

INFORMAČNÍ BULLETIN



České statistické společnosti

Ročník 31, číslo 5, prosinec 2020

Obsah

Vědecké a odborné články

Pavel Stríž

Vítejte ve světě animací! 3

Pavel Stríž

První kontakt s programem Manim 21

Zprávy a informace

Pavel Stríž

TeXLive 2020: Novinky v TeXovém světě 30

Pavel Stríž

Sbohem, drahý Karle, sbohem! 43

Zdeněk Hlávka

Vyhlášení soutěže: Data Analysis Competition 2021 53

Informační Bulletin České statistické společnosti vychází čtyřikrát do roka v českém vydání. Příležitostně i mimořádné české a anglické číslo. Vydavatelem je Česká statistická společnost, IČ 00550795, adresa společnosti je Na padesátém 81, 100 82 Praha 10. Evidenční číslo registrace vedené Ministerstvem kultury ČR dle zákona č. 46/2000 Sb. je E 21214. Časopis je sázen v programu TeX, ve formátu LuaHBTeX s písmy balíku Cšfonts.

The Information Bulletin of the Czech Statistical Society is published quarterly.
The contributions in the journal are published in English, Czech and Slovak languages.

Předseda společnosti: Mgr. Ondřej Vencálek, Ph.D., Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta Univerzity Palackého, 17. listopadu 12, 771 46 Olomouc, e-mail: ondrej.vencalek@upol.cz.

Redakce: prof. RNDr. Gejza DOHNAL, CSc. (šéfredaktor), prof. RNDr. Jaromír ANTOCH, CSc., doc. RNDr. Zdeněk KARPÍŠEK, CSc., RNDr. Marek MALÝ, CSc., doc. RNDr. Jiří MIČÁLEK, CSc., prof. Ing. Jiří MILITKÝ, CSc., doc. Ing. Iveta STANKOVIČOVÁ, PhD., doc. Ing. Josef TVRDÍK, CSc., Mgr. Ondřej VENCÁLEK, Ph.D.

Redaktor časopisu: Mgr. Ondřej VENCÁLEK, Ph.D., ondrej.vencalek@upol.cz.
Informace pro autory jsou na stránkách společnosti, <http://www.statspol.cz/>.

DOI: 10.5300/IB, <http://dx.doi.org/10.5300/IB>

ISSN 1210–8022 (Print), ISSN 1804–8617 (Online)

Toto číslo bylo vytištěno s laskavou podporou Českého statistického úřadu.

VÍTEJTE VE SVĚTĚ ANIMACÍ!

WELCOME TO THE WORLD OF ANIMATIONS!

Pavel Stríž

E-mail: pavel@striz.cz

1. O animacích

V dobách dřívějších se na webové stránky často dávaly animované gify. Tyto dny se složením jednotlivých obrázků a jejich extrakcí z gifů pomáhá ImageMagick či odnož GraphicsMagick.

Druhý oblíbený formát je Flash. Firma Adobe však končí s podporou programu Flash Player v prosinci 2020. Tedy například tyto animace

<https://melusine.eu.org/syracuse/metapost/animations/>

se nám hned tak v budoucnu nepodaří otevřít. Na Linuxu lze na přehrání užít program **gnash**.

```
$ sudo apt install gnash
```

Co se týče zařazení animace do PDF, tak jednu z možností přes JavaScript zmínili J. Holeček a P. Sojka v článku Animations in pdfTeX-generated PDF ve sborníku *TeX, XML, and Digital Typography*, Springer, 2004, str. 179–191. O rok později to zmiňuje i J. Gilg v článku PDF-Animationen v časopisu *Die TeXnische Komödie* 4, 2005, str. 30–37. Podpůrný balíček **interactiveplot** vzniká roku 2014 a vzniká balík **AcroTeX**, některé části jsou zadarmo, některé nabízené k prodeji.

