

L^AT_EX A PUBLIKOVANIE NA INTERNETE

RUDOLF BLAŠKO (SK)

Abstrakt. V dnešnej dobe značne pokročili nielen schopnosti počítačov, ale aj zručnosti ich používateľov. Nastal veľký rozmach webových prehliadačov a hlavne aplikácií, ktoré sú schopné v nich fungovať. V tomto smere nezaostávajú ani L^AT_EX, T_EX a s nimi spriaznené aplikácie a súbory. Rôzne animácie dvojrozmerné aj trojrozmerné už bez problémov dokáže L^AT_EX exportovať nielen do pdf súborov, ale aj na internet, kde si ich môže užívateľ bez problémov pozrieť vo webovom prehliadači.

Kľúčové slová. L^AT_EX, webový prehliadač, animácia, Asymptote, animate, media9.

L^AT_EX AND PUBLISHING ON THE INTERNET

Abstract. Nowadays, not only the capabilities of computers, but also the skills of their users have advanced considerably. There has been a great boom in web browsers and especially applications that are able to function in them. In this regard, L^AT_EX, T_EX and their related applications and files are not far behind. Latex can easily publish various animations two and three-dimensional not only to pdf files, but also to the Internet, where the user can view them without any problems in a web browser.

Keywords. L^AT_EX, web browser, animation, Asymptote, animate, media9.

Úvod

V dnešnej dobe značne pokročili nielen schopnosti počítačov, ale aj zručnosti ich používateľov. Nastal veľký rozmach webových prehliadačov a hlavne aplikácií, ktoré sú schopné „prevádzkovať“. Dnešní mladí ľudia skoro všetko skúmajú a riešia pomocou mobilného telefónu, preto sa neustále zvyšuje význam aplikácií schopných fungovať v mobilných telefónoch. Dôležitým faktorom sú aj technológie schopné preniesť informácie a činnosť týchto užívateľov práve do mobilného telefónu. To sú napríklad interaktívne odkazy a QR kódy. V tomto smere nezaostávajú ani technológie systému L^AT_EX.

Na tvorbu QR kódov existuje jednoduchý balíček `qrcode`, dvojrozmerné animácie vytvoríme pomocou balíčka `animate`, ktorý je schopný spojiť a nasledne riadiť sekvencie obrázkov. Tieto obrázky môžeme ovládať alebo sa môžu spúšťať samostatne a nezávisle od užívateľa. Tento proces môže byť ukončený alebo aj neukončený nekonečný. Na vkladanie multimediálnych súborov je vhodný balíček `media9`, pomocou ktorého môžeme do dokumentu vkladať filmy, zvukové

nahrávky ap. Pomocou tohto balíčka môžeme do dokumentu vkladať aj interaktívne 3D obrázky vytvorené pomocou veľmi dobrého programu na kreslenie 2D a 3D grafov funkcií *Asymptote*.

1. Tvorba QR kódov

QR kódy tvoria veľmi jednoduchý a zároveň veľmi účinný Open Source prostriedok na prenos informácií z tlačenej podoby na elektronické média. Postačí iba fotografický prístroj a vhodný softvér. Tejto problematike som sa už venoval na OSSConf v roku 2015 [2], preto iba stručne zopakujem ako jednoducho vytvoriť funkčný QR kód pomocou, ktorého budeme ďalej distribuovať vhodné informácie. Najjednoduchšie je použiť balíček `qrcode`, ktorý načítame do preambuly pomocou príkazov `\usepackage[forget]{qrcode}` alebo `\usepackage[final]{qrcode}`. Rozdiel medzi parametrami `forget` a `final`, je taký, že pri `final` sa vždy generuje nový obrazec, zatiaľčo pri `forget` sa nový obrazec generuje iba pri zmenách údajov. QR obrázok môžeme generovať priamo v hlavnom dokumente, alebo si ho vygenerovať zvlášť a načítať pomocou príkazu `\includegraphics`.

Pri samostatnom generovaní kódu je najvhodnejšie použiť triedu `standalone`, ktorá automaticky oreže formát výstupného súboru podľa veľkosti generovaného obrázka. Ilustruje to nasledujúci príklad, ktorý generuje QR kód s odkazom na domovskú stránku autora.

```
\documentclass[10pt]{standalone}
\usepackage[forget]{qrcode}

\begin{document}
  \qrcode[version=0,level=L,height=3cm]{
    https://frcatel.fri.uniza.sk/users/beerb}
\end{document}
```

Vygenerovaný QR kód načítame do hlavného dokumentu pomocou príkazu `\includegraphics` a pomocou príkazu `\href` ho môžeme v elektronickej forme aktivovať do interaktívnej podoby. V papierovej forme bude kód fungovať po odфотографовaní. Náš súbor s QR kódom sa nazýva `qr1.pdf`.

```
\href{https://frcatel.fri.uniza.sk/users/beerb}{
  \includegraphics[width=3em]{qr1.pdf}}
```

