

ASYMPTOTE A $\LaTeX$ ASYMPTOTE AND $\LaTeX$

Rudolf Blaško

Adresa: Katedra mat. metod a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko

E-mail: beerb@frcatel.fri.uniza.sk

Abstrakt: $\TeX$, resp. $\LaTeX$ patria medzi najlepšie DTP programy na svete a v prípade odborných textov (matematických, fyzikálnych, chemických ap.) existuje len veľmi málo programov (aj to iba komerčných), ktoré by im mohli konkurovať. Kancelárske programy Office nemôžu $\TeX$ -u konkurovať ani v prípade hladkej sadzby. Aj nakresliť graf matematickej funkcie nie je problém. Existuje mnoho programov, ktoré dokážu grafy pomerne pekne vykresliť. Problém nastáva s exportom a následným použitím v inom dokumente. Problém býva hlavne so sprievodným textom, t. j. použitým písmom. $\LaTeX$ má k dispozícii už niekoľko rokov veľmi silný prostriedok – balíček *TikZ*. Skúsenejší pisateľ dokáže pomocou balíčka *TikZ* nakresliť prakticky všetko od jednoduchých obrázkov, cez grafy funkcií až po zložité schémy. Iná možnosť pri písaní dokumentov v $\TeX$ -u, resp. $\LaTeX$ -u je použitie programu *Asymptote*, ktorý je súčasťou distribúcie $\TeX$ Live. Nevýhodou je nutnosť viacnásobného prekladu dokumentu (vrátane *Asymptote*). Nepopierateľná pridaná hodnota (pri použití *TikZ*-u aj *Asymptote*) ostáva v možnosti použitia všetkých dostupných prostriedkov $\LaTeX$ -u. V spojení s balíčkom *animate* môžeme grafy a obrázky vytvorené pomocou *TikZ*-u a *Asymptote* jednoducho animovať.

Kľúčová slová: $\TeX$, $\LaTeX$, balíček *TikZ*, *Asymptote*, balíček *animate*, reálna funkcia, 3D funkcia.

Abstract: $\TeX$, respectively, $\LaTeX$ belong to the best desktop publishing programs in the world and what is concerning the scientific texts (mathematical, physical, chemical, etc.), there are very few programs (even commercial ones) that could compete with them. The “Office type” programs cannot compete with $\TeX$ even with smooth text typeset. Even drawing a graph of a mathematical function is not a problem. There are many programs that can render graphs quite nicely. The problem arises with the export and subsequent use in another document. The problem is especially with the accompanying text, i.e. font used. $\LaTeX$ contains a very strong tool – the *TikZ* package – for several years now. The experienced writer can draw almost everything from simple pictures, function graphs to complex diagrams

using the `TikZ` package. Another option when writing documents in $\text{\TeX}$, respectively, $\text{\LaTeX}$ is the `Asymptote` program included with $\text{\TeX}$ Live. The disadvantage is the necessity of multiple translations of the document (if it includes `Asymptote`). The undeniable added value (using both `TikZ` and `Asymptote`) remains in the use of all available $\text{\LaTeX}$ tools. In conjunction with the `animate` package, you can easily animate graphs and images.

Keywords: $\text{\TeX}$, $\text{\LaTeX}$, package `TikZ`, `Asymptote`, package `animate`, real function, 3D function.

Úvod

$\text{\TeX}$ [3], resp. $\text{\LaTeX}$ patria medzi najlepšie DTP programy na svete a v prípade odborných textov (matematických, fyzikálnych, chemických, technických ap.), kde sa vyžadujú špeciálne symboly alebo dokonca zložitejšie vzťahy, existuje len veľmi málo programov (aj to iba komerčných), ktoré by im mohli konkurovať. V uvedených odborných kruhoch sú samozrejme aj najčastejšie používané. Ale to sa žiaľ, netýka Slovenska a Česka. Aj keď sú $\text{\TeX}$ a $\text{\LaTeX}$ zadarmo (Open Source), mnoho autorov uprednostní komerčný Microsoft Office, v lepšom prípade Libre Office. Tieto programy sú výborné hlavne pre sekretárky, prípadne pre hladkú sadzbu. Pre vedecké a odborné texty (matematické, technické, chemické ap.) sú nevhodné a nepraktické. Ak chce niekto písať matematické vzorce a rovnice pomocou týchto kancelárskych programov, dokáže to, ale samozrejme s neporovnateľne väčším úsilím a neporovnateľne nižšou kvalitou.

Ak píšeme odborný text, mnohokrát potrebujeme svoje výsledky ilustrovať grafmi nejakých matematických funkcií. Sú to nielen funkcie jednorozmerné (ich grafy sú 2D), ale aj viacrozmerné s grafmi zobrazenými v 3D priestore. Nakresliť graf takejto matematickej funkcie nie je problém. Existuje mnoho programov (napr. komerčné programy **Wolfram Mathematica**, **Maple**, **Matlab**, Open Source **Maxima**, resp. **wxMaxima**, prípadne on-line **WolframAlpha**), ktoré dokážu grafy pomerne pekne vykresliť. Problém nastáva s exportom a následným použitím v inom dokumente. Statické grafy sa exportujú pomerne jednoducho, oveľa ťažšie je exportovať animované grafy. V prípade 2D grafov môžeme v uvedenom programe vytvoriť animovaný súbor vo formáte `.gif` a následne ho importovať do $\text{\TeX}$ -ovského dokumentu. Oveľa výhodnejšie je v tomto prípade vytvoriť pre každý relevantný záber jeden grafický súbor (najlepšie vo formáte `.pdf`, `.png` alebo `.jpg`) a tie následne spojiť do jediného animovaného obrázka pomocou balíčka `animate`. Takto vytvorené obrázky môžeme pomocou jednoduchého menu ovládať (meniť rýchlosť prehrávania, prehrávať od začiatku do konca, pozastaviť prehrávanie ap.).

Balíček `animate` velmi dobře spolupracuje s obrázkami vytvorenými pomocou balíčka `TikZ` a taktiež s obrázkami vytvorenými v programe `Asymptote`. Navyše aj 3D objekty vytvorené pomocou `Asymptote` môžeme tiež animovať (meniť veľkosť, otáčať do rôznych smerov ap.).

1. Asymptote

Program `Asymptote: The Vector Graphics Language` vyvinuli Andy Hammerlindl, John C. Bowman (University of Alberta) a Tom Prince [2, 1]. `Asymptote` je Open Source prostriedok, ktorý je k dispozícii na základe licencie GNU Lesser General Public License (LGPL). `Asymptote` funguje samozrejme nielen pod Linux-om, ale aj pod Mac OS a Microsoft Windows. Je vhodný na kreslenie grafov matematických funkcií a to dvojrozmerných a aj trojrozmerných.

Na domovskej stránke asymptote.sourceforge.io nájdeme mnoho užitočných informácií (dokumentácia, zdrojové a inštalačné súbory, galérie 2D a 3D príkladov, index, tutoriál, fórum, odkazy na iné stránky, ...).

Môžeme si z tejto stránky stiahnuť inštalačné súbory a `Asymptote` nainštalovať na svoj počítač. `Asymptote` výstižne charakterizuje hneď prvý odstavec na tejto stránke: „`Asymptote` je výkonný popisný jazyk vektorovej grafiky, ktorý poskytuje prirodzené prostredie pre kreslenie technických obrázkov a schém. Popisky a rovnice sú vysádzané pomocou $\text{\TeX}$ -u, resp. $\text{\LaTeX}$ -u, de facto štandardu pre sadzbu matematiky.“ Hlavnou výhodou `Asymptote` oproti iným grafickým programom je, že je navyše aj programovacím jazykom. Je inšpirovaný `METAPOST`-om, syntax jazyka je odvodená zo syntaxe `C++`. Grafické príkazy sú implementované v samotnom jazyku `Asymptote`, čo umožňuje ich prispôbenie konkrétnym situáciám. Spomeňme ešte niektoré zaujímavé vlastnosti `Asymptote`, viď <https://asymptote.sourceforge.io/right.html>:

- Generovanie vysoko kvalitnej vektorovej grafiky s možnosťami výstupu do formátov PostScript (`.ps`, resp. `.eps`), `.pdf`, `.svg`, WebGL a PRC.
- Možnosť vkladania 3D vektorovej grafiky WebGL (Web Graphics Library pre JavaScript) do súborov `.html`.
- Možnosť vkladania 3D vektorovej grafiky PRC (Product Representation Compact) do súborov `.pdf`.

