GRAPHICS TOOLS IN LATEX

Vladimír POLÁŠEK – Lubomír SEDLÁČEK

Abstract: This paper deals with the basic tools for managing graphics and graphics creation in LaTeX with a focus on mathematics. It briefly describes how to embed external graphics objects, to edit them, and describes how to program the graphics focused on mathematics directly in LaTeX.

Key words: Mathematics, LaTeX, graphics programming.

GRAFICKÉ NÁSTROJE LATEXU

Resumé: Příspěvek se zabývá základními nástroji LaTeXu pro správu a tvorbu grafiky se zaměřením na matematiku. Stručně popisuje, jakým způsobem lze vkládat externí grafické objekty, upravovat je, a popisuje, jakým způsobem lze programovat grafiku zaměřenou na matematiku přímo v LaTeXu.

Klíčová slova: matematika, LaTeX, programování grafiky.

1 Úvod

Typografický systém LaTeX je určen pro tvorbu zejména odborných publikací, ale i dalších dokumentů, které vyžadují vysokou typografickou úroveň. Nezbytným prvkem každého z těchto materiálů je jeho grafické zpracování, jehož součástí především u textů z matematiky jsou různá schémata, diagramy, grafy funkcí. Běžný uživatel má spoustu možností, jak si tyto grafické objekty vytvořit pomocí volně stažitelného softwaru, např. Graph, Geogebra aj., nebo komerčních softwarů jako jsou Mathematica, Maple apod. U takto vytvořených obrázků ale není možné vždy zaručit to podstatné, typografickou kvalitu a jednotný vzhled vkládaného objektu a okolního textu. LaTeX umožňuje vkládat a spravovat obrázky, jak si naznačíme v tomto článku, ale především disponuje nástroji, na které bychom se v tomto článku chtěli zaměřit a které lze využít např. pro kreslení grafů funkcí jedné i více proměnných. Výhodou je, že si uživatel může vzniklé obrázky libovolně upravit a doplnit bez ztráty na jejich kvalitě.

2 Vkládání a správa grafiky z externích programů

LaTeX není editor obrázků jako třeba software Photoshop od firmy Adobe, přesto umožňuje jistou správu obrázků vkládaných do textu. Pro práci s obrázky je vhodné vyvolat balíček `graphicx`, tzn. v preambuli dokumentu napsat `\usepackage{graphicx}`. U tohoto balíčku můžeme navolit parametry typu `pdftex`, `dvips`, aj., což jsou ovladače pro

překladače LaTeX a PDFLaTeX (a další), a které umožňují vytvářet dokumenty ve zvoleném formátu. PDFLaTeX má např. přednastaven `pdftex` automaticky a výsledkem je tedy soubor formátu PDF.

V závislosti na překladači je nutné volit formát obrázku, který vkládáme do textu. Nebo obráceně, podle formátu obrázku zvolíme vhodný překladač. Budeme se nyní zabývat pouze nejužívanějšími překladači LaTeX a PDFLaTeX.

Jediný formát, který můžeme použít při překladu LaTeXem je Encapsulated PostScript (EPS). Je to tzv. vektorový formát, ale je schopen uchovávat i bitmapové obrázky. Existuje spousta softwarů, jež dokáží ukládat obrázky ve formátu EPS (přípona `.eps`). Mezi nejznámější lze zařadit Adobe Illustrator, CorelDRAW, které jsou komerční a navíc limitovány operačním systémem, do formátu EPS obrázky ukládá i Adobe PhotoShop. Dále lze jmenovat programy Inkscape, který ukládá v EPS formátu, je multiplatformní, ale nedokáže přímo EPS obrázky otevřít, nebo GIMP, což je grafický editor sloužící ke tvorbě a úpravě bitmapových obrázků. Nelze opomenout softwaru Mathematica, Maple apod., které dokáží exportovat zobrazené grafy do obrázků formátu EPS ale i dalších.

Při použití překladače PDFLaTeX máme širší výběr, pokud jde o formát vkládaného obrázku. Můžeme vkládat JPG, což je běžný formát obrázků nebo fotek stažených např. z internetu nebo digitálních fotoaparátů, dále PNG, což je asi nejvhodnější formát pro PDFLaTeX, dále lze vložit přímo soubor formátu PDF, který obsahuje

obrázky. Formát EPS nelze užít přímo, ale lze ho snadno převést pomocí programu epstopdf, který spolupracuje s TeXem, na soubor typu PDF.

