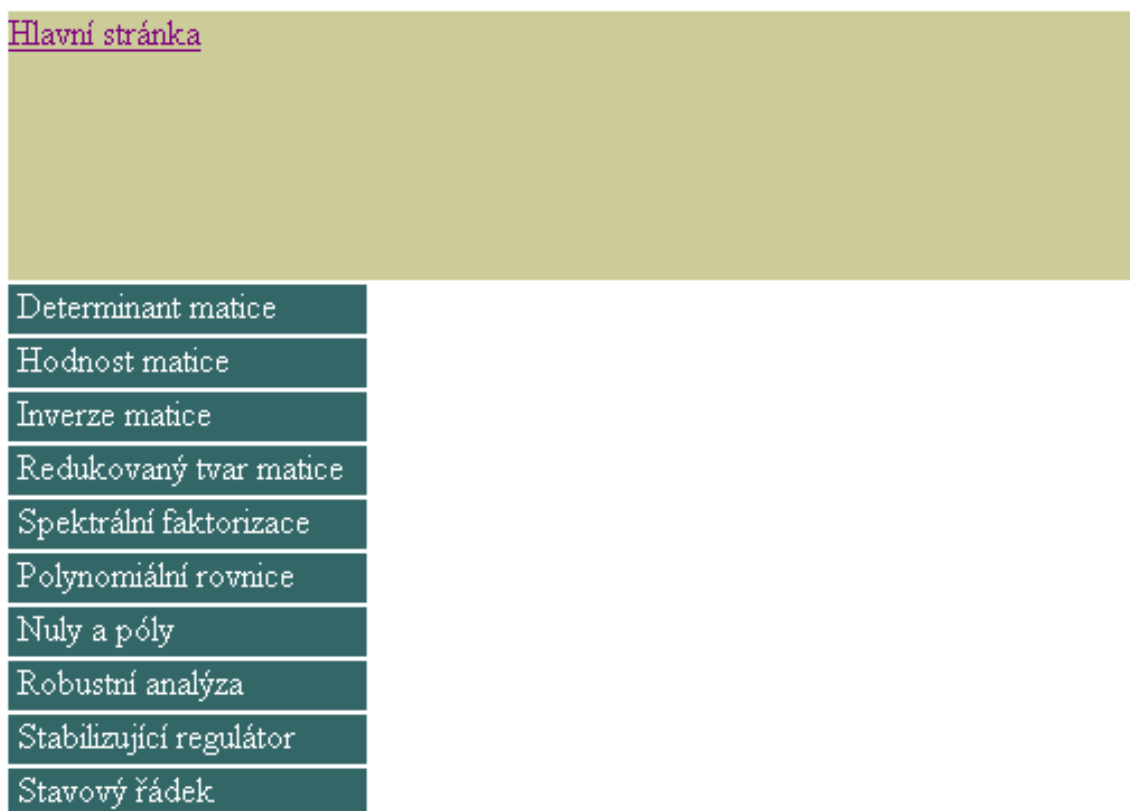


Obrázek 3.5: Adresářová struktura

Kapitola 4

Práce s aplikací

V této kapitole popíšeme jak pracovat s aplikací. Na obr. 4.1 vidíme úvodní obrazovku.



Obrázek 4.1: Úvodní stránka aplikace

4.1 Zadávání matice

Při zadávání matice máme dvě možnosti. Využít Javascript, viz obr. 4.2(a). Matici budeme zadávat po jednotlivých prvcích.

Determinant matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="3*s^2+s+5"/>	<input type="text" value="s-3"/>
<input type="text" value="32*s^2"/>	<input type="text" value="14*s+4"/>

(a)

Determinant matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="3*s^2+s+5"/>	<input type="text" value="s-3"/>	<input type="text" value="0"/>
<input type="text" value="32*s^2"/>	<input type="text" value="14*s+4"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

(b)

Determinant matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="3*s^2+s+5"/>
--

