
Interaktivní 3D grafika v HTML a PDF dokumentech

ROMAN PLCH, PETRA ŠARMANOVÁ

Příspěvek si všímá problematiky vkládání interaktivní 3D grafiky vytvořené CAS systémem Maple do webových stránek a PDF dokumentů. Popisuje, jak využít mapleovskou knihovnu JavaViewLib k exportu interaktivních objektů a jak lze tuto dynamickou grafiku následně pomocí programu JavaView prezentovat na webu. V druhé části příspěvku autoři popisují vytváření PDF dokumentů se zařazenou interaktivní 3D grafikou pomocí pdf \TeX u a balíčku movie15.

1. Úvod

Vhodně vytvořená a okomentovaná grafika přispívá k pochopení probírané problematiky a rozvoji geometrické představivosti studentů. Ilustrační grafiku lze použít k objasňování nového teoretického pojmu či závislosti daného jevu na parametrech, k dokreslení geometrického významu řešených úloh a případně k ověření „reálnosti“ řešení.

Jedním z možných dělení grafiky je na grafiku statickou a dynamickou. Mezi statickou grafiku počítáme jakékoliv obrázky, s nimiž nemůžeme dále manipulovat. Interaktivní grafika nám naproti tomu umožňuje aktivně pracovat s objektem, např. prohlédnout si ho ze všech stran, zvětšovat a zmenšovat, zobrazit detail vybrané části, zobrazit normálové vektory, měnit nastavení barev, průhlednost objektu a mnoho dalšího (dle možností zobrazovacího programu).

V první části příspěvku popíšeme, jak lze interaktivní grafické objekty vytvořit pomocí systému Maple za použití knihovny JavaViewLib a jak je lze prezentovat na webu (s využitím zobrazovacího programu JavaView). Druhá část příspěvku je věnována zkušenostem autorů s vkládáním interaktivních 3D objektů do PDF dokumentu, počínaje úpravami 3D grafiky v Maplu, jejím exportem do formátu VRML, převodem do formátu U3D pomocí programu Deep Exploration, až po začlenění do PDF dokumentu pomocí pdf \TeX u a balíčku movie15. Ve výsledném PDF dokumentu je tak zachována možnost interakce uživatele s 3D objektem bez nutnosti lokální instalace programu Maple či jiných grafických programů.

2. Interaktivní grafika v HTML dokumentu

JavaView je prohlížeč geometrických útvarů a výpočetní program napsaný v jazyce Java. Může být použit jako samostatná aplikace pro zobrazování a manipulaci

s 3D objekty nebo jako applet na webových stránkách pro vzdálené použití skrze webový prohlížeč.

Program JavaView nabízí velmi pěkné zobrazovací prostředí, které umožňuje přibližování, posunutí, rotace, ovládání barev, textur, řeší průhlednost, viditelnost, vyhlazování nerovností, animování a mnoho dalšího. JavaView umožňuje import a export 3D objektů různých formátů (JVX, VRML, OFF, OBJ, DXF, ...) a formátů souborů s grafikou programů Maple a Mathematica. JavaView nejenže uchovává všechny možnosti modifikace grafických objektů, které Maple umožňuje, ale dokonce ho v mnohém převyšuje.

2.1. Mapleovská knihovna JavaViewLib

K tomu, abychom mohli jednoduše exportovat z Maplu interaktivní grafiku, slouží mapleovská knihovna JavaViewLib (JVL), která se tak stává spojovacím článkem mezi systémem Maple a programem pro manipulaci s 3D grafickými objekty JavaView. Statické grafické výstupy z Maplu tak mohou být nahrazeny dynamickými obrázky zobrazenými pomocí appletu JavaView.

JavaViewLib umožňuje export interaktivní 3D grafiky do formátu MPL (interní datový formát Maplu) a JVX (oficiální formát programu JavaView, založený na jazyku XML), s kterými si „rozumí“ program JavaView. Kromě těchto výstupů je umožněn přímý export do HTML kódu a okamžité zobrazení interaktivní 3D grafiky pomocí webového prohlížeče.

Knihovnu je možno zdarma stáhnout z adresy <http://www.javaview.de/maple/> a nakopírovat do adresáře, kde jsou uloženy mapleovské knihovny.

Na začátku práce v Maplu je třeba nastavit cestu ke knihovně:¹

```
> libname:="C:\\Program Files\\Maple 11\\LIB\\JavaViewLib\\";
> libname;
```

Knihovnu načteme příkazem:

```
> with(JavaViewLib);
```

```
[exportHTM, exportHTMLite, exportJVX, exportMPL, exportValidate, genTag,
genTagLite, getBrowser, getInfo, getInfoState, getOS, import, importJVX,
importMPL, runApplet, runAppletLite, runJavaView, runMarkupTree, set,
setEnabledValidate, setWorkingPath, viewGallery]
```

Přítom se nám zobrazí seznam funkcí, které JVL nabízí. Některé funkce mají příponu Lite. Tyto využívají alternativní verzi programu, tzv. JavaViewLite. Jedná se o verzi optimalizovanou pouze pro prohlížení geometrických objektů bez možnosti větších manipulací s těmito objekty. Výhodou je rychlejší načítání appletů.

¹Nastavení cesty závisí na použitém operačním systému.

2.2. Export grafů

Maple používá k popisu generované grafiky vlastní datovou strukturu, užívající jen prostého textu. První z možností exportu pomocí JavaViewLib využívá přímo tohoto zápisu, téměř nezměněný ho přepíše do vnějšího souboru s příponou MPL.