V následujícím odstavci si naznačíme, jakým způsobem se vkládají obrázky do dokumentu, tedy možnosti jejich zápisu ve zdrojovém souboru LaTeXu. Příkaz

```
\includegraphics[parametr 1,  
parametr 2,...]{nazev obrazku}
```

má několik volitelných parametrů a povinný parametr „nazev obrázku“. Některé ze základních volitelných parametrů naleznete v následující tabulce

width=X	šířka obrázku X
height=X	výška obrázku X
scale=X	změna rozměrů v určitém poměru daném násobkem X
angle=X	rotace obrázku o X stupňů

Lze zvolit i více parametrů najednou, stačí je oddělit čárkou. Vkládaný obrázek se vloží do dokumentu přesně na místo, kde je ve zdrojovém souboru, což ale může způsobit nevhodné uspořádání stránky dokumentu. Proto se v dalším odstavci zmíníme i o tzv. plovoucích prostředí, které se automaticky starají o vhodné umístění obrázků, vkládají titulky a umožňují křížové odkazy. Vkládáme-li obrázky formátu eps, je možné využít i balíček epsfig a jeho příkaz \epsfbox. Pro vkládání celých stránek dokumentů formátu PDF je nutné vyvolat balíček pdfpages, který umožňuje vložit vícestránkové externí PDF soubory do vytvářeného PDF dokumentu pomocí příkazu

```
\includepdf[<key=val>]{<filename>  
},
```

kde <key=val> je seznam čárkou oddělených parametrů a <filename> název vkládaného PDF dokumentu. Tento příkaz má spoustu volitelných parametrů, mezi hlavní parametry patří např. pages pro výběr stran, landscape pro rotaci stránky o 90 stupňů, addtolist přidá novou stránku do seznamu obrázků nebo tabulek. Příkaz \includepdf[nup=2x1,pages={1,{}},scale=0.35]{kapitola_1.pdf} ve zdrojovém kódu vloží do vytvářeného dokumentu první stranu ze souboru kapitola_1.pdf, spolu s ní vloží prázdnou stranu a uspořádá je vedle sebe, obě strany jsou zmenšeny v určitém poměru.

Zmínili jsme pojem plovoucí prostředí. Představte si situaci např. v MS Word. Jestliže vložíme příliš velký obrázek na příslušnou stranu, tak se umístí na stranu další, ale zanechá velkou „díru“ v textu bez toho, aby se přeskupil následující text. Je třeba dalších manuálních úprav, zatímco LaTeX se stará o toto všechno automaticky. Do plovoucích prostředí vkládáme objekty, u kterých nesmí dojít ke zlomu při přechodu na novou stránku. LaTeX rozeznává plovoucí prostředí table, figure, ale lze vytvářet libovolně i vlastní prostředí. Tyto prostředí mohou mít titulky, které je popisují, a jsou také číslovány, takže se na ně lze odkazovat odkudkoliv z textu. K vytvoření plovoucích obrázků se využívá prostředí figure.

```
\begin{figure}[specifikace  
umístění]
```

```
... obsah ...
```

```
\end{figure}
```

Parametr specifikace umístění umožňuje autorovi jistou kontrolu nad tím, kde jsou plovoucí prostředí umístěna. Lze zvolit i více specifikací a LaTeX pak vybere tu nejvhodnější, např. specifikace h znamená umístit obrázek zde, tedy přibližně ve stejném bodě jako ve zdrojovém textu, specifikace t a b znamenají umístění nahoře nebo dole na stránce, kombinace h! přepíše vnitřní parametry LaTeXu pro určení „vhodné“ pozice plovoucího prostředí a umístí obrázek přesně na místo jako ve zdrojovém kódu, specifikace p umístí obrázek na samostatnou stranu, kde není obyčejný text, ale pouze plovoucí prostředí. Jako obsah plovoucího prostředí figure volíme příkaz

```
includegraphics s příslušnými parametry,  
příkaz caption pro vytvoření titulku, případně
```

```
příkaz label pro vytvoření tzv. návěští  
křížového odkazu, ale i další, které mohou sloužit  
např. k rozmístění a popisu více obrázků  
v jednom prostředí figure. Následující kód vloží  
do dokumentu obrázek jtie.png o šířce 10  
cm, výška se přizpůsobí automaticky, příkaz  
\centering umístí obrázek na střed stránky:  
\begin{figure}  
\caption{Logo JTIE.}  
\centering  
\includegraphics[width=10cm]{jtie  
.png}  
\label{logo}  
\end{figure}
```

Obr 1: Logo JTIE.

Uvnitř plovoucího prostředí je definován příkaz `\label{logo}`, tedy návěští křížového odkazu, příkaz `\ref{logo}` vrátí v dokumentu odpovídající číslo obrázku, ve stylu `article` a `report` jsou obrázky automaticky číslovány průběžně, ve stylu `book` se číslo skládá z čísla kapitoly a pořadového čísla v rámci kapitoly.