Obecně se může animace uložit jako (audio)videostopa. K tomu nám slouží především balíky **ffmpeg** a ve starších linuxových distribucích **avconv**.

```
$ sudo apt install ffmpeg
```

Ve světě open source software existuje nespočet nástrojů na přehrání videa, např. **mpv**, **vlc** a pro Raspberry Pi optimalizovaný **omxplayer**.

```
$ sudo apt install mpv vlc-bin
```

Zařazení audiovideo stop do pdf nabízí TeXový balíček **movie15** a nyní jeho nástupce balíček **media9**. Vedle toho umožňují zařadit soubory s Flash animacemi a 3D objekty (PRC, U3D).

```
$ texdoc media9 movie15
```

2. animate v2020-04-25

Vrcholem v $\TeX$ ovém světě je balíček `animate`, který umožňuje zařadit animace vznikající vrstvením obrázků na sebe, jejich případné časování a výběr kreslených částí, parametr `timeline` (to je výhodné u rozsáhlých obrázků skrz velikost výsledné animace) a nově pomáhá s generováním animovaných svg. Zkusme si prvně získat animace ve čtyřech základních nástrojích dostupných v $\TeX$ Live u ukázek mimo $\TeX$ Live.

2.1. METAPOST v2.0

Dokumentaci získáme přes

```
$ texdoc metapost metafun-p
```

Jedna z nejstarších galerií je od Vincent Zoonekynd z roku 1999.

<http://zoonek.free.fr/LaTeX/Metapost/metapost.html>

Archiv: <https://www.ctan.org/tex-archive/info/metapost/examples>

Jednoduchou ukázkou vzniku animace přes sérii obrázku nalezneme zde:

<https://adityam.github.io/context-blog/post/metapost-animation>

Pokročilé animace hledejte na:

<http://www-math.univ-poitiers.fr/~phan/animations.html>

<https://melusine.eu.org/syracuse/metapost/animations>

Díky knihovně `luamplib` umíme psát kód METAPOSTu přímo v $\TeX$ ovém dokumentu, zájemce nechť nahlédne na tuto ukázkou:

<https://melusine.eu.org/syracuse/luatex/luamplibAnimate>

Při problémech s písmy na úrovni METAPOSTu se doporučuje užít v preambuli `prologues:=3`. Ukážeme si animaci vykreslení celého odstavce ze zmíněné `galerie`. Jen se mi nepodařilo ji vygenerovat přes balíček `luamplib` přímo z $\TeX$ ového dokumentu.

<https://melusine.eu.org/syracuse/metapost/animations/mehats>

Soubor `010.mp` vypadá takto:

```
filenametemplate "%j-%3c.mps";
verbatimtex%&latex
\documentclass{article}
\usepackage{lmodern} \usepackage[utf8]{inputenc} \usepackage[T1]{fontenc}
\begin{document}
etex;
picture tex_pct, glp_pct; numeric glp_num, pth_num[]; path glp_pth[][];
tex_pct:=btex{\begin{minipage}{\textwidth}\begin{center}
Ukázka animace spojených sil\\baličků METAPOST a animate!
\end{center}\end{minipage}}etex;
glp_pct:=nullpicture;
string fnt_str, txt_str, sub_str; numeric txt_wd; glp_num:=0;
```



```

for tkn within tex_pct:
  if textual tkn:
    fnt_str:=fontpart tkn; txt_str:=textpart tkn; txt_wd:=0;
    for glp_idx=0 upto (length txt_str-1):
      sub_str:=substring (glp_idx, glp_idx+1) of txt_str;
      pth_num[glp_num]:=0;
      for sub_tkn within glyph ASCII sub_str of fnt_str
        scaled (fontsize fnt_str/1000) xscaled xpart tkn
        yscaled ypart tkn shifted (txt_wd+xpart tkn, ypart tkn):
          glp_pth[glp_num][pth_num[glp_num]]:=pathpart sub_tkn;
          addto glp_pct doublepath glp_pth[glp_num][pth_num[glp_num]];
          pth_num[glp_num]:=pth_num[glp_num]+1;
      endfor
      glp_num:=glp_num+1; txt_wd:=txt_wd+
        (xpart tkn)*xpart urcorner (sub_str infont fnt_str);
    endfor
  fi
endfor
numeric bg_wd, bg_hg; picture bg_pct; bg_wd:=1280; bg_hg:=300;
bg_pct:=nullpicture;
addto bg_pct contour origin--(bg_wd, 0)--(bg_wd, bg_hg)--(0, bg_hg)--cycle;
numeric fg_wd, fg_hg; transform fit_trn;
fg_wd:=xpart(urcorner glp_pct-llcorner glp_pct); fg_hg:=ypart(urcorner
  glp_pct-llcorner glp_pct);
fit_trn:=identity shifted -.5[llcorner glp_pct, urcorner glp_pct]
  scaled .9min(bg_wd/fg_wd, bg_hg/fg_hg) shifted +.5[llcorner bg_pct,
  urcorner bg_pct];
color bg_clr, fg_clr; pen fg_pen; numeric dot_scl; bg_clr:=white;
fg_clr:=black; fg_pen:=pencircle scaled 2; dot_scl:=4;
numeric duration, fps, f_num; duration:=10; fps:=25; f_num:=fps*duration;
for idx=0 upto (f_num/2-1):
  beginfig(idx)
    draw bg_pct withcolor bg_clr; drawoptions (withcolor fg_clr);
    for i=0 upto glp_num-1:
      for j=0 upto pth_num[i]-1:
        path pth; numeric tim; pth:=glp_pth[i][j] transformed fit_trn;
        tim:=arctime 2(arclength pth)/f_num*idx of pth;
        draw subpath (0, tim) of pth withpen fg_pen;
        draw point (tim) of pth withpen fg_pen scaled dot_scl;
      endfor
    endfor
    drawoptions ();
  endfig;
endfor
end.