Ak máme nainštalovaný $\text{\TeX}$ a neplánujeme `Asymptote` samostatne používať, potom ho nemusíme samostatne inštalovať, pretože je štandardnou súčasťou inštalácie $\text{\TeX}$ -u a veľmi dobre spolupracuje s $\text{\LaTeX}$ -om pomocou balíčka `asymptote`. Pre nás je zaujímavé, že 3D obrázky vytvorené pomocou

Asymptote a exportované do súborov **.pdf**, môžeme rôzne otáčať, zmenšovať ap. Dvojmerné obrázky dokážeme animovať pomocou balíčka **animate**. Balíčkom **animate** sa zaoberal P. Stríž na OSSConf 2010 v príspevku „Jak jsem se skamarádil s L^AT_EXovým balíčkem **animate**“ [5], pričom autor obrázky vytváral na úrovni METAPOST-u.

2. Asymptote a T_EX

Najväčšia výhoda **Asymptote** je už spomínaná spolupráca a prepojenie s T_EX-ovým, resp. L^AT_EX-ovým dokumentom. Po vložení príkazu

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

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

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

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

```
$ lualatex subor.tex
$ asy subor-*.asy
$ lualatex subor.tex
```

Prvý krok, preklad pomocou **lualatex-u**, vytvorí v domovskom adresári súboru **subor.tex** súbory **subor-1.asy**, **subor-2.asy**, ... (ku každému prostrediu **asy** jeden súbor) a súbor **subor.pre**.

Druhý krok, preklad pomocou **Asymptote**, vytvorí v uvedenom adresári súbory **subor-1.tex**, **subor-2.tex**, ..., **subor-1.pre**, **subor-2.pre**, ..., a pomocné súbory **_subor-1.pdf**, **_subor-2.pdf**, ..., a tiež **subor-1_0.pdf**, **subor-2_0.pdf**, ... Pri vytváraní 3D grafiky sa vygenerujú aj iné súbory, napr. **subor-?+0.prc**, **subor-?+0_0.pdf**, **subor-?_0.pbsdat** a **subor.out**.

Po druhom kroku je súbor **subor.pdf** nefunkčný, preto musíme urobiť ešte jeden preklad pomocou **lualatex-u**. Ak sa pri písaní dokumentu údaje v prostrediach potrebných pre **Asymptote** nezmenia, preklad pomocou **asy** nemusíme robiť a postačí jeden preklad pomocou **lualatex-u**.

Externý súbor **asysubor.asy** so zdrojovým kódom **Asymptote** môžeme do L^AT_EX-ovského dokumentu vložiť pomocou príkazu

```
\asyinclude[<volby>]{asysubor.asy}
```

Iná možnosť ako používať L^AT_EX s implementovanými zdrojovými kódmi Asymptote je použitie utility latexmk zo stránky <http://mirror.ctan.org/support/latexmk/>. Dostupné aj v T_EXLive.

Nastavenia potrebné pre Asymptote, ktoré používame vo viacerých prostrediach asy, nemusíme písať samostatne v každom z nich. Môžeme ich definovať na začiatku (môžeme aj v preamble) v prostredí asydef, napr.

```
\begin{asydef}
  texpreamble("\usepackage[math]{iwona}");
  real cc=1.5; // hodnota cc > 0
  import graph;
\end{asydef}
```

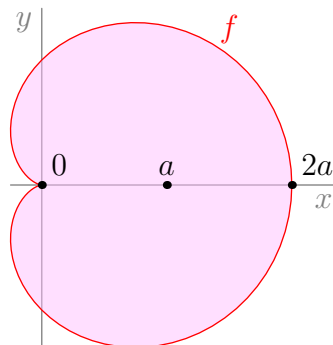
Príkaz texpreamble("\usepackage[math]{iwona}"); zariadi, že v celom dokumente, vrátane prostredí asy nastaví sanserifový font iwona (viď <https://tug.org/FontCatalogue/iwona/>). Ak tento príkaz použijeme iba v nejakom konkrétnom prostredí asy, tieto fonty sa nastavia v celom dokumente a iba v tomto prostredí asy a v ostatných prostrediach asy sa nenastavia. Ak dáme do preamble príkaz \usepackage[math]{iwona} fonty sa nastavia iba v celom dokumente, ale nie v prostrediach asy.

Príkaz import graph; načíta modul graph na kreslenie grafov reálnych funkcií. Týchto modulov (Base modules) je okolo 40 (<https://asymptote.sourceforge.io/doc/>, resp. súbor asymptote.pdf, ktorý dostaneme zadáním texdoc asymptote na príkazovej úrovni – v oboch prípadoch „Kapitola 8: Base modules“) a do prostredí asy sa načítavajú príkazom import. Základný modul je plain, ktorý je nastavený implicitne. Na animovanie dvojrozmerných obrázkov používame modul animation. Na vykresľovanie trojrozmerných grafov používame modul graph3. Z ostatných modulov spomeňme napríklad interpolate (modul implementuje Lagrangeovú, Hermitovú a štandardnú interpoláciu kubických spline do Asymptote), geometry (rôzne geometrické symboly, štruktúry, krivky ap.), stats (rôzne štatistické funkcie), patterns (rôzne vzory na vyplňanie plôch), markers (rôzne možnosti na označovanie vrcholov, hrán, ciest, uhlov ap.), tree, binarytree, drawtreetree (binárne a jednoduché stromy), palette (rôzne nastavena farieb), three (možnosť konverzie dvojrozmerných kriviek na trojrozmerné a naopak), grid3 (trojrozmerné mriežky), flowchart (vývojové diagramy) a contour (vrstevnice).

Rovnica krivky zvanej kardioida (srdcovka) má v polárnych súradniciach rovnicu $\rho(\varphi) = a(1 + \cos \varphi)$, $\varphi \in \langle 0, 2\pi \rangle$, $a > 0$. Táto krivka sa dá taktiež zostrojiť ako epicykloida, t. j. rovinná krivka, ktorú pri kotúľaní kružnice k

po vonkajšej strane pevne danej kružnice K opisuje daný bod B , ktorý je pevný vzhľadom na k . Kardioidu dostaneme, ak majú obe kružnice rovnaký polomer a bod B leží na obvodě kružnice k . Na obr. 1 je kardioida vykreslená pomocou Asymptote.

```
\documentclass{article}
\usepackage[inline]{asymptote}
\begin{document}
\pagestyle{empty}
\begin{asy}
import graph;size(0,5cm);
xaxis("$x$",above=true,p=gray);
yaxis("$y$",above=true,p=gray);
real f(real t){return 1+cos(t);}
path g=polargraph(f,0,2pi)--cycle;
filldraw(g,pink+opacity(.5),drawpen=red);
label("\large$f$", (1.5,1.25),red);
dot("$0$", (0,0),N+E);
dot("$a$", (1,0),N);dot("$2a$", (2,0),N+E);
\end{asy}
\end{document}
```



Obr. 1: Kardioida $f: \rho(\varphi) = a(1 + \cos \varphi)$, $\varphi \in \langle 0, 2\pi \rangle$, $a > 0$

Modul **animation** umožňuje generovať animované obrázky. Tieto animácie využívajú prevodný program **ImageMagick** na zlúčenie viacerých obrázkov do súborov vo formáte **.gif** alebo **.mpeg**. Vhodnejšie je použiť odvodený modul **animate**, ktorý generuje vyššiu kvalitu obrázkov vo formáte **.pdf** s možnosťami ovládania. Tento spôsob je podmienený inštaláciou balíčka **animate** v **L^AT_EX**-u a použiť ho môžeme v prostredí **asy** nasledovne:

```
usepackage("animate");
settings.tex="lualatex";
defaultpen(.25);
import animation;
```

Ak animovanú časť označíme **animation a**, potom ovládacie menu animácie dostaneme napríklad príkazom

```
label(a.pdf(delay=250, "buttonsize=10pt, controls, autoplay,
loop, palindrome", multipage=false));
```