Dalším prostředím pro práci s obrázky je `wrapfigure`. Umožňuje tzv. obtékání obrázků, tedy vkládání grafiky a textu vedle sebe „do řádku“. Alternativou je využití balíčku `picins`, který umožňuje mnohem lepší kontrolu nad umístěním obtékané grafiky, ale není přímo součástí distribuce LaTeXu. Je možné vkládat ohraničení obrázků, pomocí dalších příkazů kontrolovat umístění popisku, druh a velikost rámečků.

Ke správě grafiky lze využít spoustu dalších rozšiřujících balíčků, např. `floatflt` nebo `placeins` a další. Je vhodné zmínit i balíček `color`, který pracuje s barvami u všech objektů – textových i grafických. Barvy se vybírají podle toho, jaký model zvolíme. Standardním modelem je `rgb`, jehož barva je dána trojicí čísel v rozsahu od nuly do jedné, které specifikují jas červené, zelené a modré složky dané barvy. Dalšími modely jsou `gray` pro práci s odstíny šedi a `cmymk`. Některé z barev jsou předdefinovány, pro definici jiné barvy slouží příkaz

```
\definecolor{jmeno}{model}{hodnoty}
```

Lze nastavit barvu textu, rámečku kolem textu nebo obrázku i celé strany dokumentu.

3 Přehled nástrojů pro programování grafiky v LaTeXu

Pro základní práci s grafikou je plovoucí prostředí `figure` plně postačující, a neslouží jen pro vkládání externích obrázků, ale je možné do něj zapisovat nebo spíše programovat grafiku přímo v LaTeXu a plovoucí prostředí s takto vytvořenou grafikou funguje stále na stejném principu. Programování v LaTeXu je jistě pro začínajícího uživatele náročné, ale s využitím vhodného balíčku, tedy jakéhosi souboru `maker` kompatibilního s makry TeXu, LaTeXu nebo

PDFLaTeXu, bude výstupem po grafické stránce kvalitní dokument, kde je důležitá především jednotnost vzhledu vytvořeného obrázku a okolního textu.

Kvalitní výsledek, ale omezené možnosti poskytuje prostředí `picture`, které je určeno pro kreslení vektorových obrázků složených z přímých čar, šipek, kružnic, oválů a libovolných textů.

Programovacím jazykem METAPOST uživatel dokáže vytvořit složitější vektorovou grafiku. METAPOST dokáže řešit např. lineární rovnice, vypočítat průsečík křivek a kreslit tak obrázky omezené matematickými nebo geometrickými souvislostmi. Nevýhodou je, že zdrojový kód obrázku se nepíše přímo do kódu vytvářeného dokumentu, ale zdrojový text si napíšeme v jakémkoliv textovém editoru a programem `mpost` tento text převedeme na postscriptový obrázek. Ten se pak vloží do dokumentu tvořeného v LaTeXu některým z výše popsaných způsobů. Programovací jazyk METAPOST je distribuován společně s LaTeXem a jeho princip vychází z METAFONTu, což program vytvořený D. Knuthem, autorem TeXu. METAFONT je původní grafický program pro TeX, ale slouží především k vytvoření návrhu a digitalizaci písem. Z jazyku METAPOST pak vychází balíček `mfpic`, pomocí kterého píšeme kódy obrázků přímo ve zdrojovém souboru LaTeXu, při překladu (např. PDFLaTeXem) se vytvoří externí soubory, na ně pak použijeme opět program `mpost`. Pomocí balíčku `mpic` lze mimo jiné kreslit grafy matematických funkcí, např. následujícím příkazem vykreslíme funkci $\sin x$ na intervalu :

```
\function{0,6.28,0.1}{sin x}
```

Zde METAPOST vykreslí funkci tak, že postupuje od 0 postupně po kroku 0,1 až k 6,28. Krok jsme zvolili libovolně, ale samozřejmě, čím menší krok, tím je křivka hladší.