```

Pomocný soubor je 010-metapost.tex:

```

\documentclass{article}
\usepackage{animate}
\usepackage{graphicx}

```

```
\begin{document}
\animategraphics[width=0.75\textwidth, controls=all,
poster=last]{10}{010-}{000}{124}
\end{document}
```

Spouštíme:

```
$ mpost 010.mp
$ lualatex 010-metapost.tex
$ lualatex 010-metapost.tex
```

Ukázka animace spojených sil balíčku METAPOST a animate!



2.2. PSTricks v2.97 a nespočet jeho balíčků

Galerie najdeme na stránkách programu:

<http://tug.org/PSTricks/main.cgi?file=packages>

Na animace se častokrát používá pomocný balíček `multido`, ukázky ze světa PSTricks najdeme přímo v balíčku `animate`. Z galerií vypíchněme:

<https://tug.org/PSTricks/main.cgi?file=Animation/gif/gif>

<https://tug.org/PSTricks/main.cgi?file=Animation/basics>

melusine.eu.org/syracuse/pstricks/pst-solides3d/animations

```
$ texdoc multido animate
```

Zvláštní kategorii tvoří server s blogy <http://pstricks.blogspot.com>. Narazil jsem na celou řadu zajímavých balíčků, např. `xint`. Na serveru je představena celá řada vznikajících a pracovních balíčků. Zmíním vybrané.

Dle vzoru <https://geargenerator.com> vzniká balíček `pst-gears`, v poslední verzi v0.6. Verze pro 2D je ke stažení na:

manuel.luque.free.fr/pst-gears-2020/pst-gear-2020-v0.6.zip

drive.google.com/drive/folders/1zyXX3w525m99YPM4wkSd3acJRbcCVs4o

Verze pro 3D, `pst-gearsiiid`, ve verzi v3, je dostupná na:

<http://manuel.luque.free.fr/gearsIIID/pst-gearsIIID-v3.zip>

drive.google.com/open?id=1sSIVv2rqbFHhCkyX_VvZ5oKLIMXxdrv2

Zaujal mě i balíček `pst-crayon`, v3.1, ze kterého si přebereme ukázku.