Výstup predchádzajúceho príkazu bude mať nasledujúci tvar



2. Tvorba 2D animácií

Ak chceme animovať nejaký dvojrozmerný pohyb, musíme ho rozčleniť, analogicky ako na filmovom plátne, na jednotlivé okienka. V praxi to znamená, že pre každú časť pohybu musíme vytvoriť samostatný záber, t. j. samostatný obrázok. Tieto obrázky potom pomocou balíčka `animate` uvedieme do pohybu. Potom môžeme tento pohyb posúvať smerom dopredu alebo dozadu, spomaľovať alebo zrýchľovať, prípadne zastaviť, skočiť priamo na začiatok alebo koniec pohybu ap. Pre tento účel je vhodné do animácie aktivovať riadiace menu.

Postup vytvorenia animácie ukážeme na jednoduchom grafe, ktorý ilustruje prechod hyperkocky medzi jednotlivými dimenziami. Obrázky vytvoríme pomocou balíčka `TikZ`. Uvedený súbor `graf1.pdf` s jednotlivými obrázkami k tejto animácii nájde čitateľ na adrese <https://frcatel.fri.uniza.sk/ossconf/zborniky-online/ossconf2024/files/graf1.pdf>



Vygenerovaný zdrojový súbor s obrázkami je síce vo formáte `.pdf`, ale my ho potrebujeme konvertovať do formátu `.svg`, aby sme ho mohli zobrazovať vo webovom prehliadači. Na to existuje v distribúcii `TeXlive` program `dvisvgm`, ale ten konvertuje zo základného výstupného formátu L^AT_EX-u a to formátu `.dvi`. Tento formát v dnešnej dobe už mládež nepozná, ale starí kozáci a nadšenci áno. Generujeme ho namiesto príkazu `pdflatex` príkazom `latex`.

Obrázky zo súboru `graf1.pdf` animujeme pomocou prostredia `animateinline` v novom súbore, napr. `graf1-web.tex`, následne preložíme do `graf1-web.dvi` a konvertujeme do súboru `graf1-web.svg`. Pred samotnou animáciou potrebujeme vedieť počet obrázkov, ktoré animujeme, aby sme mohli samotný proces automatizovať. V našom prípade uvedený súbor `graf1.pdf` obsahuje 72 obrázkov a súbor `graf1-web.tex` má tvar:

```
\documentclass[dvisvgm]{standalone}
\usepackage{graphicx}
\usepackage{animate}

\begin{document}
\begin{animateinline}[autoplay, poster=first, controls, loop,
                    palindrome, buttonsiz=14pt]{4}
  \multiframe{72}{ir=1+1}{
    \includegraphics[width=10cm,page=\ir]{graf1.pdf}}
\end{animateinline}
\end{document}
```

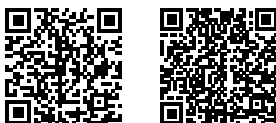
Parametre prostredia `animateinline` znamenajú, že sa animácia spusti automaticky od prvého obrázka a potrvá nepretržite (pokiaľ ju nezastavíme) od začiatku do konca a späť, `controls` znamená riadiace menu, jeho veľkosť bude 14pt. Posledný (povinný) parameter 4 definuje 4 obrázky za sekundu. Hodnota 0.25 by znamenala jeden obrázok za 4 sekundy.

Pre vytvorenie súboru `graf1-web.dvi` použijeme 2-krát príkaz `latex` a následne príkaz `dvisvgm` na konverziu do súboru `graf1-web.svg`.

```
latex graf1-web.tex
latex graf1-web.tex
dvisvgm --font-format=woff --exact --zoom=-1
--page=- graf1-web.dvi
```

Súbor `graf1-web.svg` môžeme priamo spustiť vo webovom prehliadači alebo ho zapúzdriť do `.html` súboru, v ktorom môžeme využiť `javascript` servera `cdn.mathjax.org` na sadzbu online matematiky pomocou $\text{\LaTeX}$ -u. Do hlavičky `.html` súboru treba pridať nasledovné riadky.

```
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE> obrazok </TITLE>
<script type="text/javascript"
src="https://cdn.mathjax.org/mathjax/latest/
      MathJax.js?config=TeX-AMS-MML_HTMLorMML">
MathJax.Hub.Config({tex2jax: {inlineMath: [ ['$','$'], ["\\(", "\\)"] ],
displayMath: [ ['$$','$$'], ['\[','\]' ] ]}});
</script>
</HEAD>
```



Ak chceme súbor animovať do `.pdf` súboru, použijeme priamo na danom mieste v texte už spomínaný príkaz

```
\begin{animateinline}[autoplay, poster=last, controls, loop,
                      palindrome, buttons=14pt]{4}
  \multiframe{72}{ir=1+1}{
    \includegraphics[width=.5\linewidth,page=\ir]{graf1.pdf}
  }
\end{animateinline}
```

a dostaneme nasledujúci animovaný obrázok.