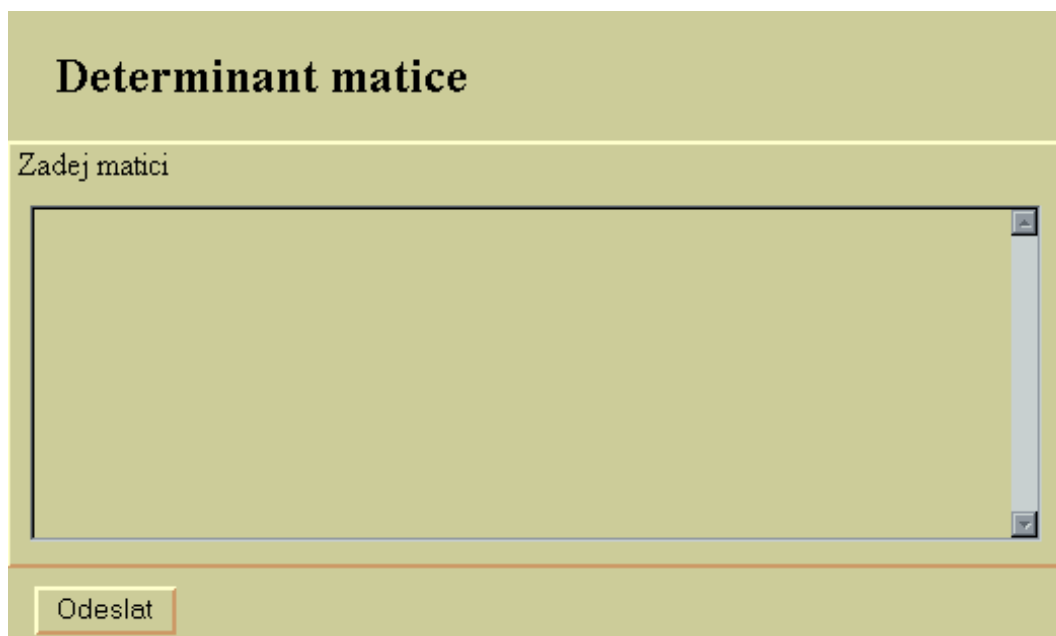
(c)

Obrázek 4.2: Zadávání matice: (a) Původní matice; (b) Větší matice; (c) Menší matice

Nejdříve zadáme požadovaný rozměr matice a vybereme si, zda budeme chtít ponechat hodnoty předchozí matice nebo ne. Stisknutím tlačítka OK dojde k vytvoření matice. Hodnoty každého prvku budou nulové v případě, že jsme nechtěli ponechat hodnoty původní matice. V případě výběru ponechání původních hodnot zůstanou odpovídající prvky matice nastavené na předchozí hodnotu. Pokud má nová matice větší rozměry, zbytek se doplní nulami, viz obr. 4.2(b). Při menších rozměrech nové matice provedeme vlastně oříznutí, viz obr. 4.2(c).

V případě, že by zvolená nabídka měla smysl pouze pro čtvercové matice, povolujeme editovat jeden rozměr matice a druhý nastavujeme podle prvního.

Pro možnost bez Javascriptu zadáváme matici jako celek do velkého vstupního pole, viz obr. 4.3. Matici zadáváme stejně jako v MATLABu. Celou matici uzavřeme do hranatých závorek. Sloupce oddělujeme mezerou nebo čárkou. Nový řádek následuje za středníkem. Do textového pole píšeme samotnou matici bez přiřazení některé proměnné.



The image shows a web interface for calculating a determinant. The title is "Determinant matice". Below the title is a text input field with the label "Zadej matici". At the bottom of the interface is a button labeled "Odeslat".

Obrázek 4.3: Zadávání matice bez Javascriptu

4.2 Polynomiální rovnice

Při nevyužití Javascriptu vidíme na stránce všechny tři matice A, B a C. S využitím Javascriptu ukrýváme matici C podle vybraného typu rovnice, viz obr. 4.4. Rovněž podle typu rovnice nastavujeme možnosti výběru dalšího parametru, který

určuje, zda budeme hledat výsledek s minimálním stupněm matice X , Y nebo nepoužijeme ani jednu z těchto nabídek.

Polynomiální rovnice

Rozměr matice A x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

Rozměr matice B x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

Výběr rovnice (X, Y neznámé): Další parametr:

- AX=B
- AXB=C
- AX+BY=C
- A'X+X'A=B
- A'X+Y'A=B
- AX+YB=C
- X'A+AX'=B
- XA=B
- XA+YB=C

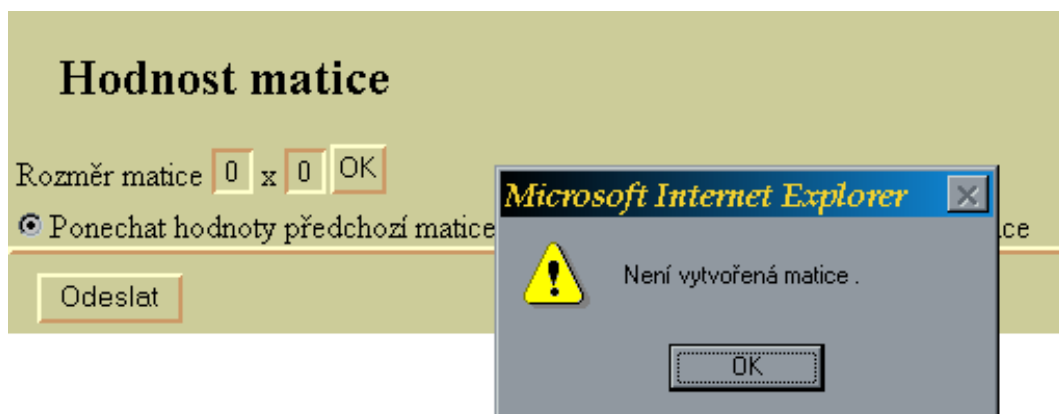
Obrázek 4.4: Polynomiální rovnice s využitím Javascriptu

4.3 Kontrola údajů

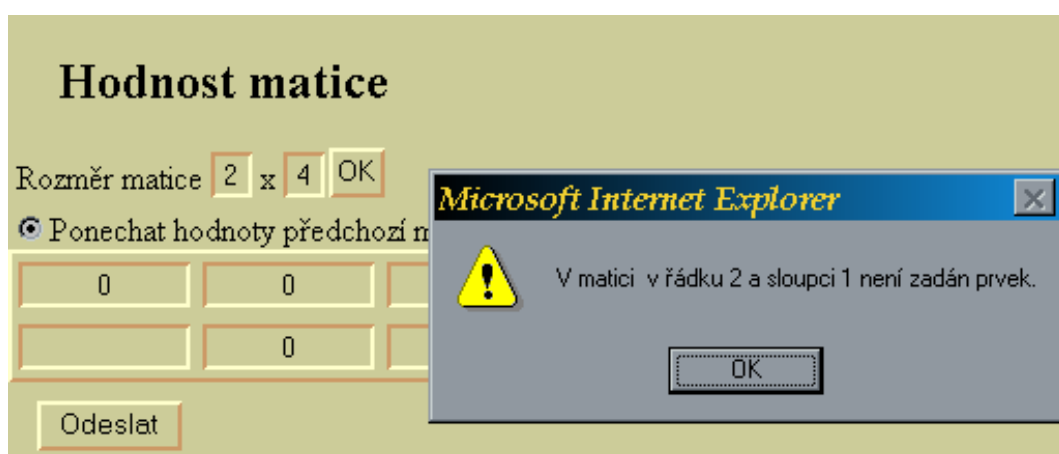
V případě nevyužití Javascriptu žádnou kontrolu neprovádíme. Pomocí Javascriptu kontrolujeme zadané hodnoty těsně před odesláním formuláře MATLABu. Když dojde k nějaké chybě, nedovolíme odeslat formulář.

U každé matice zjišťujeme, zda je vůbec zadána, obr. 4.5(a). Jestli jsou vyplněné všechny prvky matice, obr. 4.5(b), a zda-li nějaký prvek neobsahuje nepovolený znak, obr. 4.5(c). V chybové hlášce popíšeme chybu, i na jakém místě v matici nastala.

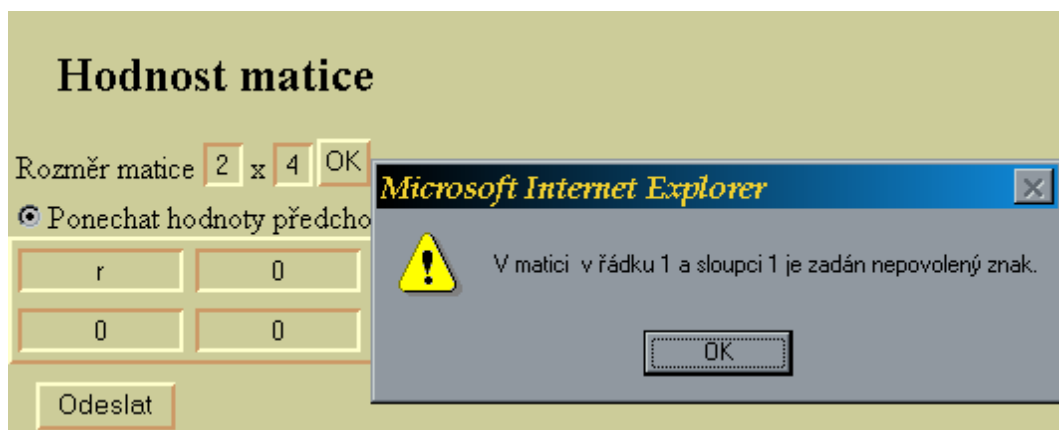
U polynomiálních rovnic provádíme kontrolu rozměrů matic, aby si odpovídaly počty řádků i sloupců na pravé i levé straně rovnice, obr. 4.6.