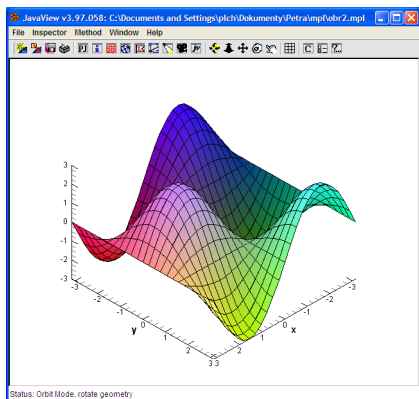
- > `graf:=plot3d(sin(x+y), x=-Pi..Pi,y=-Pi..Pi,axes=framed):`
- > `exportMPL(graf,"sin"):`

V pracovním adresáři (adresář C:\Program Files\Maple 11\JavaViewLib, pokud není nastaven příkazem `setWorkingPath` jiný adresář), podadresáři `mpl`, se uloží soubor `sin.mpl`. Při zobrazení souboru MPL pomocí JavaView se respektuje většina parametrů příkazu `plot3d` pro vykreslování grafu. Stačí tedy vytvořit výslednou podobu grafu v Maplu a dále nemusíme soubor MPL nijak upravovat (obr. 1).

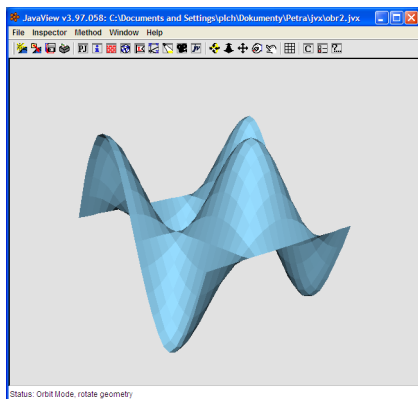
Analogicky fungují příkazy pro export do formátu JVX, mateřského jazyka JavaView. Tento je založený na značkovacím jazyce XML.

- > `exportJVX(graf,"sin"):`

Při exportu do formátu JVX pomocí příkazu `exportJVX` dojde k uložení zobrazovaného objektu, nikoliv k uložení nastavení parametrů zobrazení (barvy, osy, osvětlení atd., obr. 2). Chceme-li uchovat parametry zobrazení 3D grafiky nastavené v Maplu, je vhodnější exportovat grafiku z Maplu do MPL, soubor MPL zobrazit pomocí JavaView a teprve poté uložit grafiku do souboru JVX a parametry zobrazení do souboru JVD (Display Settings). Chceme-li tuto grafiku znovu zobrazit, načítáme do JavaView oba soubory (JVX i JVD). Stejně postupujeme při jakýchkoliv úpravách provedených přímo programem JavaView.



Obrázek 1: Export do MPL



Obrázek 2: Export do JVX

Příkaz

```
> exportHTML(graf, "sin"):
```

uloží soubor `sin.htm` do podadresáře `htm` pracovního adresáře. HTML soubor obsahuje párový tag `<applet>`, v němž jsou přímo zapsána data exportovaného grafu, stejná, která by byla zapsána do externího souboru formátu MPL. Žádný další soubor tedy neexistuje. Pokud jako druhý parametr příkazu `exportHTML` použijeme jméno souboru s extenzí MPL nebo JVX, vytvoří se soubor `sin.htm` (v podadresáři `htm`), který obsahuje odkaz na soubor `sin.mpl` (v podadresáři `mpl`), resp. `sin.jvx` (v podadresáři `jvx`).

Chceme-li vytvořit HTML soubor s více applety, je výhodné příkazem `genTag` vygenerovat zdrojový kód appletu přímo do mapleovského zápisníku a pak tento kód kopírovat do HTML souboru.

```
> genTag(graf, "sin.mpl"):
```

Příkaz `viewGallery` vytvoří stránku s odkazy na všechny soubory, které jsou uloženy v podadresáři `htm` pracovního adresáře.

2.3. Zobrazení grafů pomocí JavaView

Příkaz

```
> runJavaView(graf):
```

spustí program `JavaView` a zobrazí v něm zadaný graf funkce.

Příkaz

```
> runApplet(graf):
```

provede totéž jako příkaz `exportHTML(graf)`, jen navíc otevře vygenerovanou stránku ve webovém prohlížeči (obr. 3).