Hypocykloida je rovinná krivka, ktorú pri kotúľaní kružnice k po obvodě vnútornej strany pevne danej kružnice K (kružnica k je vo vnútri kružnice K) opisuje daný bod B , ktorý je pevný vzhľadom na k . Ak označíme $R > 0$ polomer kružnice K , $r > 0$ polomer kružnice k ($r < R$), $c \geq 0$ vzdialenosť bodu B od stredu kružnice k , potom môžeme hypocykloidu parametricky vyjadriť v tvare

$$x = (R-r) \cos \frac{rt}{R} + c \cos \frac{(R-r)t}{R}, \quad y = (R-r) \sin \frac{rt}{R} - c \sin \frac{(R-r)t}{R}, \quad t \in R.$$

Pohyb bodu B pre prípad $R = \frac{5r}{3}$, $c = \frac{3r}{2}$ je znázornený na obr. 2. Keďže sú pomery $\frac{r}{R} = \frac{3}{5}$, $\frac{(R-r)}{R} = \frac{2}{5}$ racionálne čísla, funkcie sínus a kosínus sú periodické s periódou 2π , pohyb sa uzavrie po piatich okružoch, t. j. po hodnote $t = 10\pi$. Uvedená hypocykloida je periodická s periódou 10π a má tvar

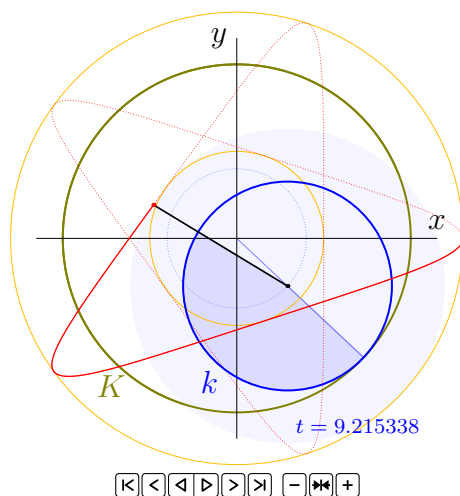
$$x = \frac{2r}{3} \cos \frac{3t}{5} + \frac{3r}{2} \cos \frac{2t}{5}, \quad y = \frac{2r}{3} \sin \frac{3t}{5} - \frac{3r}{2} \sin \frac{2t}{5}, \quad t \in (0, 10\pi).$$

Na začiatku zdrojového kódu sú definované základné hodnoty $c = cc$, $R = rv$, $r = rm$, rt koniec definičného intervalu, n počet vykresľovaných bodov grafu.

```
\begin{asy}
real cc=1.5,u=5,v=3,rv=u/v,rm=1,rt=2*u,rp=rv-rm;int n=90;
import graph;
usepackage("animate");settings.tex="lualatex";defaultpen(.25);import animation;
size(0cm,5cm);
pair wheelpoint(real t)
{return (rp*cos(t*rm/rv)+cc*cos(rp*t/rv),rp*sin(t*rm/rv)-cc*sin(rp*t/rv));}
guide wheel(guide g=nullpath,real a,real b,int n)
{real width=(b-a)/n;for(int i=0;i<n;++i)
{real t=a+width*i;g=g--wheelpoint(t);} return g;}
real tinterval=rt*pi,t1=0,t2=t1+tinterval;draw(circle((0,0),rv),olive+.75);
animation a;
pair z1=wheelpoint(t1);dot(z1,red);real dt=(t2-t1)/n;
for(int i=0;i<n;++i) {save();
real t=t1+dt*i;real kx=rp*cos(rm*t/rv);real ky=rp*sin(rm*t/rv);
filldraw(circle((kx,ky),cc),.2paleblue+white,.2paleblue+white+.5);
draw((0,0)--(rv*cos(rm*t/rv),rv*sin(rm*t/rv)),lightblue);
if (t>0) {filldraw((kx,ky)--arc((kx,ky),rm,180*rm*t/rv/pi,
-180*rp*t/rv/pi)--cycle,white+.75blue+opacity(.25),drawpen=lightblue);}
draw(circle((0,0),rv),olive+.75);label("$K$",(-.6*rv,-.75*rv),SW,olive);
draw(circle((0,0),rp),dotted+blue+white);
draw(circle((0,0),rp-cc),yellow+.35red);draw(circle((0,0),rp+cc),yellow+.35red);
label("$x$", (rv+.25,0),N);draw((-rv-.25,0)--(rv+.25,0));
label("$y$", (0,rv+.25),W);draw((0,-rv-.25)--(0,rv+.25));
draw(wheel(t1,t2,8*n),dotted+red);draw(circle((kx,ky),rm),blue+.75);
label("$k$", (kx-.6,ky-.75),SW,blue);draw((kx,ky)--wheelpoint(t),black+.625);
dot((kx,ky));dot(wheelpoint(t),red+black);draw(wheel(t1,t,8*max(1,i)),red+.5);
label("\scriptsize$t="+string(t,7)+"$", (.3*rv,-rv),SE,blue);
a.add();restore();} erase();
label(a.pdf(delay=250,"buttonsize=10pt,controls,autoplay,loop,
palindrome",multipage=false));
\end{asy}
```

3. Asymptote a 3D obrázky

Vytvoríť 3D obrázok v programe *Asymptote* je pomerne jednoduché. Výsledný obrázok je porovnateľný s 3D obrázkami vytvorenými v komerčných programoch Wolfram Mathematica a Maple. K používaniu *Asymptote* ma priviedla nutnosť. V pripravovanej publikácii z matematickej analýzy funkcií



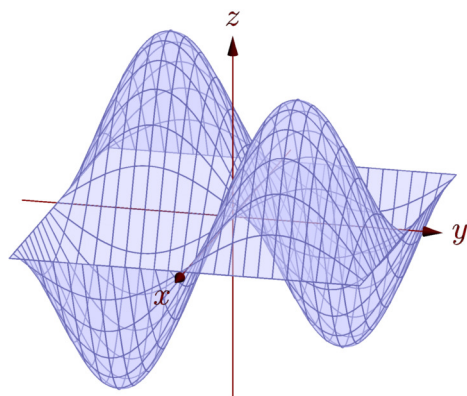
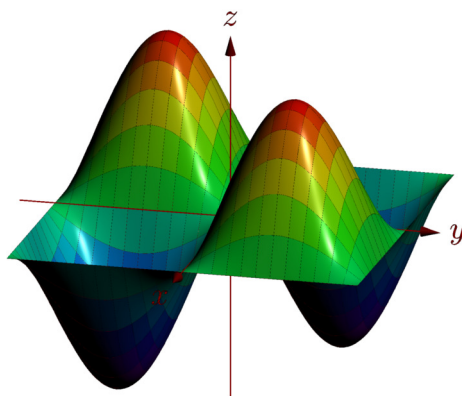
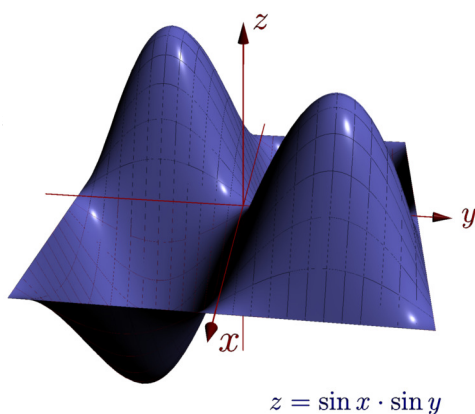
Obr. 2: Příklad animácie hypocykloidy

viacerých premenných som potreboval vykresľovať grafy funkcií dvoch premenných. Jedna z možností je použiť balíček `TikZ` — ďalší výkonný prostriedok na vektorovú grafiku implementovaný do `TeX`-u. Problém je, že v `TikZ`-e nakreslíme prekrásne 2D a aj 3D obrázky (grafy funkcií, schémy, štatistické grafy, chemické grafy atď.), ale 3D obrázky nedokážeme interaktívne otáčať, zmenšovať, pohybovať. Druhá možnosť bola použiť obrázky z komerčného programu Wolfram Mathematica, ktorého licenciu mám v legálnom držaní. Z tohto programu som nedokázal exportovať obrázky akceptovateľné v `TeX`-u, t. j. `.pdf` formáte. Ešte som mohol použiť Open Source program `wxMaxima`, ale v tomto prípade je situácia obdobná. Pre nasledujúce obrázky sú definované nové farby v prostredí `asydef`.

```
\begin{asydef}
  pen blackIV=black*0.40+white*0.60;
  pen blackII=black*0.20+white*0.80;
  pen PPblueblack=blue*0.50+black*0.50;
  pen PPoliveblack=heavygreen*0.50+black*0.50;
  pen PPblueblackI =PPblueblack*0.60+white*0.40;
  pen blueII=blue*0.20+white*0.80;
  pen blueV=blue*0.50+white*0.50;
\end{asydef}
```