https://drive.google.com/open?id=0Bw5_RBu0n8-qbkhrVGN1REVRUGs

Soubor `020.tex` vypadá takto:

```

\documentclass[pstricks]{standalone}
\usepackage{pst-plot,pst-3d,pst-gears,pst-node}
\usepackage[nomessages]{fp} \makeatletter
\define@key[psset]{\theta1}{\def\psk@thetaA{#1}}
\define@key[psset]{\theta2}{\def\psk@thetaB{#1}}
\psset{theta1=-90,theta2=90}
\def\psElasticFixedTwoWheels{\pst@object{psElasticFixedTwoWheels}}
\def\psElasticFixedTwoWheels@i{\begin@SpecialObj
\FPset{\ZA}{\psk@ZA}\FPset{\ZB}{\psk@ZB} \FPset{\module}{\psk@m}
\FPeval{\RA}{\ZA*\module/2}\FPeval{\RB}{\ZB*\module/2} \FPeval{\OB}{\RA+\RB}
\FPeval{\RAP}{(\RA*2-2.5*0.2)/2} \FPeval{\RBP}{(\RB*2-2.5*0.2)/2}
\FPset{\OMEGAA}{-1} \FPeval{\OMEGAB}{(-\OMEGAA)*\ZA/\ZB}
\FPset{\ANGLE}{\psk@wheelrotation} \FPeval{\ANGLErad}{\ANGLE*\FPpi/180}
\FPeval{\nombrePoints}{trunc(2*\ANGLE+5,0)}
\FPeval{\thetaA}{(\psk@thetaA)*\FPpi/180}
\FPeval{\thetaB}{(\psk@thetaB)*\FPpi/180}
\FPeval{\xA}{0.9*\RAP*cos(\thetaA+\OMEGAA*\ANGLErad)}
\FPeval{\yA}{sin(\thetaA+\OMEGAA*\ANGLErad)*\RAP*0.9}
\FPeval{\xB}{cos(\thetaB+\OMEGAB*\ANGLErad)*\RBP*0.9}
\FPeval{\yB}{sin(\thetaB+\OMEGAB*\ANGLErad)*\RBP*0.9+\OB}
\FPeval{\xM}{(\xA+\xB)/2} \FPeval{\yM}{(\yA+\yB)/2}
\ThreeDput[normal=0 0 1](0,0,0){\psgrid[subgriddiv=0,gridlabels=0pt]
\rput(0.05,-0.05){\pstgears[circles=false, polarangle=90, fillstyle=solid,
color1=black, color2=black]}
\pstgears[circles=false,polarangle=90,fillstyle=solid]
\parametricplot[linecolor=red, plotpoints=\nombrePoints, algebraic,
linewidth=0.1]{0}{\ANGLErad}{
(\RAP*0.9*cos(\thetaA+\OMEGAA*t)+\RBP*0.9*cos(\thetaB+\OMEGAB*t))/2|
(\RAP*0.9*sin(\thetaA+\OMEGAA*t)+\RBP*0.9*sin(\thetaB+\OMEGAB*t)+\OB)/2}
\psline{->}(0,0)(0,1) \psline{->}(0,0)(1,0)
\pscircle[linestyle=dotted](0,0){\RA} \pscircle[linestyle=dotted](0,\OB){\RB}}
\ThreeDput[normal=0 1 0](\xA,\yA,0) {\psline[linewidth=0.1]{-}(0,0)(0,1)
\pnode(0,1){P1}}
\ThreeDput[normal=0 1 0](\xB,\yB,0)
{\psline[linewidth=0.1]{-}(0,0)(0,1)\pnode(0,1){P2}}
\ThreeDput[normal=0 1 0](\xM,\yM,0) {\pnode(0,1){P3} \pnode(0,0){P4}}
\psline[linecolor=blue](P1)(P2) \psline[linecolor=red]{->}(P3)(P4)
\psdot[linecolor=blue](P3)
\end@SpecialObj}\ignorespaces} \makeatother
\begin{document}
\multido{\i=0+45}{17}{\begin{pspicture}(-5,-5)(5,6)
\psElasticFixedTwoWheels[Z1=35, Z2=10, m=0.15, viewpoint=-1 -2 2,
arrowinset=0, arrowsize=0.2, wheelrotation=\i, linewidth=0.025,
color1=yellow, color2=green]
\end{pspicture}}
\end{document}

```

Pomocný soubor je 020-pstricks.tex:

```

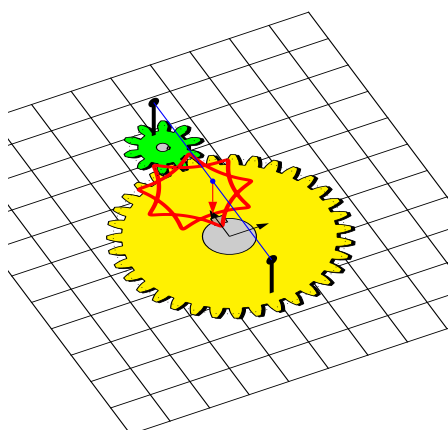
\documentclass{standalone}
\usepackage{animate}

```

```
\usepackage{graphicx}
\begin{document}
\animategraphics[width=0.5\textwidth,controls=all,poster=last]{1}{020}{-}{+}
\end{document}
```

Spouštíme:

```
$ latex 020.tex
$ dvips 020.dvi
$ ps2pdf 020.ps
$ lualatex 020-pstricks.tex
$ lualatex 020-pstricks.tex
```



2.3. Asymptote v2.65

V galerii programu <https://asymptote.sourceforge.io/> je blok animací:

<https://asymptote.sourceforge.io/gallery/animations>

Zaujala mě galerie P. Ivaldiho na <http://asy.marris.fr/asymptote/> s animacemi: <http://asy.marris.fr/asymptote/animations/index.html>.