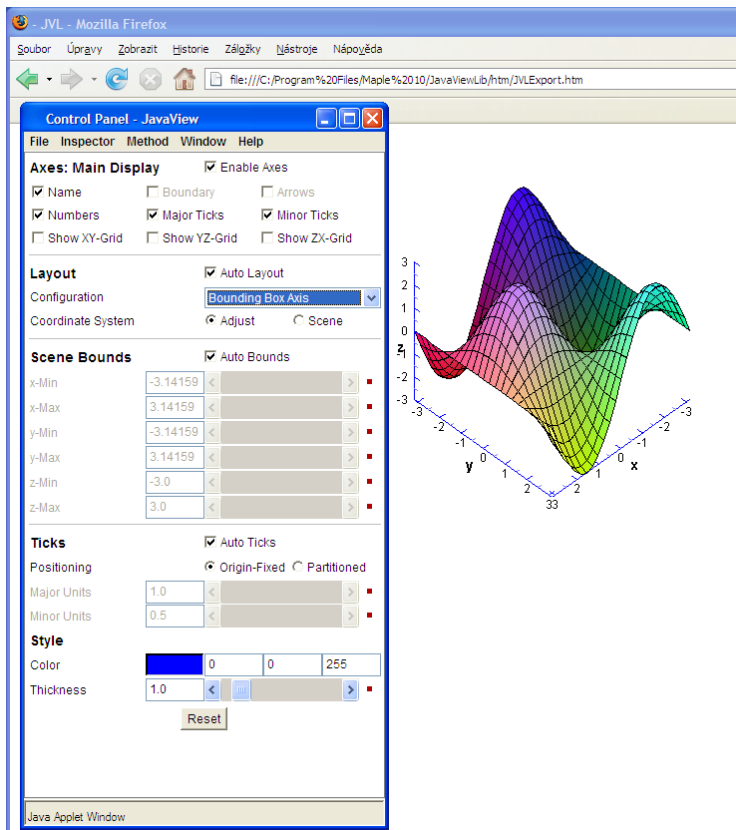
2.4. Ostatní příkazy

Seznam parametrů a jejich hodnot získáme příkazem:

```
> getInfo();
```

```
JavaViewLib State Information
```

```
-----+-----  
[W ] Applet Width      | 300  
[H ] Applet Height    | 300  
[A~] Applet Alignment  | center  
[CB] Applet Codebase   | ..  
[AS] Applet Archive   | jars/javaview.jar,jars/jvx.jar  
[AL] Applet Archive Lite | jars/jvLite.jar  
[BG] Applet Backcolor  | 200 200 200  
[R ] AutoRotate       | show 1. 1. 1.  
[X ] Axes             | hide
```



Obrázek 3: Ukázka začlenění interaktivní grafiky do HTML dokumentu

```
[BC] Background Colour | show 255 255 255
[BI] Background Image | hide images/jv1Logo.gif
[B ] Border | hide
[BB] Bounding Box | hide
[V~] Camera Direction | show 0 0 1
[DC] Depth Cueing | hide
[EA] Edge Aura | hide
[PT] Show jvx Points | hide small
[ED] Show jvx Edges | hide
[FC] Show jvx Faces | show
[T ] Title |
[WK] Working Path | C:\Program Files\Maple 11\JavaViewLib\
```

```
[IP] Installation Path | C:\Program Files\Maple 11\JavaViewLib\
[OS] Operating System | Windows XP
[BR] Browser | firefox
[JV] JavaView Runtime | bin\javaview.bat
[DG] Debugging | false
[HTM] HTM Extension | htm
[MPLVER] Maple Version | 11.020000
```

Vypsání hodnoty konkrétního parametru dosáhneme příkazem:

```
> getInfoState("BB");
                                "hide"
```

Nastavení provedeme pomocí příkazu set:

```
> set(W="600", H="800");
```

Příkazem

```
> set(reset);
```

nastavíme znovu původní parametry.

Příkaz

```
> setWorkingPath("C:\Documents and Settings\plch\Dokumenty"):
```

nastaví cestu k pracovnímu adresáři a vytvoří automaticky podadresáře `mpl`, `jvx`, `htm` a `images`, `jars` a `models`. Toto nastavení zůstává platné jen v aktuálně otevřeném zápisníku. Po zavření a znovunačtení zápisníku je nastavení zrušeno.

Nápovědu ke knihovně JVL spustíme jedním z příkazů `?JVL`, `?JavaView` nebo `?JavaViewLib`.

Kromě exportu knihovna JVL nabízí i import souboru do Maplu. Grafický objekt formátu MPL nebo JVX můžeme příkazem `import` načíst do Maplu. Chceme-li importovat soubor jiného formátu než MPL a JVX, musíme nejprve načíst tento soubor do JavaView a uložit ho v jednom z požadovaných formátů.

Příkaz `runMarkupTree` načte informace z mapleovské grafiky nebo souboru JVL (JVX nebo MPL) a vygeneruje odpovídající XML reprezentaci, kterou zobrazí v okně webového prohlížeče. Při použití tohoto příkazu nedochází k exportu (vytváření) souborů.

V případě, že máme problémy např. se spuštěním JavaView, můžeme se příkazem `set(DEBUG=true)` přepnout do režimu ladění. Program JavaView nám pak vypisuje podrobnější informace o běžících procesech. Nastavením hodnoty `DEBUG` na `false` toto zobrazování opět vypneme.

Podrobný popis všech příkazů knihovny lze najít v nápovědě k JVL přímo v Maplu.

2.5. Prezentace interaktivní grafiky na webu

Pro zobrazení 3D objektů na webu je třeba nakopírovat JavaView do adresáře, kde jsou umístěny HTML dokumenty. Při prohlížení HTML stránky se pak

automaticky spustí applet JavaView a zobrazí zadaný soubor s 3D objektem.

Dále uvádíme příklad volání appletu v HTML dokumentu:

```
<APPLET code=javaview.class height=200 width=200
  archive=jars/javaview.jar,jars/jvx.jar,jars/vgpapp.jar>
<PARAM NAME=cabbase VALUE=jars/javaview.cab,jars/jvx.cab,
  jars/vgpapp.cab>
<PARAM NAME=model VALUE=obr.mpl>           %jméno souboru s 3D objektem
<PARAM NAME=axes VALUE=show>               %zobrazuje osy
<PARAM NAME=autorotate VALUE=show>        %spustí rotaci objektu
<PARAM NAME=control VALUE=Hide>           %kontrolní panel je skrytý
<PARAM NAME=background VALUE=255 0 0>     %barva pozadí
</APPLET>
```

Všechny parametry, které je možno použít při volání appletu jsou popsány na adrese <http://www.javaview.de/doc/userManual/appletParm.html>.