Pri tvorbe grafov pomocou `Asymptote` sa štandardne definuje transparentné pozadie až na 3D grafy. S načítaním modulu `graph3` sa nastaví biele pozadie obrázku, dajú sa nastaviť rôzne farby až na transparentné pozadie. Na úvod uvedieme príklad trojrozmerného grafu vytvoreného v `Asymptote`.

```
\begin{asy}
import graph3;import palette;
size(175,175,IgnoreAspect);
currentprojection=perspective(2,.55,0.25);
real f(pair z){return sin(z.x)*sin(z.y);}
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,
    Spline),lightblue,render(merge=true))
surface f=surface(f,(-pi,-pi),(pi,pi),
    24,Spline);
draw(f,mean(palette(f.map(zpart),
    Rainbow(24))),brown);
draw(Label("$x$",1),
    (-1.2pi,0,0)--
    (1.2pi,0,0),brown,Arrow3);
draw(Label("$y$",1),
    (0,-1.2pi,0)--
    (0,1.2pi,0),brown,Arrow3);
draw(Label("$z$",1),
    (0,0,-1.2)--(0,0,1.2),brown,Arrow3);
label("$z=\sin x \cdot \sin y$",
    (.5,.5,-1.35),SE,deepblue);
\end{asy}
```



Obr. 3: Graf funkcie $z = \sin x \cdot \sin y$, $(x, y) \in \langle -\pi, \pi \rangle \times \langle -\pi, \pi \rangle$

Na obr. 3 je znázornený v karteziánskych súradniciach graf funkcie $z = \sin x \cdot \sin y$ na intervale $\langle -\pi, \pi \rangle \times \langle -\pi, \pi \rangle$. Uvedené sú tri grafické modifikácie tohto obrázku. Prvý podobrázok (vpravo hore) zodpovedá napísanému zdrojovému kódu, na ostatných dvoch je zmenená projekcia

```
currentprojection=perspective(2,.55,0.25);
na currentprojection=orthographic(2,.55,.25);
```

Na druhom podobrázku (vľavo dole) je v riadku

```
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,Spline),
        lightblue,render(merge=true));
```

nahradená farba `lightblue` farbou `blue+opacity(.75)`, t. j.

```
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,Spline),
        blue+opacity(.75),render(merge=true));
```

a na poslednom podobrázku je uvedený riadok a nasledujúce dva riadky

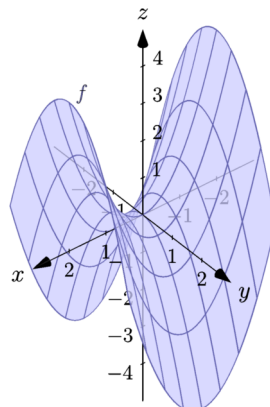
```
surface f=surface(f,(-pi,-pi),(pi,pi),24,Spline);
draw(f,mean(palette(f.map(zpart),Rainbow(24))),brown);
```

nahradené jediným riadkom

```
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,Spline),
        blue!+opacity(.5),meshpen=PPblueblackI+thick(),
        nolight,render(merge=true));
```

Na obr. 4 je graf kvadratickej formy $f(x, y) = x^2 - y^2$, $(x, y) \in R^2$ znázornený na intervale $\langle -2, 2 \rangle \times \langle -2, 2 \rangle$.

```
\begin{asy}
import graph3;
texpreamble("\usepackage{amsmath}");
size(3.5cm,0);
currentprojection=orthographic(2,2.5,2.5);
real f(pair z){return z.x^2-z.y^2;}
surface s=surface(f,(-2,-2),(2,2),nx=8,Spline);
draw(s,blue!+opacity(.75),meshpen=PPblueblackI+thick(),
        nolight,render(merge=true));
label(scale(.65)*"$f$",2X+4Z,-X,PPblueblackI);
draw(Label("$f$",position=EndPoint),(.05,0,4)--(-.05,0,4));
// ...
draw(Label("$x$",position=EndPoint),(-3,0,0)--
(3,0,0),Arrow3);
draw(Label("$y$",position=EndPoint),(0,-3,0)--
(0,3,0),Arrow3);
draw(Label("$z$",position=EndPoint),(0,0,-5)--
(0,0,5),Arrow3);
\end{asy}
```



Obr. 4: Graf funkcie $f(x, y) = x^2 - y^2$, $(x, y) \in R^2$

Na obr. 5 je graf dvojrozmernej funkcie

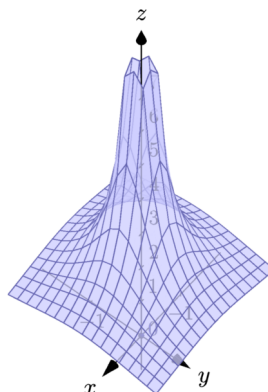
$$f(x, y) = \frac{\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2} + 1}, \quad (x, y) \in R^2 - \{(0, 0)\}.$$

Obrázok ilustruje limitný prechod $f(x, y) \rightarrow \infty$ pre $(x, y) \rightarrow (0, 0)$. Funkčné hodnoty funkcie f v okolí bodu $(0, 0)$ s polomerom $r = 0.02$ sú z grafu

```

\begin{asy}
import graph3;size(3.5cm,0);
currentprojection=orthographic(1.25,1,2.5);
real f(pair z){real r=z.x^2+z.y^2; return r>=0.02
? min(8,sqrt(z.x^2+z.y^2)/
(sqrt(z.x^2+z.y^2+1)-1)): 0;}
bool allow(pair z){return f(z)>0;}
surface s=surface(f,(-1.5,-1.5),(1.5,1.5),nx=16,allow);
draw(s,blue!75,opacity(.75),meshpen=PPblueblack!thick(),
nolight,render(merge=true));
draw(Label("$z$",position=EndPoint),(0,0,7)--(0,0,7));
// ...
draw(Label("$x$",position=EndPoint),(-2,0,0)--
(1,0,0),Arrow3);
draw(Label("$y$",position=EndPoint),(0,-2,0)--
(0,1,0),Arrow3);
draw(Label("$z$",position=EndPoint),(0,0,-1)--
(0,0,9),Arrow3);
draw(scale(.05,.05,.05)*unitsphere,PPblueblack!1);
label(scale(.75)*"$0$",.25Z+.25Z,Y,PPblueblack);
\end{asy}

```



Obr. 5: Graf funkcie $f(x, y) = \frac{\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2 + 1} - 1}$, $(x, y) \in \mathbb{R}^2 - \{(0, 0)\}$ v okolí bodu $(0, 0)$

vyrezané. Keďže graf vykresľuje iba 16 vrstevníc rovnobežných (teda ich priemetov do roviny xy) s osou x a 16 vrstevníc rovnobežných s osou y hrany okolia tvoria uzavretú lineárnu lomenú čiaru. Pri väčšom počte vrstevníc by sa graf „vylepšil“, ale aj zbytočne predĺžil preklad. Ale výpovedná hodnota obrázku by sa prakticky nezlepšila.

Graf dvojrozmernej funkcie $f(x, y) = x^2 - y^2 + 3$, $(x, y) \in \mathbb{R}^2$ (obr. 6) je znázornený pomocou polárnych súradníc. Definičný obor, t. j. nezávislé premenné x a y sú vyjadrené pomocou vzťahov $x = \rho \cos \varphi$, $y = \rho \sin \varphi$ a obraz pomocou vzťahu $f(x, y) = (\rho \cos \varphi)^2 - (\rho \sin \varphi)^2 + 3$. V našom prípade to je časť príkazu

```

return ((xx)*cos(t.y),(xx)*sin(t.y),
((xx)*cos(t.y))^2-((xx)*sin(t.y))^2+3);

```

Súčasťou grafu je krivka **p1** spájajúca maximálnu hodnotu a minimálnu hodnotu zobrazeného grafu a jej priemet do roviny xy . Obe sú definované parametricky pomocou trojice bodov **x(real t)**, **y(real t)**, **w(real t)**, resp. **x(real t)**, **y(real t)**, **z(real t)**.