PostScript je programovací jazyk určený pro popis stránek dokumentu tak, aby tento popis byl nezávislý na výstupním zařízení. Nejucelenějším balíčkem `maker` LaTeXu pro práci s PostScriptem je `pstricks`. Umožňuje vkládat postscriptovou grafiku přímo do zdrojového souboru LaTeXu, ať už se jedná o jednoduché křivky, grafy matematických funkcí jedné proměnné nebo i trojdimenzionální objekty. K tomu slouží spousta dalších doplňujících balíčků `maker`, jako jsou např. `pst-plot` a

pst-3dplot pro kreslení grafů funkcí, pst-node, pst-tree pro definování uzlů, stromů, matematických diagramů a jiné balíčky. Při překladu zdrojového souboru obsahující pstricks, je nutné použít překladač LaTeX, který vytvoří soubor s příponou dvi [1,2], který se pak převádí na rozšířený PDF formát. Pokud bychom chtěli použít přímo PDFLaTeX a vytvářet rovnou PDF soubory, celý proces bude fungovat tak, že se při překladu zdrojového souboru vytvoří zvláštní soubor s příponou pdf obsahující všechny obrázky vytvořené pomocí pstricks a ty se automaticky vloží do výsledného souboru. Pro účely tvorby výukových materiálů jsou upstricks výhodné rozsáhlé možnosti při tvorbě grafů funkcí jedné i dvou proměnných. Zvláštností je jejich zápis ve zdrojovém kódu, kdy se využívá tzv. postfixová notace (zkráceně RPN), což je způsob zápisu matematického výrazu, kde operátor následuje své operandy. Např. „1+2“ se v RPN zapíše jako „1 2 +“ nebo „cos x“ jako „x cos“. Pstricks navíc nepoužívá symboly pro základní operace jako je sčítání, ale používá vlastní příkazy. Srovnáme zápis pro vykreslení grafu funkce sin ve zdrojovém kódu LaTeXu s použitím pstricks oproti mfpic:

```
\psplot{0}{6.28}{x 57.296 mul sin}
```

Ze zápisu je zřejmý interval $[0, 2\pi]$ zadáný v radiánech. Dále $1 \text{ rad} = 57,296$ stupňů, převádíme radiány na stupně, protože funkce sin zde má argument ve stupních. Příkaz mul v pstricks znamená násobení. Tedy druhý argument příkazu psplot se v klasické notaci zapíše ve tvaru $\sin(57,296 \cdot x)$, kde x je dáno v radiánech. Veškeré využití pstricks naleznete v rozsáhlých manuálech [3].

Další grafický balík, který krátce popíšeme, je TikZ/PGF [4], kde PGF (zkratka pro „portable graphics format“) poskytuje základní příkazy pro tvorbu grafiky a TikZ je jeho nadstavbou pro snadnější uživatelský přístup. Syntaxe příkazů balíku TikZ vychází jak z METAPOSTU, tak i z pstricks, ale TikZ není plně kompatibilní ani s jedním z těchto nástrojů. Spolupracuje ale s libovolnými překladači, tedy výstupem mohou být soubory typu ps i pdf. Pro práci v TikZ je nutné v preambuli zdrojového kódu LaTeXu psát `\usepackage{tikz}`, volitelné jsou pak doplňující knihovny TikZ, které píšeme pomocí příkazu `\usetikzlibrary`. Ty poskytují další předdefinované grafické objekty, případně rozšiřují samotný systém TikZ/PGF. Knihovny

nejsou volány automaticky, protože většinu z nich uživatel při své práci nepotřebuje. Pro srovnání s předchozími nástroji, příkazem `\draw[domain=0:6.28] plot[smooth] (\x,{sin(\x r)});` vykreslíme graf funkce sinus na intervalu $(0, 2\pi)$. Zadané body `(\x,{sin(\x r)})` jsou příkazem plot s parametrem smooth spojeny do hladké křivky (oproti spojení lineárními úseky), kde proměnná $\x$ probíhá v intervalu daném parametrem domain. Existuje i další způsob, jak kreslit grafy funkcí jedné proměnné. K tomu je třeba mít nainstalován program Gnuplot. Kromě knihoven TikZ je možné využít dalších podpůrných balíčků, které rozšiřují výrazně možnosti TikZ. Jsou to např. tkz-2D pro zjednodušení kreslení diagramů ve dvoudimenzionálním souřadnicovém systému, tkz-graph a tkz-berge pro kreslení kombinatorických grafů, nebo 3dplot pro kreslení trojdimenzionálních souřadných systémů a jednodušších trojdimenzionálních schémat. Pokud jde o TikZ, nesmíme opomenout zmínit, že některé matematické softwary a grafické editory dokáží exportovat grafiku do kódu TikZ/PGF. Dále, dnes existují grafické editory, které dokáží překládat TikZ kód v reálném čase a současně obrázek znázornit ve vedlejším okně.

Ještě než přejdeme k poslední části tohoto článku, kde uvedeme ukázky již pro výuku připravených materiálů, zmíníme grafický balík PGFPLOTS. Ten plně vychází z TikZ/PGF, ale jeho znalost pro práci PGFPLOTS není nutná. Hlavním cílem balíku PGFPLOTS je jednoduše vytvářet kvalitní grafy funkcí ve dvou i třech dimenzích s důrazem na konzistenci velikosti a typu písma v celém dokumentu.