Chceme-li využít JavaView pouze pro zobrazení připravených interaktivních geometrií a nepotřebujeme objekt modelovat, je vhodné využít lite verzi JavaView, která je při načítání appletů mnohem rychlejší než plná verze.

Výše popsaným způsobem je možno vytvářet galerie interaktivních 3D objektů k podpoře různých partií matematiky. Hotové galerie k diferenciálnímu a integrálnímu počtu funkcí více proměnných jsou volně přístupné na adresách <http://www.math.muni.cz/~plch/diplomky/musil/> a <http://www.math.muni.cz/~plch/diplomky/galerie/>. V těchto galeriích jsou prezentovány nejen interaktivní 3D grafy funkcí, ale i 3D animace. Zařazení animací je provedeno stejným způsobem jako zařazení ostatních objektů.

3. Interaktivní grafika v PDF dokumentu

I když se možnosti publikování matematiky na webu s rozvojem podpory jazyka MathML výrazně zlepšují, PDF zůstává pro matematiky stále nejvhodnějším formátem. V předchozích odstavcích jsme si ukázali možnosti vkládání interaktivní grafiky do dokumentu na webu, vyvstává tedy přirozená otázka, zda je něco podobného možné i v rámci PDF dokumentu.

Tato možnost úzce souvisí s vytvořením a rozvojem formátu U3D. Specifikace tohoto formátu byla zveřejněna v září roku 2005 organizací Ecma². Formát byl navržen jako standard pro data 3D objektů. Cílem bylo vytvořit jednotný datový formát pro jednoduchou výměnu dat a komunikaci mezi aplikacemi zaměřenými na práci s grafickými objekty. Tento formát začala podporovat i firma Adobe a to od specifikace PDF-1.6 (Adobe Acrobat 7.0 a Adobe Reader 7.0).

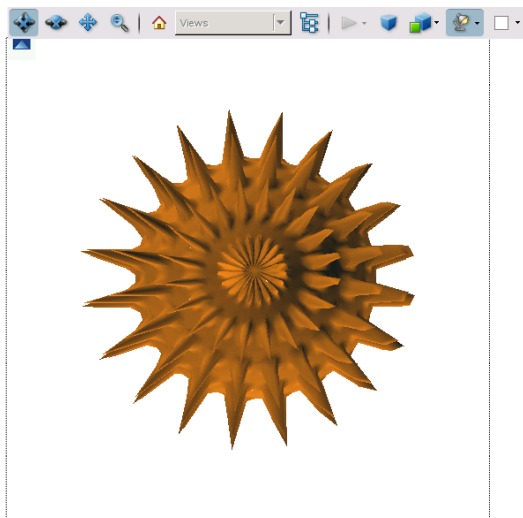
²Mezinárodní standardizační instituce.

3.1. Vytvoření 3D objektu ve formátu U3D

K vytváření „matematických“ 3D objektů je možno využívat celou řadu specializovaných či obecných matematických programů. V dalším se ale omezíme pouze na popis převodu matematické 3D grafiky z programu Maplu do formátu U3D.

Aktuální verze Maplu (11) nenabízí³ možnost exportu 3D grafiky do formátu U3D. Otestovali jsme několik cest pro převod mapleovské grafiky do formátu U3D – přímé uložení grafiky v Maplu ve formátech DXF a VRML, zobrazení mapleovské grafiky pomocí JavaView a následné uložení ve formátech OBJ nebo VRML. Nejlepší výsledky dává export z Maplu do VRML a následný převod do U3D formátu pomocí komerčního programu Deep Exploration (<http://www.righthemisphere.com/products/dexp/>)⁴.

Popíšme si nyní nejjednodušší způsob, jak lze z mapleovské grafiky získat soubor U3D. Postup si ilustrujme na konkrétním příkladě (výsledek je na obrázku 4).



Obrázek 4: Grafický objekt ve formátu U3D

1. Vytvoříme obrázek v Maplu (obrázek převzat z [8]):

³Prozatím není export do U3D formátu implementován ani v jiných matematických programech.

⁴V době vzniku tohoto příspěvku jsme objevili volně šiřitelný program pro konverze do formátu U3D Meshlab (<http://meshlab.sourceforge.net/>). Nemáme s ním ale prozatím praktické zkušenosti.


```

> pict:=plot3d(1+sin(10*x)^8*sin(10*y)^8/2,x=0..2*Pi,
> y=0..Pi,coords=spherical,grid=[100,100],
> style=patchnogrid, scaling=constrained,
> projection=.5,color=[.8,.6,.2]);

```

Přímý export z Maplu do VRML provedeme pomocí příkazu `vrm1` z balíčku `plottools`. Příkazem

```

> plottools[vrm1](pict,"ob1.wrl");

```

vytvoříme soubor `ob1.wrl` v požadovaném formátu. Ne všechny parametry mapleovské grafiky se však při exportu zachovají – více o tom pojednáme v kapitole 3.2.

2. Soubor `ob1.wrl` načteme programem Deep Exploration. Při jeho otevírání je třeba v menu „Settings“ nastavit „Flip coordinate system“ na hodnotu „No“, jinak bude program Deep Exploration používat jiný souřadný systém, než program Maple.