Zde je ještě jedna galerii <http://www.piprime.fr/asymptote/> s animacemi: http://www.piprime.fr/developpeur/asymptote/animation-asy_asy.

Vybral jsem následující ukázkou.

http://www.piprime.fr/1208/animation_asymptote-fig0090/

Dočasně jsem skrz generování gifů vyhodil bezpečnostní pravidla:

```
$ cd /etc/ImageMagick-6/
$ sudo mv policy.xml policy-old.xml
```

Soubor 030.asy vypadá takto:

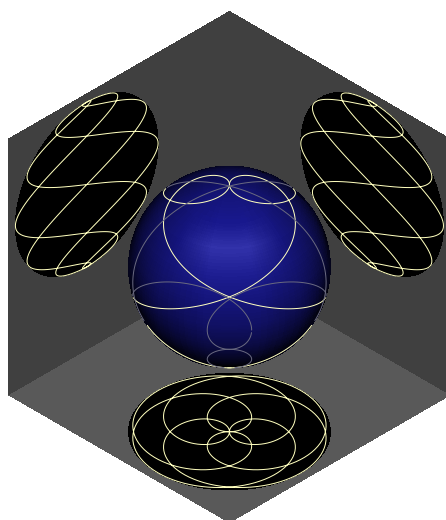
```
size(0,10cm); import graph3; import animation; import solids;
currentlight.background=black; settings.render=0;
animation A; A.global=false; int nbpts=500; real q=2/5; real pas=5*2*pi/nbpts;
int angle=4; real R=0.5; pen p=rgb(0.1,0.1,0.58); triple center=(1,1,1);
transform3 T=rotate(angle,center,center+X+0.25*Y+0.3*Z);
real x(real t){return center.x+R*cos(q*t)*cos(t);}
real y(real t){return center.y+R*cos(q*t)*sin(t);}
real z(real t){return center.z+R*sin(q*t);}
currentprojection=orthographic(1,1,1);
    currentlight=(0,center.y-0.5,2*(center.z+R));
triple U=(center.x+1.1*R,0,0), V=(0,center.y+1.1*R,0);
path3 xy=plane(U,V,(0,0,0)); path3 xz=rotate(90,X)*xy;
path3 yz=rotate(-90,Y)*xy; triple[] P; path3 curve; real t=-pi;
for (int i=0; i<nbpts; ++i){t+=pas; triple
    M=(x(t),y(t),z(t)); P.push(M); curve=curve..M;} curve=curve..cycle;
draw(surface(xy),grey); draw(surface(xz),grey); draw(surface(yz),grey);
triple xyc=(center.x,center.y,0); path3 cle=shift(xyc)*scale3(R)*unitcircle3;
surface scle=surface(cle); draw(scle, black);
draw(rotate(90,X)*scle, black); draw(rotate(-90,Y)*scle, black);
draw(surface(sphere(center,R)),p); triple
    vcam=1e5*currentprojection.camera-center;
for (int phi=0; phi<360; phi+=angle) {bool[] back,front; save();
    for (int i=0; i<nbpts; ++i) {P[i]=T*P[i]; bool test=dot(P[i]-center,vcam)>0;
        front.push(test);}
    curve=T*curve; draw(segment(P,front,operator ..), paleyellow);
    draw(segment(P,!front,operator ..),0.5*(paleyellow+p));
    draw((planeproject(xy)*curve)^(planeproject(xz)*curve)^(
        planeproject(yz)*curve), paleyellow); A.add(); restore();}
A.movie(options="-density 350 -resample 96 -quality 100 -depth 8 -strip");
```

Pomocný soubor je 030-asymptote.tex:

```
\documentclass{article}
\usepackage{animate}
\usepackage{graphicx}
\begin{document}
\animategraphics[width=0.5\textwidth,controls=all,poster=last]{1}{_030+}{0}{89}
\end{document}
```

Spouštíme:

```
$ asy -vk 030.asy
$ for soubor in `find -iname _030\*.eps`; do
> core=${soubor%.eps}
> echo $soubor; ps2pdf $soubor
> pdfcrop -- hires $core; mv $core-crop.pdf $core.pdf
> done
$ lualatex 030-asymptote.tex
$ lualatex 030-asymptote.tex
```



2.4. TikZ v3.1.5b

TikZ si získal nemalou oblibu. Má rozsáhlou dokumentaci.

```
$ texdoc tikz
```

Největší galerie, tzv. T_EXsample, se skládá z příspěvků mnoha uživatelů.