3. Tvorba 3D animácií

3D pohyblivé obrázky môžeme vytvoriť viacerými spôsobmi. V tomto príspevku ukážeme ako vytvoriť 3D grafy dvojrozmerných funkcií, ktoré môžeme otáčať, zmenšovať alebo zväčšovať prípadne posúvať. Veľmi vhodným nástrojom pre tvorbu takýchto grafov je Open Source program **Asymptote**, ktorý je priamo integrovaný do prostredia L^AT_EX-u. Bližšie informácie a množstvo príkladov vrátane dvoj aj trojrozmerných animácií aj so zdrojovými súbormi nájde čitateľ na jeho domovskej stránke <https://asymptote.sourceforge.io/>. Po vložení príkazu

```
\usepackage[inline]{asymptote}
```

do preamble L^AT_EX-ového zdrojového dokumentu môžeme **Asymptote** scripty písať priamo do textu do prostredia **asy**:

```
\begin{asy}
    zdrojový text pre Asymptote
\end{asy}
```

pričom za príkazom `\begin{asy}` nesmie byť prázdny riadok. Príkaz `\end{asy}` musí byť na samostatnom riadku a musí byť na riadku jediný (ani poznámka pomocou `%` nemôže byť za ním). L^AT_EX dokument sa prekladá na niekoľkokrát [3], preto je vhodné si vytvoriť script na takýto preklad. Napríklad, ak sa náš zdrojový súbor nazýva **subor.tex**, potom preklad by mal prebiehať nasledovne:

```
pdflatex subor.tex
asy subor-*.asy
pdflatex subor.tex
```

Obrázok môžeme samozrejme vytvoriť aj v inom súbore a pomocou balíčka **media9** vložiť do pôvodného súboru, pričom sa zachová jeho pohyblivosť. Takto sme vytvorili súbor **graf2.pdf** s grafom funkcie $z = x \cdot y \cdot (x^2 - y^2) / (x^2 + y^2)$, ktorý

čitateľ nájde na adrese <https://frcatel.fri.uniza.sk/ossconf/zborniky-online/ossconf2024/files/graf2.pdf>.



Pri preklade do pdf vyzerá preambula `graf2.tex` nasledovne:

```
\documentclass{standalone}
\usepackage[inline]{asyptote}
```

Pri preklade do html je potrebné do preambuly súbora `graf2-web.tex` vložiť:

```
\begin{asydef}
settings.outformat="html";
settings.inlineimage=false;
settings.embed=false;
\end{asydef}
```

a súbor prekladať s parametrom `-v`, t. j.

```
pdflatex graf2-web.tex
asy -v graf2-web-*.asy
```

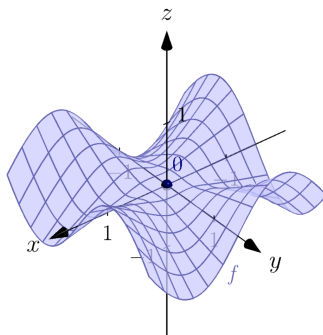
pričom dostaneme súbor `graf2-web-1.html`, ktorý môžeme spustiť priamo v prehliadači.



Vygenerovaný pohyblivý súbor `graf2.pdf` vložíme do textu pomocou príkazu `\includemedia` nasledovne

```
\includemedia[add3Djscrip=asylabls.js,
add3Djscrip=3Dspintool.js,activate=onclick,noplaybutton,
3Dc2c=3.75 1.75,3Dcoo= 0 0 0,3Droo=150, 3Dortho=.0125]{
\includegraphics[height=.4\textwidth]{graf2.pdf}}{graf2-1+0.prc}
```

a dostaneme pohyblivý obrázok (samozrejme nie na papieri), ktorý môžeme ovládať pomocou kurzora myšky.



Podakovanie. Príspevok vznikol s príspevím grantu KEGA 019ŽU-4/2023 „Inovatívne učenie matematiky s podporou Open Source“ podporeného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] BRINGHURST, R.: *The Elements of Typographic Style*, Hartley Marks Publishers 2004, ISBN 0-88179-205-5.
- [2] BLAŠKO, R.: *Sadzba QR kódov*, Zborník príspevkov OSSConf 2015, 1.–3. júla 2015, Žilina, ISBN 978-80-970457-7-7.
- [3] BLAŠKO, R.: *Asymptote a L^AT_EX*, Zborník príspevkov OSSConf 2018, 2.–4. júla 2018, Žilina, ISBN 78-80-554-1627-4.

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,
E-mailová adresa: `beerb@frcatel.fri.uniza.sk`