V programu Deep Exploration můžeme před konečným exportem do formátu U3D grafiku upravovat. Lze použít různé transformace jako je posunutí, rotace, zmenšení nebo zvětšení ve směru souřadných os. To vše můžeme provádět přímo myší na scéně nebo hodnoty transformací vyplnit do předložené tabulky.

Před uložením grafického objektu je dále nutné v menu „Tools“ – „3DTools“ nastavit „Convert To Double Sided“, jinak se může stát, že při pohledu ze zadní strany bude objekt neviditelný.

3. Provedeme uložení grafiky do formátu U3D.
4. Soubor U3D zařadíme do PDF dokumentu (více viz. kapitola 3.3).

Na obrázku 4 vidíme začleněnou grafiku i s nástroji 3D Toolbaru, který je součástí Adobe Readeru. Toolbar se zobrazí umístěním kurzoru myši na obrázek. Základní možnosti Toolbaru jsou dynamický zoom, posunutí, natočení, změna osvětlení, změna barvy pozadí či skrytí, zobrazení nebo izolování pouze určitých prvků modelu. Možné je rovněž využití různých zobrazovacích módů (Solid, Transparent, Shaded Illustration atd.).

3.2. Úskalí exportu z Maplu do VRML

Jak jsme již naznačili v předcházející části, vzhled grafiky ve VRML souboru se může lišit od podoby, kterou nastavíme parametry příkazu `plot3d` v Maplu. Tato část je proto věnována jednomu úskalí exportu grafiky z Maplu do VRML.

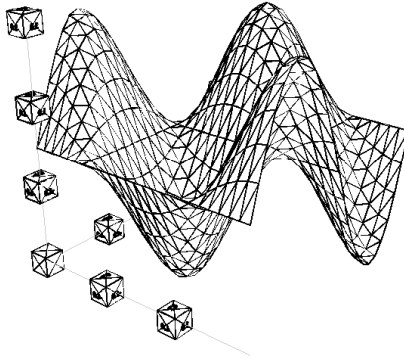
Příkazy

```

> vrm1(plot3d(3*sin(x)*cos(y),x=-Pi..Pi,
> y=-Pi..Pi,tickmarks=[3,3,3],axes=frame), 'soubor.wrl',
> tickback_color=brown,tickfore_color=yellow);

```

exportujeme funkci společně s osami. Značky na osách budou ale znázorněny jako krychle, na jejichž stranách je uveden popis. Výsledek (obr. 5) není pro matematické publikace dobře použitelný.



Obrázek 5: Ukázka exportu os z Maplu do VRML

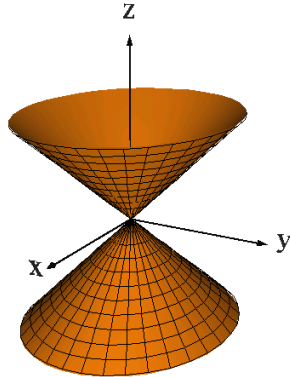
Vzhledem k tomu, že zobrazení souřadných os je v „matematické“ grafice často nezbytné, připravili jsme v Maplu proceduru pro kreslení os tak, aby zůstávaly při exportu do VRML (a následně do U3D) zachovány. Procedura `PlotAxes` je ke stažení na adrese <http://www.math.muni.cz/~plch/u3d/>, její použití ilustruje následující příklad:

```
> fce:=plot3d([1.2*u*cos(v),1*u*sin(v),u],u=-3..3,v=0..2*Pi,
> style=patch):
> osy:=plotAxes(4,4,5,.015):
> ob1:=display([fce,osy], color=yellow):
> vrml(ob2,"ob2.wrl");
```

Osy jsou vykresleny procedurou `plotAxes`, jejíž parametry jsou délky os (po řadě x , y , z) a jejich tloušťka. Funkce a osy zobrazíme společně pomocí příkazu `display` a následně uložíme do souboru `ob2.wrl`, výsledek po exportu je znázorněn na obrázku 6.

3.3. Vložení grafického objektu ve formátu U3D do PDF dokumentu

Získaný grafický objekt ve formátu U3D vložíme do PDF dokumentu buď pomocí komerčního produktu Adobe Acrobat3D (<http://www.adobe.com/products/acrobat3d/>) nebo použijeme sázecí systém $\text{T}_{\text{E}}\text{X}$ a balíček `movie15` ([5]). V dalším se věnujeme pouze „nekomerční“ cestě, tj. využití $\text{T}_{\text{E}}\text{X}$ u a balíčku `movie15`.



Obrázek 6: Funkce a osy vytvořené procedurou `Plotaxes`

Pro přímý výstup do PDF dokumentu použijeme `pdfLATEX` (požadována verze nejméně 1.20). Je-li PDF dokument generován cestou `LATEX → dvips → ps2pdf`, je nutno po balíčku `movie15` načíst ještě balíčky `graphics` a `hyperref`. Pro korektní zobrazení výsledného dokumentu musíme použít Adobe Reader verze 8.1. (a vyšší).

Balíček načteme ve zdrojovém souboru příkazem

```
\usepackage[3D]{movie15}
```

a vlastní začlenění interaktivní grafiky provedeme příkazem `\includemovie`, jehož syntaxe je:

```
\includemovie[volby]{šířka}{výška}{soubor.u3d}.
```

Podrobný popis všech voleb pro začleňování 3D grafických objektů najdeme v manuálu k balíčku `movie15` ([5]). Zmíníme jen některé z nich.