Na záver uvedieme na ilustráciu ešte tri priestorové grafy vytvorené pomocou programu *Asymptote*. Na prvom (obr. 7) je znázornená priestorová špirála f zadaná v parametrickom tvare a dotyková priamka v jej bode.

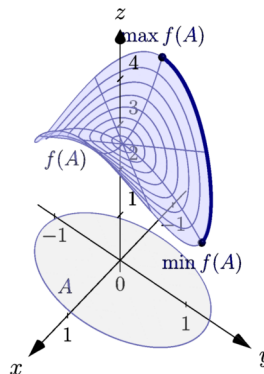
```

\begin{asy}
import graph3;size(3.5cm,0);
currentprojection=orthographic(1.25,1,2.5);
draw(shift(.2X+.2Y)*surface(scale(.75,1.2)*unitcircle),blackII+opacity(.25),nolight);
triple f(pair t) { real xx=t.x;
return ((xx)*cos(t.y), (xx)*sin(t.y), ((xx)*cos(t.y))^2-((xx)*sin(t.y))^2+3);}

surface s=surface(f,(0,0),(1,2*pi),8,8,
    usplintype=new splintype[] {notaknot,notaknot,monotonic},
    vsplintype=Spline);
draw(shift(.2X+.2Y)*scale(.75,1.2,1)*s,blueII+opacity(.5),
    meshpen=PPblueblackI+thick(),nolight,render(merge=true));
real x(real t) {return cos(t);}
real y(real t) {return sin(t);}
real z(real t) {return 0;}
real w(real t) {return cos(t)*cos(t)-sin(t)*sin(t)+3;}
path3 p1=graph(x,y,z,0,2*pi,operator ..);
draw(shift(.2X+.2Y)*scale(.75,1.2,1)*p1,PPblueblackI);
path3 p2=graph(x,y,w,pi/2,pi,operator ..);
draw(shift(.2X+.2Y)*scale(.75,1.2,1)*p2,PPblueblack+1.5);
draw(Label("$0$",position=EndPoint),(0,0,.05)--(0,0,-.15));
draw(Label("$4$",position=EndPoint),(0.05,0,4)--(-.05,0,4));
// ...

draw(Label("$x$",position=EndPoint),(-1.25,0,0)--(1.75,0,0),Arrow3);
draw(Label("$y$",position=EndPoint),(0,-1.25,0)--(0,2,0),Arrow3);
draw(Label("$z$",position=EndPoint),(0,0,-.25)--(0,0,5.1),Arrow3);
label(scale(.75)*"$A$",.8X-.2Y,Z,PPblueblackI);
label(scale(.75)*"$f(A)$",.8X-.2Y+3.25Z,-Z,PPblueblackI);
draw(shift(-.55X+.2Y+4Z)*scale(.05,.05,.05)*unitsphere,PPblueblack+1);
label(scale(.75)*"$\max\{f(A)\}$",- .55X+.2Y+4.2Z,Z,PPblueblack);
draw(shift(.2X+1.4Y+2Z)*scale(.05,.05,.05)*unitsphere,PPblueblack+1);
label(scale(.75)*"$\min\{f(A)\}$",.2X+1.4Y+1.8Z,-Z,PPblueblack);
\end{asy}

```



Obr. 6: Graf funkcie $f(x, y) = x^2 - y^2 + 3$, $(x, y) \in R^2$

```

\begin{asy}
import graph3;size(3cm,0);
currentprojection=orthographic(2,2.5,2);
draw(box(0,(-.383616,1.180650,1.95)),blackII+linewidth(.25)+opacity(.5));
draw(shift(-.383616X)*scale(.05,.05,.05)*unitsphere,black);
draw(shift(1.180650Y)*scale(.05,.05,.05)*unitsphere,black);
draw(shift(1.95Z)*scale(.05,.05,.05)*unitsphere,black);
label(scale(.5)*"$x(t_0)$",(-.383616,0,0),Y,black);
label(scale(.5)*"$y(t_0)$", (0,1.180650,0),X,black);
label(scale(.5)*"$z(t_0)$", (0,0,1.95),X,black);
draw(Label(scale(.65)*"$x$",position=EndPoint),(-2,0,0)--(2,0,0),Arrow3);
draw(Label(scale(.65)*"$y$",position=EndPoint),(0,-2,0)--(0,2,0),Arrow3);
draw(Label(scale(.65)*"$z$",position=EndPoint),(0,0,-.25)--(0,0,4.2),Arrow3);
triple p(real t) { return (2*t*sin(2*pi*t)/pi,2*t*cos(2*pi*t)/pi,t); }
path3 pp=graph(p,0,2.95,operator ..);
draw(Label(scale(.65)*"$f$",position=EndPoint),pp,PPblueblack+1);
draw(scale(.05,.05,.05)*unitsphere,PPblueblack+1);
draw(shift(-.383616X+1.180650Y+1.95Z)*scale(.075,.075,.075)*unitsphere,PPblueblack+1);

```

```
label(scale(.6)*"$f(t_0)$",(-.383616,1.180650,1.95),Y,PPblueblack+1);
triple d(real t) { return (-.383616+7.221514*t,1.180650+3.015794*t,1.95+t); }
path3 dd=graph(d,-.24,.24,operator ..);
draw(Label(scale(.65)*"$d$",position=EndPoint),dd,PPoliveblack+1);
triple d0(real t) { return (-.383616+7.221514*t,1.180650+3.015794*t,0); }
path3 dd0=graph(d0,.24,-.24,operator ..);
draw(dd0--dd--cycle,blackII+linewidth(.25));draw(dd0,blackII+.5);
\end{asy}
```