(a)



(b)



(c)

Obrázek 4.5: Kontrola matice: (a) Není zadaná matice; (b) Není zadán prvek matice; (c) Nepovolený znak v prvku matice

Polynomiální rovnice

Rozměr matice A x

Ponechat hodnoty předchozí matice

<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>


Rozměr matice B x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="0"/>

Výběr rovnice (X, Y neznámé): Další parametr:

Microsoft Internet Explorer

 Neodpovídá si počet řádků matice A a B.

Obrázek 4.6: Neodpovídající si rozměry matic

4.4 Příklady použití

Na následujících obrázcích jsou příklady použití a zobrazené výsledky.

Determinant matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

<input type="text" value="1"/>	<input type="text" value="s"/>	<input type="text" value="s^2"/>
<input type="text" value="1+s"/>	<input type="text" value="s"/>	<input type="text" value="1-s"/>
<input type="text" value="0"/>	<input type="text" value="-1"/>	<input type="text" value="-s"/>

Obrázek 4.7: Determinant matice - zadání

Determinant matice

1	s	s ²
s + 1	s	-s + 1
0	-1	-s

Výsledek je

-s ² - s + 1

Obrázek 4.8: Determinant matice - výsledek

Polynomiální rovnice

Rozměr matice A x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

1	s	1+s
s-1	1	0

Rozměr matice B x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

s	0
0	1

Výběr rovnice (X, Y neznámé): Další parametr:

Obrázek 4.9: Polynomiální rovnice - zadání

Řešení rovnice $AX=B$, kde A je

1	s	s + 1
s - 1	1	0

B je

s	0
0	1

Výsledek je

1	-1
-s + 1	s
s - 1	-s + 1

Obrázek 4.10: Polynomiální rovnice - výsledek

Redukovaný tvar matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

$1+s^2$	-2
s-1	1

Řádkový redukovaný tvar matice

Sloupcový redukovaný tvar matice

Obrázek 4.11: Redukovaný tvar matice - zadání

Zadaná matice

$s^2 + 1$	-2
$s - 1$	1

Řádkově redukovaný tvar matice je

$s + 1$	$-s - 2$
$s - 1$	1

Obrázek 4.12: Redukovaný tvar matice - výsledek

Hodnost matice

Zadej matici

$[s^2+s \ s \ s^3-1; 2 \ s+3 \ s; s-4 \ s^2+5 \ s]$

Odeslat

Obrázek 4.13: Hodnost matice - zadání

Hodnost matice

$s^2 + s$	s	$s^3 - 1$
2	$s + 3$	s
$s - 4$	$s^2 + 5$	s

Výsledek je 3

Obrázek 4.14: Hodnost matice - výsledek

Inverze matice

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

1	s	s ²
1+s	s	1-s
0	-1	-s

Obrázek 4.15: Inverze matice - zadání

Inverze matice P

1	s	s ²
s + 1	s	-s + 1
0	-1	-s

Výsledek je $(1/\det P)\text{adj } P$, kde $\det P$ (determinant matice)

adj P (adjugovaná matice)

-s ² - s + 1	0	-s ³ - s ² + s
s ² + s	-s	s ³ + s ² + s - 1
-s - 1	1	-s ²

Obrázek 4.16: Inverze matice - výsledek

Spektrální faktorizace

Rozměr matice x

Ponechat hodnoty předchozí matice Neponechat hodnoty předchozí matice

$34-56*s^2$	$-13-22*s+60*s^2$	$36+67*s$
$-13+22*s+60*s^2$	$46-1e+002*s^2$	$-42-26*s+38*s^2$
$36-67*s$	$-42+26*s+38*s^2$	$59-42*s^2$