Volbou `3Dlights=<lighting scheme>` nastavíme osvětlení objektu, například `3Dlights=Day` nastaví denní barvy. Implicitně se používá osvětlení specifikované v 3D modelu.

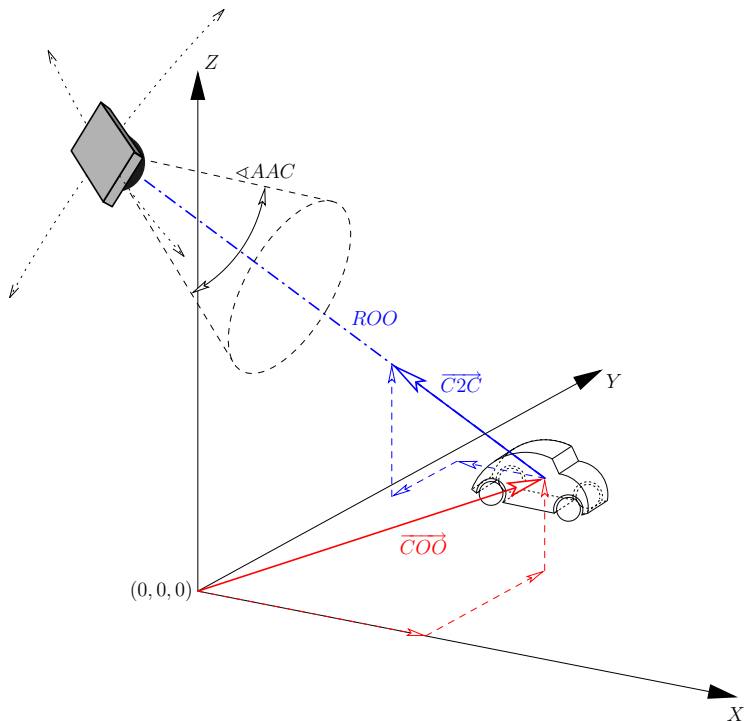
Volbou `3Dbg=<r> <g> ` nastavíme barvu pozadí. Hodnoty je možné zadávat jako čísla v pohyblivé desetinné čárce v rozsahu od 0 do 1.

Volbou `3Drender=<render mode>` určíme zobrazovací mód, např. zobrazení drátěného modelu nastavíme pomocí `3Drender=Wireframe`.

Za nejdůležitější volby považujeme ty, kterými ovlivňujeme umístění objektu na scéně, případně různé pohledy na objekt. Budeme se jim proto věnovat podrobněji.

3.4. Optimální umístění objektu na scéně

Umístění objektu na scéně (obr. 7) je specifikováno pomocí vektoru \overline{COO} směřujícího z počátku soustavy souřadnic do středu objektu (volba 3Dcoo), pomocí vektoru $\overline{C2C}$ směřujícího ze středu objektu do virtuální kamery (volba 3Dc2c) a pomocí vzdálenosti ROO virtuální kamery od objektu (volba 3Droo). Kromě toho lze volbou 3Daac nastavit průzorový úhel (aperture angel) kamery a volbou 3Dro11 otočení kamery o daný úhel kolem optické osy.



Obrázek 7: Umístění objektu na scéně, převzato z [5]

Pokud nejsou tyto parametry nastaveny, je virtuální kamera umístěna na pozici $(0, 0, 0)$ a dále $3Droo=0$, $3Dcoo=0 \ 0 \ 0$, $3Dc2c=0 \ -1 \ 0$, $3Daac=30$, $3Dro11=0$.

Vzhledem k tomu, že výpočet optimálního nastavení těchto parametrů je poměrně obtížný, je výhodné použití příkazu `\movieref` s parametry `3Dcalculate` nebo `3Dgetview`, které nám pomohou určit hodnoty výše zmíněných parametrů.

K určení vzdálenosti virtuální kamery tak, aby byl vidět celý objekt (nebo skupina objektů) slouží následující konstrukce:

```
\includemovie[toolbar,label=navesti_1]
```

```
{\linewidth}{\linewidth}{soubor.u3d}\\  
\moveref[3Dcalculate]{navesti_1}{Výpočet optimální vzdálenosti}
```

Nyní opakovaně překládáme zdrojový dokument (dokud se nezbavíme varovných hlášení). Ve výsledném PDF dokumentu se vytvoří stránka s odkazem „Výpočet optimální vzdálenosti“. Kliknutím na tento odkaz se nám objeví okénko, kde jsou uvedeny optimální hodnoty parametrů 3Droo a případně 3Dcoo. Tyto hodnoty zkopírujeme do zdrojového textu a odstraníme příkaz na vytvoření odkazu. Ponecháme tedy jen příkaz `\includemovie` doplněný o číselné parametry:

```
\includemovie[toolbar,label=navesti_1,3Droo=23.33,  
3Dcoo=0 0.55 15]{\linewidth}{\linewidth}{soubor.u3d}
```

Pokud chceme ovlivnit i natočení objektu na scéně, je výhodné použít příkaz `\moveref` s parametrem `3Dgetview`.

```
\includemovie[toolbar,label=navesti_2,3Droo=23.33]  
{\linewidth}{\linewidth}{soubor.u3d}\\  
\moveref[3Dgetview]{navesti_2}{Výpočet optimálního pohledu}
```

Ve výsledném PDF dokumentu pak vidíme objekt, který pomocí prostředků Toolbaru Adobe Readeru nastavíme do námi požadované polohy. V případě, že použijeme volbu `toolbar`, zobrazí se tento umístěním kurzoru myši na obrázek. V opačném případě můžeme Toolbar vyvolat pravým tlačítkem myši (Show Toolbar / Hide Toolbar).