<http://www.texample.net/tikz/examples/tag/animations/>

Zde jsem vybral některé animace z oblasti matematiky a statistiky.

www.texample.net/tikz/examples/sine-and-cosine-functions-animation

www.texample.net/tikz/examples/animated-set-intersection

www.texample.net/tikz/examples/animated-definite-integral

www.texample.net/tikz/examples/convolution-of-two-functions

www.texample.net/tikz/examples/animated-distributions

TikZ samotný však není vhodný nástroj na 3D grafy, neumí skrývat neviděné části, není na to primárně stavěný. S tím do velké míry pomáhá balíček pgfplots, aktuálně ve verzi v1.17, a pomocný balíček pgfplotstable.

```
$ texdoc pgfplots pgfplotstable
```

Za zmínku stojí galerie, sourozenec T_EXsample, server <http://pgfplots.net>. Spojil jsem tyto dvě ukázky, 3D graf a animaci.

pgfplots.net/tikz/examples/bivariate-normal-distribution

tex.stackexchange.com/questions/266125/animate-a-pgfplots-3d-plot

Soubor 040.tex vypadá takto:

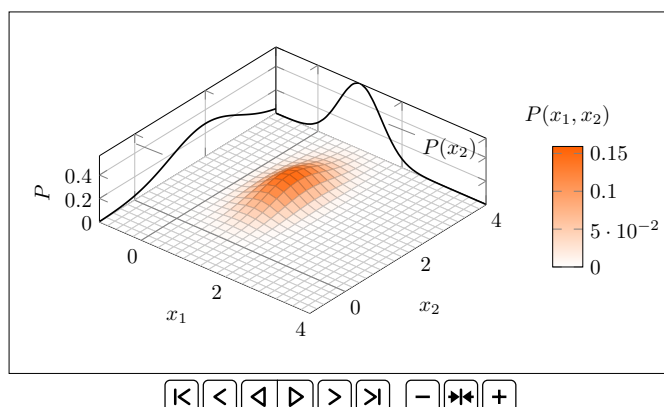
```
\documentclass{article} \pagestyle{empty}
\usepackage{pgfplots}
\pgfplotsset{width=8cm, height=6cm, compat=1.17}
\pgfplotsset{colormap={whitered}{color(0cm)=(white);
color(1cm)=(orange!75!red)}}
\begin{document}
\foreach \malAngle in {40,50,...,400}{\newpage
\begin{tikzpicture}[
declare function = {mu1=1;}, declare function = {mu2=2;},
declare function = {sigma1=0.5;}, declare function = {sigma2=1;},
declare function =
{normal(\m,\s)=1/(2*\s*sqrt(pi))*exp(-(x-\m)^2/(2*\s^2));},
declare function = {bivar(\ma,\sa,\mb,\sb) = 1/(2*pi*\sa*\sb) *
exp(-((x-\ma)^2/\sa^2 + (y-\mb)^2/\sb^2))/2;}]
\draw (-1.5cm,-1cm) rectangle (9.5cm,5cm);
\begin{axis}[colormap name=whitered, view={\malAngle}{65},
enlargelimits=false, grid=none, domain=-1:4, y domain=-1:4, samples=26,
xlabel=$x_1$, ylabel=$x_2$, ylabel=$P$, colorbar, colorbar
style={at={(1.25,0.4)}, anchor=east, height=2cm, title ={$P(x_1,x_2)$}}]
\addplot3 [surf] {bivar(mu1,sigma1,mu2,sigma2)};
\addplot3 [domain=-1:4,samples=31, samples y=0, thick, smooth]
(x,4,{normal(mu1,sigma1)});
\addplot3 [domain=-1:4,samples=31, samples y=0, thick, smooth]
(-1,x,{normal(mu2,sigma2)});
\draw [black!50] (axis cs:-1,0,0) -- (axis cs:4,0,0);
\draw [black!50] (axis cs:0,-1,0) -- (axis cs:0,4,0);
\node at (axis cs:-1,1,0.18) [pin=165:$P(x_1)$] {};
\node at (axis cs:1.5,4,0.32) [pin=-15:$P(x_2)$] {};
\end{axis}
\end{tikzpicture}}
\end{document}
```