Na ďalšom obrázku 8 je znázornený rozklad jednotkového vektora $\mathbf{u} \in R^3$, $\|\mathbf{u}\| = 1$ na uhly $\gamma_1, \gamma_2, \gamma_3$, ktoré zvierajú so súradnicovými osami x_1, x_2, x_3 . Platí $\mathbf{u} = (u_1, u_2, u_3) = (\cos \gamma_1, \cos \gamma_2, \cos \gamma_3)$, $\|\mathbf{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2} = \sqrt{\cos^2 \gamma_1 + \cos^2 \gamma_2 + \cos^2 \gamma_3} = 1$.

```
\begin{asy}
import graph3;import labelpath3;
texpreable("\usepackage{amsmath}");unitsize(.64cm);
currentprojection=orthographic(2.5,2,1);
draw(shift(2.42X)*scale(.05,.05,.05)*unitsphere);label(scale(.5)*"$u_1$", (2.2,0,0),-Z);
draw(shift(.88Y)*scale(.05,.05,.05)*unitsphere);label(scale(.5)*"$u_2$", (0,.8,0),-Z);
draw(shift(2.2Z)*scale(.05,.05,.05)*unitsphere);label(scale(.5)*"$u_3$", (0,0,2),-Y+.5Z);
draw(Label(scale(.65)*"$x_1$",position=EndPoint),(-.25,0,0)--(3,0,0),Arrow3(5));
draw(Label(scale(.65)*"$x_2$",position=EndPoint),(0,-.25,0)--(0,2.5,0),Arrow3(5));
draw(Label(scale(.65)*"$x_3$",position=EndPoint),(0,0,-.25)--(0,0,3.5),Arrow3(5));
draw(scale(2.42,.88,2.2)*unitbox,blackIV+linewidth(.25));
draw((2.42,0,0)--(2.42,.88,2.2) ^^ (0,.88,0)--(2.42,.88,2.2) ^^
(0,0,2.2)--(2.42,.88,2.2),blackIV+linewidth(.25));
draw(shift(1.1X+1.6Y+2.4Z)*scale(.075,.075,.075)*unitsphere,PPblueblack);
label(scale(.65)*"$\boldsymbol{a}$",1.1X+1.6Y+2.4Z,1.1Y-Z,PPblueblack);
draw((1.1,1.6,2.4)--(2.7,1.6,2.4) ^^ (1.1,1.6,2.4)--(1.1,3.2,2.4) ^^
(1.1,1.6,2.4)--(1.1,1.6,4),blueV+linewidth(.25));
draw(shift(1.1X+1.6Y+2.4Z)*rotate(-21.801409,X)*rotate(90,Y)
*arc(0,1.5,0,90,-44.395569,0,CCW),blueV,Arrow3(4));
label(scale(.4)*"$\gamma_1$",2.000127X+1.736410Y+2.741024Z,blueV);
draw(shift(1.1X+1.6Y+2.4Z)*rotate(47.726311,Y)*rotate(-90,X)
*arc(0,1.5,0,90,-74.940148,90,CCW),blueV,Arrow3(4));
label(scale(.4)*"$\gamma_2$",1.475127X+2.261410Y+2.741024Z,blueV);
draw(shift(1.1X+1.6Y+2.4Z)*rotate(111.801409,Z)
*arc(0,1.5,0,-90,-49.490825,90,CCW),blueV,Arrow3(4));
label(scale(.4)*"$\gamma_3$",1.475127X+1.736410Y+3.266024Z,blueV);
draw(shift(2.42X)*rotate(-21.801409,X)*arc(0,.4,0,90,-90,0,CCW),blackIV+linewidth(.25));
draw(shift(2.2506X+.0616Y+.154Z)*scale(.03,.03,.03)*unitsphere,blackIV);
draw(shift(.88Y)*rotate(47.726311,Y)*arc(0,.4,0,90,-90,90,CCW),blackIV+linewidth(.25));
draw(shift(.15125X+.715Y+.1375Z)*scale(.03,.03,.03)*unitsphere,blackIV);
draw(shift(2.2Z)*rotate(111.801409,Z)*rotate(90,X)
*arc(0,.4,0,-90,-90,90,CCW),blackIV+linewidth(.25));
draw(shift(.1694X+.0616Y+2.046Z)*scale(.03,.03,.03)*unitsphere,blackIV);
draw(Label(scale(.65)*"$\boldsymbol{u}$",position=EndPoint),
(1.1,1.6,2.4)--(1.1+2.42,1.6+.88,2.4+2.2),PPblueblack+opacity(.7),Arrow3(5));
draw(Label(scale(.65)*"$\boldsymbol{u}$",position=EndPoint),
(0,0,0)--(2.42,.88,2.2),PPblueblack+opacity(.7),Arrow3(5));
draw(rotate(-21.801409,X)*rotate(90,Y)
*arc(0,1.5,0,90,-44.395569,0,CCW),PPblueblack+opacity(.7),Arrow3(4));
label(scale(.4)*"$\gamma_1$",0.900127X+0.136410Y+0.341024Z,PPblueblack);
draw(rotate(47.726311,Y)*rotate(-90,X)
*arc(0,1.5,0,90,-74.940148,90,CCW),PPblueblack+opacity(.7),Arrow3(4));
label(scale(.4)*"$\gamma_2$",0.375127X+0.661410Y+0.341024Z,PPblueblack);
draw(rotate(111.801409,Z)
```

```

*arc(0,1.5,0,-90,-49.490825,90,CCW),PPblueblack+opacity(.7),Arrow3(4));
label(scale(.4)*"$\gamma_3$",0.375127X+0.136410Y+0.866024Z,PPblueblack);
\end{asy}

```

Na poslednom obrázku 9 je znázornená funkcia dvojrozmerná funkcia $f(x, y)$, $(x, y) \in \mathbb{R}^2$ definovaná vzťahom $f(x, y) = xy$ pre $|x| \geq |y|$ a vzťahom $f(x, y) = 0$ pre $|x| < |y|$.

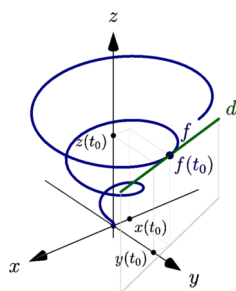
```

\begin{asy}
import graph3;
texpreamble("\usepackage{amsmath}");unitsize(1.25cm);
currentprojection=orthographic(2.25,1.75,1);
triple f(pair t) { real xx=t.x;
return ((xx)*cos(t.y), (xx)*sin(t.y), (xx)^2*sin(t.y)*cos(t.y));}
surface s1=surface(f,(0,-pi/4),(1.5,pi/4),6,6,
    usplinetype=new splinetype[] {notaknot,notaknot,monotonic},
    vsplinetype=Spline);
draw(s1,blueII+opacity(.5),meshpen=PPblueblackI+thick(),nolight,render(merge=true));
surface s2=surface(f,(0,3*pi/4),(1.5,5*pi/4),6,6,
    usplinetype=new splinetype[] {notaknot,notaknot,monotonic},
    vsplinetype=Spline);
draw(s2,blueII+opacity(.5),meshpen=PPblueblackI+thick(),nolight,render(merge=true));
triple f(pair t) { real xx=t.x;
return ((xx)*cos(t.y), (xx)*sin(t.y), 0);}
surface n1=surface(f,(0,pi/4),(1.5,3*pi/4),6,6,
    usplinetype=new splinetype[] {notaknot,notaknot,monotonic},
    vsplinetype=Spline);
draw(n1,blueII+opacity(.5),meshpen=PPblueblackI+thick(),nolight,render(merge=true));
surface n2=surface(f,(0,5*pi/4),(1.5,7*pi/4),6,6,
    usplinetype=new splinetype[] {notaknot,notaknot,monotonic},
    vsplinetype=Spline);
draw(n2,blueII+opacity(.5),meshpen=PPblueblackI+thick(),nolight,render(merge=true));
draw(scale(.05,.05,.05)*unitsphere,PPblueblack+1);
label(scale(.6)*"$\boldsymbol{0}$",(0,0,0),.5X+.25Y-.5Z);
label(scale(.6)*"$f$",(-1.3,-.75,.975),-.5X+.25Z);
draw(Label(scale(.6)*"$-1$",position=EndPoint),(0.05,0,-1)--(-.05,0,-1));
draw(Label(scale(.6)*"$1$",position=EndPoint),(0.05,0,1)--(-.05,0,1));
draw(Label(scale(.6)*"$1$",position=EndPoint),(1,0,.05)--(1,0,-.05));
draw(Label(scale(.6)*"$-1$",position=EndPoint),(-1,0,.05)--(-1,0,-.05));
draw(Label(scale(.6)*"$1$",position=EndPoint),(0,1,.05)--(0,1,-.05));
draw(Label(scale(.6)*"$-1$",position=EndPoint),(0,-1,.05)--(0,-1,-.05));
draw(Label(scale(.65)*"$x$",position=EndPoint),(-1.75,0,0)--(1.75,0,0),Arrow3);
draw(Label(scale(.65)*"$y$",position=EndPoint),(0,-1.75,0)--(0,1.75,0),Arrow3);
draw(Label(scale(.65)*"$z$",position=EndPoint),(0,0,-1.25)--(0,0,1.75),Arrow3);
\end{asy}

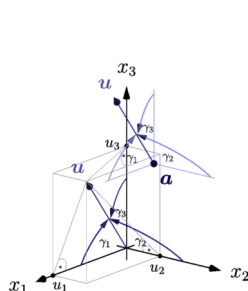
```

Záver

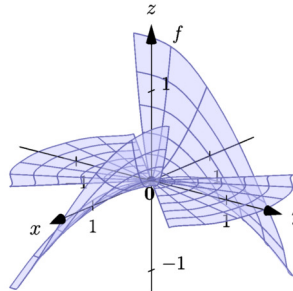
Iná možnosť na kreslenie je použitie balíčka TikZ (T ist kein Zeichenprogramm), ktorý je veľmi účinný nástroj na kreslenie jednoduchých aj zložitejších obrázkov, grafov, schém, grafov funkcií... Jeho autorom je Till Tantau [6]. Základným princípom práce s týmto balíčkom sa zaoberal A. Kozubík na OSSConf 2012 v príspevku „Naučím vás kresliť alebo predstavenie balíčka TikZ“ [4]. Balíček je štandardnou súčasťou každej inštalácie



Obr. 7: Špirála



Obr. 8: Vektory



Obr. 9: Graf funkcie

$\text{\TeX}$ -u. Manuál a mnoho užitočných informácií nájdeme na portáli CTAN
<https://www.ctan.org/pkg/pgf>.