Kliknutím na odkaz se pak otevře okno s hodnotami parametrů `3Dcoo`, `3Dc2c`, `3Droo`, `3Droll` a `3Daac`, které odpovídají aktuálnímu pohledu v Readeru. Výpis hodnot je formátován v následující syntaxi

```
{<coo_x> <coo_y> <coo_z>}{<c2c_x> <c2c_y> <c2c_z>}  
{<roo>}{<roll>}{<aac>}
```

Jednotlivé hodnoty pak překopírujeme do odpovídajících parametrů příkazu `\includemovie`. Při použití této konstrukce je vhodné do příkazu `\includemovie` uvést již odečtený parametr `3Droo`.

3.5. Vytvoření externího souboru s různými pohledy na scénu

S využitím příkazu `\moveref[3Dgetview]` je také možno vytvořit pojmenované pohledy na scénu. V Adobe Readeru si nastavíme objekt do námi požadované polohy, odečteme parametry a tyto uložíme do externího souboru `<views file>`. Tento soubor obsahuje pro každý přednastavený pohled řádek s následující syntaxí:

```
[<name>]{<coo_x> <coo_y> <coo_z>}{<c2c_x> <c2c_y> <c2c_z>}  
{<roo>}{<roll>}{<aac>}
```

Parametr `<name>` je volitelný a určuje jméno daného pohledu. Pro parametry `<coo_x>`, `<coo_y>`, `<coo_z>`, `<c2c_x>`, `<c2c_y>`, `<c2c_z>`, `<roo>`, `<roll>` a `<aac>` platí stejná pravidla jako pro odpovídající parametry ‘3Dcoo’, ‘3Dc2c’,

'3Droo', '3Droll' a '3Daac'. Je možné ponechat některé parametry prázdné použitím {}, v takovém případě se použije implicitní hodnota pro daný parametr.

3.6. Zobrazení různých pohledů na scénu

Chceme-li ve výsledném dokumentu pracovat s různými pohledy na 3D objekt, máme několik možností.

1. Pomocí volby `3Dviews=<views file>` příkazu `\includemovie` specifikujeme externí soubor s přednastavenými pohledy. Ve výsledném PDF dokumentu se tyto pojmenované pohledy zobrazí v Toolbaru a je možno z nich vybírat.
2. Pomocí příkazu `\movieref` vytvoříme hypertextový odkaz na 3D objekt. Přitom nastavíme jiný pohled na scénu. Syntaxe je

```
\movieref[<options>]{<label spec>}{<text>}
```

Přitom `<text>` specifikuje text, který se v PDF dokumentu zobrazí jako hypertextový odkaz a `<label spec>` je označení 3D objektu (objekt byl označen pomocí volby `label=<label spec>`). Pomocí `<options>` lze nastavit pohled na zobrazovaný objekt. Využít lze '3Dcoo', '3Dc2c', '3Droo', '3Droll' a '3Daac'.

Následující příklad ilustruje zobrazení objektu z různých pohledů, které lze zvolit kliknutím na nabídnuté odkazy.

Podívejte se na danou funkci z různých pohledů:\\

```
\movieref[3Droo=6, 3Dc2c=0 0 1, 3Droll=136]{obr1}
```

{Pohled shora}.

```
\movieref[3Droo=6, 3Dc2c=0.10 -0.99 -0.08, 3Droll=-112]{obr1}
```

{Pohled zespoďu}.

3. Kombinací předchozích možností je využití příkazu `\movieref` a externího souboru s přednastavenými pohledy. Pomocí volby `3Dviewindex=<index>` zvolíme předdefinovaný pohled na 3D objekt. `<index>` může obsahovat F, L, N nebo P pro přístup k první, poslední, další nebo předchozí položce v seznamu pohledů nebo číslo určující index daného pohledu v seznamu. D nastaví implicitní pohled.

Dále uvádíme zdrojový text k ukázce na obrázku 8.

```
\includemovie[toolbar,label=pict1,3Drender=SolidWireframe,
```

```
3Dcoo=0.42 0.43 0.17, 3Droo=5.5, 3Dc2c=0.73 0.40 0.53,
```

```
3Droll=-0.32, 3Dviews=view.txt, 3Dlights=Headlamp]
```

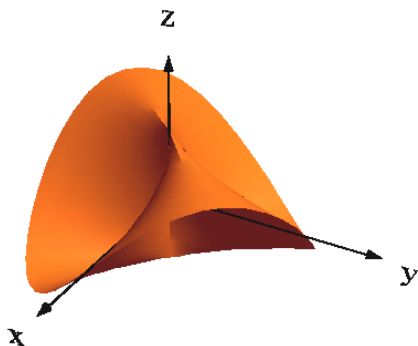
```
{0.6\linewidth}{0.6\linewidth}{p1_os3.u3d}
```

Podívejte se na danou funkci z různých pohledů --

```
\movieref[3Dviewindex=0]{obr1}{\textcolor{red}{shora}},
```

```
\movieref[3Dviewindex=1]{obr1}{\textcolor{red}{zdola}} a
```

`\movieref[3Dviewindex=2]{obr1}{\textcolor{red}{ze strany}}.`



Podívejte se na danou funkci z různých pohledů – [shora](#), [zdola](#) a [ze strany](#).