5 Ukázky grafiky v LaTeXu pomocí TikZ

Obrázky uvedené v tomto odstavci pocházejí ze studijních materiálů a učebních textů z matematiky pro střední nebo vysokou školu. Byly vytvořeny pomocí balíku TikZ, který je pro autory tohoto článku uživatelsky nejpřijatelnější, zejména pokud jde o tvorbu dvourozměrné grafiky.

Snadno lze pomocí TikZ vytvářet grafy funkcí jedné proměnné. Možností zde existuje několik, u jednodušších funkcí se využívá zápisu jako v předchozím odstavci u grafu funkce sinus. Na Obr. 2 vidíme ukázku tzv. transformace grafu, kdy ze znalosti základního grafu posouváním ve

směru osy x nebo y , příp. rotací kolem os, vzniká graf výsledné funkce.

Na dalším Obr. 3 jsme balík TikZ využili k náčrtu geometrického významu diferenciálu funkce jedné proměnné.

TikZ disponuje matematickými nástroji pro výpočet funkčních hodnot, průsečíků různých křivek apod. Toho jsme využili při úloze z planimetrie, kde bylo úkolem sestavit tečny ke dvěma zadaným kružnicím, kdy je známa vzdálenost jejich středů. Byla vytvořena prezentace, ve které se postupně odkrývaly jednotlivé kroky náčrtu. Výsledek je na Obr. 4.

Zajímavou ukázkou možností TikZ je tvorba animované grafiky. Na Obr. 5 je znázorněna definice funkce kosinus pomocí jednotkové kružnice.

Poslední Obr. 6 obsahuje jednoduchý náčrt ve trojrozměrném prostoru. Jedná se o grafické znázornění geometrického významu parciálních derivací funkce dvou proměnných.

Obr 2: Transformace grafu.

Obr 3: Diferenciál funkce jedné proměnné.

Obr 4: Úloha z planimetrie.

Obr 5: Slajd z animované prezentace.

Obr 6: Náčrt ve 3D.

6 Závěr

Použití výše uvedených nástrojů pro programování grafiky v systému LaTeX, zvláště grafického balíku TikZ, je určeno zejména náročnějšímu uživateli, který klade důraz na to, aby vysoká strukturální úroveň a typografická přesnost prezentovaného matematického textu korespondovala s vloženými obrázky, vytvořenými grafickými schématy, animacemi, diagramy, grafy, případně jinými grafickými objekty. Pomocí takto kvalitně zpracovaných vektorových schematických obrázků je usnadňováno pochopení poznatků, které jsou primárně vyjádřeny textem. V některých případech však mohou plnit i funkci hlavního zdroje učební informace, od kterého se pak verbální složka učiva odvíjí. [5] Je zřejmé, že vhodná prezentace učiva využitím těchto grafických nástrojů povede k výraznému zkvalitnění pedagogické práce, a to zvláště z hlediska názornosti a efektivity výuky.

7 Literatura

- [1] RYBIČKA, J. LaTeX pro začátečníky. 3. vydání. Brno: Konvoj, 2003. 238 s. ISBN 80-7302-049-1.
[2] POLÁŠEK, V., SEDLÁČEK, L. Tvorba výukových prezentací z matematiky v

typografickém systému LaTeX. *Journal of Technology and Information Education*. 2012, Olomouc - EU, Univerzita Palackého, Ročník 4, Číslo 1, s. 97 -102. ISSN 1803-537X (print). ISSN 1803-6805 (on-line).

[3] PSTricks web site [online]. TUG, 2012 [cit. 2012-10-05]. URL:< <http://tug.org/PSTricks/main.cgi/>>.

[4] TANTAU, T. TikZ and PGF, Version 2.10. (online). ©2007 [cit. 2012-05-18]. URL:< <http://sourceforge.net/projects/pgf>>

[5] LEPILO, O. Teorie a praxe tvorby výukových materiálů. 2010, Olomouc.

URL:< <http://zvyp.upol.cz/publikace/lepil.pdf>>
ISBN 978-80-244-2489-7.

Mgr. Vladimír Polášek, Ph.D.

Mgr. Lubomír Sedláček, Ph.D.

Ústav matematiky

Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Nad Stráněmi 4511

760 05, Zlín, ČR

Tel: +420 576 035 057

E-mail: vpolasek@fai.utb.cz,

lsedlacek@fai.utb.cz

Www pracoviště: www.fai.utb.cz