Obrázek 4.17: Spektrální faktorizace matice - zadání

Spektrální faktorizace matice

$-56s^2 + 34$	$60s^2 - 22s - 13$	$67s + 36$
$60s^2 + 22s - 13$	$-100s^2 + 46$	$38s^2 - 26s - 42$
$-67s + 36$	$38s^2 + 26s - 42$	$-42s^2 + 59$

Výsledek je

$-0.41659s - 2.0563$	$-0.38571s - 5.2052$	$-0.35273s + 1.9782$
$-4.0297s + 5.4539$	$-0.64075s - 4.3368$	$5.4599s + 7.3723$
$-6.2919s - 0.16308$	$9.972s + 0.31343$	$-3.4735s + 0.85798$

Obrázek 4.18: Spektrální faktorizace matice - výsledek

Kapitola 5

Závěr

Spojením jednotlivých samostatných částí (WWW stránek a MATLAB Web Serveru) byla vytvořena fungující internetová aplikace umožňující on-line řešení vybraných problémů s polynomy a polynomiálními maticemi (výpočet hodnoti, determinantu, inverze a redukovaného tvaru polynomiální matice, řešení diofantických rovnic, výpočet spektrální faktorizace, a další).

Byla snaha, aby aplikace vypadala ve všech prohlížečích stejně. Ne vždy se to povedlo. Je to způsobeno odlišnou implementací standardů W3C. Např. u pozicování bloků dochází v Internet Exploreru k chybné interpretaci vlastnosti `width`. Do této vlastnosti navíc připočítává vnitřní okraj (`padding`) a rámeček (`border`) bloku. Na funkčnost aplikace nemá toto žádný vliv. Pouze v podnabídce v menu nejsou jednotlivé položky u Mozilly odděleny čarou.

Pro odladování funkcí napsaných v Javascriptu je vhodná Mozilla, protože vypisuje místo vzniku chyby v kódu, i k jaké chybě došlo. Zapnutý Javascript v prohlížeči není podmínkou fungování aplikace. Zadávání matic řešíme dvěma způsoby. První způsob, zadávání matice po jednotlivých prvcích, je přehlednější, ale pomalejší. Navíc proběhne kontrola údajů. Druhý způsob, zadání matice v celku, je rychlejší. Můžeme totiž okopírovat matici z MATLABu do vstupního textového pole (tag `textarea`).

Kontrola zadaných údajů před odesláním formuláře neslouží k úplnému potlačení všech chyb v posílaných datech. Nezadání prvku, či celé matice, zadání nepovoleného znaku, nebo neodpovídající si rozměry matic jsou zachyceny, a tím alespoň trochu zmenšíme zatěžování serveru.

V M souboru dochází k zachycování výjimek. Zamezuje se tím, aby došlo k chybě na straně serveru. Vzniklou chybu vypisujeme uživateli.

Výstupní dokument MATLAB Web Serveru může být pouze HTML dokument.

Což je nevýhoda, pokud chceme zobrazit např. matematický výraz pomocí MathML. Pro Internet Explorer existuje program MathPlayer, který umožňuje zobrazení MathML v HTML i XML souboru. V HTML dokumentu ostatní prohlížeče, např. Mozilla, nezobrazí MathML správně. V XML dokumentu už s tímto problémem není. Toto řešíme přidáním odkazu na XML soubor do výstupního HTML dokumentu.

Ve výstupním dokumentu je využita konstrukce tabulky s nestandardním atributem `autogenerate`, kterému se přiřazuje jméno proměnné. MATLAB potom vytvoří tabulku podle velikosti proměnné. Použitím atributu `autogenerate` není dokument validní (podle standardů W3C).

Literatura

- [1] POLYX, LTD. *Polyx* [online]. Poslední revize 2004-1-1 [cit. 2004-05-18]. <<http://www.polyx.com>>.
- [2] THE MATHWORKS, INC. *Matlab* [online]. Poslední revize 2004-04-20 [cit. 2004-05-18]. <<http://www.mathworks.com/products/matlab>>.
- [3] THE MATHWORKS, INC. *Matlab Web Server* [online]. Poslední revize 2004-04-20 [cit. 2004-05-18]. <<http://www.mathworks.com/products/webserver>>.
- [4] W3C. *Dom level 1,2,3* [online]. Poslední revize 2004-04-07 [cit. 2004-05-18]. <<http://www.w3c.org/DOM>>.