Obrázek 8: Hypertextové odkazy na předdefinované pohledy

3D objekty v PDF dokumentech lze ovládat i pomocí JavaScriptu. Script se specifikuje volbou `3Djscript=<JavaScript file>` a spustí se aktivováním 3D objektu v dokumentu. Více o použití JavaScriptu v PDF dokumentech v [1].

4. Závěr

Program JavaView je používán již od roku 1999. Jeho možnosti jsou neustále rozšiřovány a doplňovány, stejně tak je neustále zdokonalováno i jeho propojení s CAS systémy Maple, Mathematica a Mupad. Publikování interaktivní 3D grafiky na webu je tedy z pohledu matematiků výborně vyřešeno a nesetkáváme se při něm s výraznějšími problémy. Pomocí JavaView můžeme zobrazovat i 3D grafiku vytvořenou programem Matlab (3D objekt v Matlabu exportujeme do souboru formátu VRML a ten poté otevřeme pomocí JavaView).

Začlenění podpory pro vkládání interaktivních 3D objektů do formátu PDF výrazným způsobem ovlivnilo možnosti vytváření elektronických matematických publikací. Interaktivní matematickou grafiku můžeme nyní vkládat do textů vysoké typografické kvality, připravovaných systémem $\text{T}_{\text{E}}\text{X}$. Ve výsledném dokumentu přitom zůstává zachována možnost interakce uživatele s 3D objektem. Na

základě našich zkušeností je možné interaktivní grafiku vytvořenou programem Maple poměrně snadno převádět do formátu U3D a vkládat do PDF dokumentů.

Kromě programu Deep Exploration, popisovaného v tomto příspěvku, je ke konverzi do formátu U3D možno používat i Acrobat3D Toolkit. Tento program je dodáván společně s Acrobatem 3D a vzniká ve spolupráci Adobe se společností Right Hemisphere (tvůrci programu Deep Exploration).

Současný prudký rozvoj v této oblasti nás však opravňuje doufat, že brzy bude možné exportovat 3D objekty z Maplu do formátu U3D buď přímo, nebo pomocí programu JavaView, a tvorba interaktivních matematických textů tak bude ještě jednodušší než dosud.

Reference

- [1] Adobe: *JavaScript for Acrobat 3D Annotations* [online], duben 2007. Dostupné na www: http://www.adobe.com/devnet/acrobat/pdfs/js_3d_api_reference.pdf.
- [2] Došlá Z., Plch R., Sojka P.: *Matematická analýza s programem Maple. Díl 1, Diferenciální počet funkcí více proměnných*. Brno, Masarykova univerzita, 1999. CD-ROM. ISBN 80-210-2203-5.
- [3] Dugaro, S. P., Polthier K.: *Visualizing Maple Plots with JavaViewLib* [online]. Dostupné na www: <http://www.zib.de/polthier/articles/jvLib/jvLibFinal.pdf>.
- [4] Filipec Z., Plch R.: *Maple a JavaView*. Využití Maplu ve výuce a výzkumu na vysokých školách a akademiích věd, 1. vyd. Brno: Econ publishing s. r. o., 2002, ISBN 80-86433-10-2.
- [5] Grahn A.: *The movie15 package* [online], leden 2008. Dostupné na www: <http://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/movie15/doc/movie15.pdf>.
- [6] Gregor P.: *3D v PDF*, bakalářská práce, Brno, Masarykova univerzita, 2008.
- [7] *Javaview – interactive 3D geometry and visualization*, leden 2008. <http://www.javaview.de/>.
- [8] Klimek G., Klimek M.: *Discovering Curves and Surfaces with Maple*, Springer-Verlag 1997, ISBN 0-387-94890-2.
- [9] Kuben, J.: *Diferenciální počet funkcí více proměnných*. 1. vydání, Brno, Vojenská akademie v Brně, 2001. Skriptum.
- [10] Musil V.: *Prezentace matematické grafiky (Integrální počet funkcí více proměnných) na webu s programem JavaView*, diplomová práce MU Brno, 2007.
- [11] Plch R., Šarmanová P.: *Interaktivní prezentace matematické grafiky na webu a v PDF dokumentech*. Sborník semináře Technologie pro e-vzdělávání 2007., Praha: ČVUT FEL, 2007, ISBN 978-80-01-03756-0. Dostupné na www: <http://acm.osu.cz/paper/tpev2007/TPEV-2007-el-sbornik.pdf>.

- [12] Plch R., Šarmanová P.: *Galerie interaktivní grafiky pro podporu výuky matematické analýzy*. Sborník příspěvků 3. konference Využití počítačů ve výuce matematiky. 1. vydání. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2007, ISBN 978-80-7394-048-5.
- [13] Šelleová S.: *Prezentace matematické grafiky na webu s programem Java View*, diplomová práce MU Brno, 2007.

Summary: Interactive 3D Graphics in HTML and PDF Documents

The paper presents the authors' experience with including interactive 3D objects into HTML and PDF documents, starting with modifying 3D graphics in Maple by means of the library `JavaViewLib`, followed by its export into the `MPL` or `JVX` format and finishing by the web integration. In the second part, the authors describe exporting Maple 3D graphics into the `VRML` format, then its conversion to `U3D` with the use of `Deep Exploration`, and finish with its embedding into a PDF document by means of `pdfTEX` and the `movie15` Package. This procedure preserves the possibility of the user's interaction with 3D objects even in the final PDF document without the necessity of the local installation of Maple or other graphical programs.

Roman Plch, plch@math.muni.cz
Petra Šarmanová, petra.sarmanova@vsb.cz