Literatúra

- [1] Blaško, R.: *Asymptote a $\text{\LaTeX}$* , Zborník príspevkov medzinárodnej konferencie OSSConf 2018 a 2019, 2.–4. júla 2018, Žilina, str. 7–14. ISBN 978-80-554-1627-4. cit. 3
- [2] Hammerlindl, A., Bowman, J., Prince, T.: *Asymptote: The Vector Graphics Language*, <https://asymptote.sourceforge.io/>. cit. 3
- [3] Knuth, D. E.: *The $\text{\TeX}$ book*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9. cit. 2
- [4] Kozubík, A.: *Naučím vás kresliť alebo predstavenie balíčka TikZ*, Zborník príspevkov medzinárodnej konferencie OSSConf 2012, 2.–4. júla 2012, Žilina, str. 91–96. ISBN 978-80-970457-2-2. cit. 14
- [5] Stríž, P.: *Jak jsem se skamarádil s $\text{\LaTeX}$ ovým balíčkem animate*, Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, str. 177–184. ISBN 978-80-970457-0-8. cit. 4
- [6] Tantau, T.: *TikZ & PGF*, Manual for Version 3.1.3, <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>. cit. 14

NĚKOLIK POSTŘEHŮ K ASYMPTOTE SEVERAL NOTES ON ASYMPTOTE

Pavel Stríž

E-mail: pavel@striz.cz

1. Vzpomínky na Asymptote

Na své první pokusy s [Asymptote](#) si pamatuji dobře, neb jsem řešil tři školské grafy ve 3D paralelně v Asymptote a přes [Jmol](#) v prostředí [Sage](#). Od té doby již neužíváme balíček [movie15](#), ale jeho nástupce balíček [media9](#).

```
$ texdoc movie15 media9
```

Inspirací ke studiu mi byla hlavně tato galerie, <http://www.piprime.fr/asymptote/>. Autorem ukázek je Philippe Ivaldi.

2. Instalace Asymptote

Poněvadž pracuji na starších strojích, vždy něco nejede, byť používám poslední verzi T_EXLive. V tomto případě se mi nepodařilo generovat pdf skrz starší grafickou kartu, až po zadání

```
export MESA_GL_VERSION_OVERRIDE=1.2
```

do souboru `~/.bashrc` a načtení přes `source ~/.bashrc` se mi to podařilo, ale to jen pro obrázky vykreslené ve 2D. U hledání problému mi pomohly parametry Asymptote `-v`, `-vv` (nebo `-v -v`) a hlavně `-vvv` (nebo `-v -v -v`). Ve výpisech lze jít až do páté úrovně (str.168 dokumentace). Linuxový `man` hovoří o `-v` a jistě možnosti jít hlouběji, ale nehovoří konkrétně.

```
$ texdoc asy # nebo  
$ man asy
```

Přesdl jsem na novější stroj. Na Xubuntu 20.04 bylo potřeba doinstalovat knihovnu `freeglut3`, speciálně se jednalo o knihovnu `libglut.so.3`.

```
$ sudo apt install freeglut3
```

Poté se mi podařilo vyrenderovat všechny grafy předchozího článku Rudolfa Blaška.

3. WebGL

Zajímavá je i situace, když chceme získat webovou verzi grafů. Možná si říkáte, proč je to tak důležité. Pokud pracujete v linuxovém prostředí, tak firma

Adobe zrušila podporu pro prohlížeč pdf [Adobe Reader](#). Poslední známá verze je 9.5.5 pro 32bitové stroje. Uživatelé pak musí sáhnout po virtuálním stroji typu [Oracle VM VirtualBox](#), po [Wine](#) 32bitové verzi, míchat 32bitové a 64bitové aplikace či zkusit podobné řešení ([FoxIt Reader](#), ale také jede jen něco). Stručně řečeno, prakticky na libovolném stroji s Microsoft Windows či MacOS 3D grafy interaktivně v pdf uvidíte, ale zápasíte s tím pod Linuxem.

O WebGL (vhodné i pro chytré telefony) a nastavení je zmínka na str. 137 v dokumentaci [Asymptote](#). Poslední verze programu je 2.65.

```
$ texdoc asymptote
```

Vzal jsem si jeden ze vzniklých asy souborů předchozího článku, horní graf, obr. 3 na str. 9, zkopíroval na `webgl-pokus.asy` a vyčistil do této minimální podoby (vymazání úvodních definic, globálního definování neužitých barev a promazání prázdných řádků):

```
import graph3;
import palette;
size(175,175,IgnoreAspect);
currentprojection=perspective(2,.55,0.25);
real f(pair z){return sin(z.x)*sin(z.y);}
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,Spline),lightblue,render(merge=true));
surface f=surface(f,(-pi,-pi),(pi,pi),24,Spline);
draw(f,mean(palette(f.map(zpart),Rainbow(24))),brown);
draw(Label("$x$",1),(-1.2pi,0,0)--(1.2pi,0,0),brown,Arrow3);
draw(Label("$y$",1),(0,-1.2pi,0)--(0,1.2pi,0),brown,Arrow3);
draw(Label("$z$",1),(0,0,-1.2)--(0,0,1.2),brown,Arrow3);
label("$z=\sin{x}\cdot\sin{y}$",(.5,.5,-1.35),SE,deepblue);
```

O výstupech hovoří manuál v kapitole 8.29 *three*, speciálně na str. 137. Prvně jsem si vyzkoušel náhled v okně:

```
$ asy -V webgl-pokus.asy
```

Nasledované pokusem o vygenerování webové stránky.

```
$ asy -f html webgl-pokus.asy
```

Vzniká soubor `webgl-pokus.html`, který se mi podařilo otevřít i na starém notebooku. Alternativou je zadat výstupní formát přímo v asy souboru (středník na konci řádků je nutný, je součástí syntaxe jazyka):

```
import settings;
settings.outformat="html";
// outformat="html";
```

Bez prázdných řádků soubor vypadá přibližně takto:

```
<!DOCTYPE html>
<!-- Use the following line to embed this file within another web page:
```

```
<iframe src="webgl-pokus.html" width="176" height="176"
  frameborder="0"></iframe>
-->
<html lang="">
<head>
  <title>webgl-pokus</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <meta name="viewport" content="user-scalable=no"/>
  <script
src="https://vectorgraphics.github.io/asymptote/base/webgl/asygl-1.00.js">
</script>
<script>
  canvasWidth=176;
  canvasHeight=176;
  absolute=false;
  // ...
  Centers=[
    [-10.3373,-38.23629,-223.1933],
    [81.93387,-1.42227,-289.569],
    [0.5386997,67.84467,-262.8985],
    [9.891595,-74.96387,-327.4771],
  ];
</script>
</head>
<body style="overflow: hidden;" onload="webGLStart();">
  <canvas id="Asymptote" width="176" height="176" style="border: none;">
</canvas>
</body>
</html>
```

Nepodařilo se mi dle návodu na str. 137 vložit vzniklou webovou stránku do další přes `<object>` ani přes `<iframe>` jak radí poznámka ve vzniklém souboru. V konzoli se píše `SecurityError: Permission denied to access property "document" on cross-origin object.`

Jisté řešení nabízí odpovědi na <https://stackoverflow.com/questions/36333978/error-permission-denied-to-access-property-document>, ale to je bezpečnostní riziko. Vyčištěním vzniklého souboru na úroveň prostého JavaScriptu a přes `<canvas>` se mi sice pořadilo dostat jeden graf na webovou stránku, ale jen jeden. Ideální je zůstat u hypertextových odkazů na vznikající soubory dle vzoru galerií na <https://asymptote.sourceforge.io/gallery/>.

Pro badatele jistě stojí za pozornost možnost konverze PostScriptu do formátu asy s možností náhledu a následné editace:

```
$ pstoeedit -f asy <soubor.eps> <soubor.asy>
$ asy -V <soubor.asy>
```

V případě problémů s písmy se u `pstoeedit` používá parametr `-dt`.

4. Ovládání T_EXové proměnné mimo T_EX

Rudolf Blaško byl spokojený s prvními výstupy, ale u jeho knihy bylo potřeba předchozí kroky zautomatizovat, jednalo se konkrétně o definice v prostředí `asydef`, které bylo potřeba změnit. Zdá se, že ani parametry z příkazového řádku programu `asy` tyto definice nemění. Chceme totiž knihu nechat beze změny, ale zároveň si vygenerovat webové stránky s 3D objekty.

Vyřešil jsem to přepínačem na T_EXové úrovni, který řídí dávkový soubor. Ukážme si minimální ukázkou. Tento blok jsem si přidal za autorovy definice.

```
\newcount\prepinac
\input ovladani.tex
\ifnum\prepinac=1
\begin{asydef}
settings.outformat="html";
settings.inlineimage=false;
settings.embed=false;
\end{asydef}
\fi
```

V dávkovém souboru pak stačí mít něco takového.

```
soubor="ovladani.tex"
# Získání pdf knihy...
echo "\\prepinac=0" >$soubor
lualatex kniha.tex # plus další příkazy
# Generování html s 3D objekty...
echo "\\prepinac=1" >$soubor
lualatex kniha.tex # plus další příkazy
```

Tento přístup se používá, když chceme vygenerovat několik verzí knihy, např. knihu ve více šablonách, verzi s barevnými a černobílými obrázky, verzi na tisk a na obrazovku, verzi s a bez 3D objektů, pracovní a konečnou verzi čísla časopisu ap.

5. Generování asy souborů přes R: jednoduchá ukáзка

O interaktivní 3D grafice jsem poprvé četl ve Zpravodaji Čs. sdružení uživatelů T_EXu od Romana Plcha, viz <https://www.cstug.cz/bulletin/pdf/2013-1.pdf>. Tam je zmínka o generování asy souborů, včetně jazyka R, konkrétně se jednalo o knihovnu `misc3d` zmíněnou na str. 40. Knihovna umí exportovat objekt `Triangles3D`, který se musí získat. Na knihovně staví novější knihovna `plot3D`.

Zvláštní kategorii tvoří knihovna `rgl`. Přišlo mně zajímavé vyzkoušet export do asy, tím lze získat 3D model v pdf, ale zároveň interaktivní webovou

stránku (WebGL), proto nepoužiji příkaz `writeWebGL`, ale `writeASY`. Zkusme si minimální ukázkou a v tomto článku se zaměřím už jen na webovou stránku.

Připravíme si soubor `test-rgl.R`:

```
# install.packages("rgl")
library(rgl)
with(iris, plot3d(Sepal.Length, Sepal.Width, Petal.Length, type="s",
  col=as.numeric(Species)))
writeASY()
```

Spustíme:

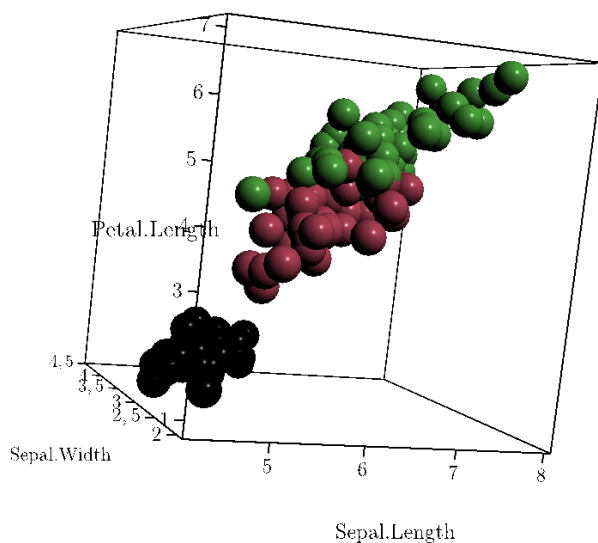
```
$ Rscript test-rgl.R
```

S největší pravděpodobností dostaneme chybovou hlášku od programu Asymptote. Psal jsem autorovi, chyba je opravena, ale ještě nová verze není ve všech linuxových a $\text{\TeX}$ ových distribucích. Nepomůže nám ani pokus přes `writeASY(ver244=TRUE)`.

Vzniká nám soubor `scene.asy`, který upravíme. Do 2. řádku zasáhneme: `settings.outformat = "html"`; a vymažeme 5 řádků, od `currentlight = light` (po nejbližší);.

Spouštíme následující řádky a dostáváme takový výsledek.

```
$ asy -v scene.asy
$ firefox scene.html
```



Badatele odkazují na prozkoumání <https://cran.r-project.org/web/packages/rgl/vignettes/rgl.html>.

6. Generování asy souborů přes R: pokročilejší ukázka

V IB č. 1/2018 jsem na str. 13–15 zmínil jednoduché [ukázky](#) z manažerského rozhodování, z oblasti rozhodování při riziku. Chtěl bych si zkusit [obdobné grafy](#), ale interaktivní. Představím vám knihovnu [gMOIP](#), resp. mohu doporučit webové stránky autora (<https://github.com/relund/gMOIP>) a jeho blog (<https://www.research.relund.dk/>).

Naše problémy jsou, že budeme chtít víc 3D modelů a určitě nebudeme chtít dělat ruční zásahy. Ukážeme si zásahy do textových souborů přes Lua, je to jazyk založený na C++, který dominuje v $\text{\TeX}$ ovém světě a hraje nemalou roli ve světě počítačových her. Speciální pozornost by si zasloužila knihovna [LPeg](#), která si získala místo ve světě regulárních výrazů.

Připravíme si soubor `test-gmoic.R`:

```
# install.packages("gMOIP")
library(gMOIP)
# definice
A <- matrix( c(3, 2, 5, 2, 1, 1, 1, 1, 3, 5, 2, 4), nc = 3, byrow = TRUE)
b <- c(55, 26, 30, 57)
obj <- c(20, 10, 15)
view <- matrix( c(-0.45, -0.446, 0.77, 0, 0.886, -0.32, 0.33, 0, 0.098,
0.835, 0.54, 0, 0, 0, 0, 1), nc = 4)
# první obrázek
loadView(v = view, zoom = 0.75)
plotPolytope(A, b, faces = c("c","c","c"), type = c("c","i","i"), plotOptimum
= TRUE, obj = obj)
rgl::writeASY(outtype = "pdflatex", prc = TRUE, title = "polytope-1")
# druhý obrázek
loadView(v = view, zoom = 0.75)
plotPolytope(A, b, type = c("c","c","i"), plotOptimum = TRUE, obj = obj,
plotFaces = FALSE)
rgl::writeASY(outtype = "pdflatex", prc = TRUE, title = "polytope-2")
```

V dalším kroku si připravíme $\text{\TeX}$ ový dokument `asy-pres-r.tex`:

```
\documentclass[a4paper]{article}
\usepackage[luatex]{graphicx}
\usepackage{asymptote}
\begin{document}
\input{polytope-1.tex}\par
\input{polytope-2.tex}
\end{document}
```

Následuje soubor v Lua, který nám upravuje získané soubory, nazval jsem jej `smaz-light.lua`:

```
-- Načtení souboru
nazev=arg[1]
print("Zpracovávám soubor "..nazev)
```

```
soubor=io.open(nazev)
obsah=soubor:read("*all")
soubor:close()
-- Zásah do souboru
obsah=unicode.utf8.gsub(obsah,"currentlight =.-%");", "", 1)
obsah=unicode.utf8.gsub(obsah, "settings%.inlineimage=true;",
    "settings.inlineimage=false;", 1)
-- Přepsání původního souboru
kam=io.open(nazev,"w")
kam:write(obsah)
kam:close()
```

Nyní nezbývá, než si připravit dávkový soubor `davka.sh`, který spustíme přes `bash davka.sh`:

```
# Základ generování asy souborů
zaklad="asy-pres-r"
Rscript test-gmoic.R
lualatex --shell-escape $zaklad.tex 1>/dev/null 2>&1
# Úpravy v asy souborech
for soubor in `find -iname $zaklad-*.asy`; do
texlua smaz-light.lua $soubor
asy -v $soubor
asy -v -f html $soubor
done
# Náhled na výsledky
firefox $zaklad-*.html
```

Co se přesně děje? Z R se vygenerují dva $\text{\TeX}$ ové soubory, které použije $\text{\TeX}$ a při prvním běhu vygeneruje asy soubory. Do nich zasáhneme na textové úrovni a přes Asymptote vygenerujeme 3D objekty a webové stránky, které si otevřeme ve webovém prohlížeči. Výhoda tohoto přístupu je, že u dalších běhů Lua $\text{\TeX}$ u vysázíme v `asy-pres-r.pdf` 3D modely. Vedle toho máme dostupnou interaktivní webovou verzi.

Zde je náhled na očekávaný výsledek ve dvou záložkách prohlížeče.