Pomocný soubor je 040-tikz.tex:

```
\documentclass{article}
\usepackage{animate}
\usepackage{graphicx}
\begin{document}
\animategraphics[width=0.75\textwidth,controls=all,poster=last]{10}{040}{}{}
\end{document}
```

Spouštíme:

```
$ lualatex 040.tex
$ lualatex 040.tex
$ pdfcrop -- hires 040.pdf
$ mv 040-crop.pdf 040.pdf
$ lualatex 040-tikz.tex
$ lualatex 040-tikz.tex
```



3. Okular v20.04-1: zobrazení animace

Vznik animace je jedna věc, jak je zobrazit v pdf je věc druhá.

Velký problém ve svobodném světě softwaru je, jak takové pdf s animacemi zobrazit. Adobe zrušilo podporu Readeru pro Linux v dubnu 2013 u verze 9.5.5 pro 32bitové počítače. **FoxIt Reader** sice animace vzniklé z balíčku **animate** umí zobrazit, ale také jen mimo Linux. Prakticky stejně je na tom prohlížeč **PDF-XChange Viewer**.

U odlehčených prohlížečů pdf (xpdfreader, mupdf, okular, evince) jsme neměli šanci. Uživatel Linuxu to musí obcházet: míchání 32 a 64bitových aplikací, Wine, přes virtuální stroj či zobrazením pdf na stroji bez Linuxu.

Poměrně velký mezník znamená Google Summer of Code 2019, kdy João Netto rozšiřuje Okular a animace vzniklé přes **animate** lze spustit.

<https://community.kde.org/GSoC/2019/StatusReports/Jo%C3%A3oNetto>

Ani po velkém úsilí, se mi nepodařilo ze zdrojových kódů

```
$ git clone https://cgit.kde.org/okular.git
```

dostat takovou verzi, která by si s tím poradila (Ubuntu 18.04, Ubuntu 20.04, Debian 10). Nepodařilo se mi to ani přes

```
$ sudo snap install --edge okular
```

Můj nejlepší odhad je, že je nevhodná verze knihovny programu Poppler.

Ovšem nahlédneme-li na zařazení nové verze 20.04.1 u distribucí <https://okular.kde.org/download.php> máme vyhráno. Nastartujeme-li Ubuntu 20.10, Arch či Gentoo, vše běží jako po másle přímo z linuxového repozitáře.

Opatrně! Je zde však řešení i pro starší distribuce. Na Xubuntu 18.04 jsem v `/etc/apt/sources.list` přidal

`deb http://cz.archive.ubuntu.com/ubuntu/ groovy main universe`
a ostatní vstupní body jsem si zakomentoval. Pak jsem si vzal na pomoc nástroj `aptitude` a po určité době hledání a řešení konfliktních balíčků se mi podařilo nainstalovat. Sledujte však pozorně, co chce nástroj odinstalovat, aby to nebyla většina linuxové distribuce.

```
$ sudo apt update
$ sudo aptitude install okular
```

Několik postřehů. Animace nejedou přes prezentační režim, ale dá se ze Settings skryt Toolbar, Navigation Panel a Page Bar a přejít do celoobrazovkového režimu přes `Ctrl+Shift+F`.

Po nakliknutí Show Forms úvodní mávající smajlík balíčku `animate` se rozběhne až po zarolování na jinou stranu a zpět. Naopak při Hide Forms zůstává stále aktivní.

Vylepšený Okular nabízí zobrazení pdf, ps, djvu, tiff, chm i formátu epub. Může se hodit i na zobrazení textových souborů, například datových, aux a log souborů při běžné práci. U svých experimentů jej používám i na zobrazení dvi souborů.

Na zobrazení swf či 3D objektů v prohlížeči pdf si ve svobodném softwarovém světě ještě počkáme, doporučuji prozatím Adobe Reader.

4. Vstup do světa xml

Formát xml jako rozšíření html asi netřeba blíže představovat. Vyřešil starší problém strukturování dat nad rámec dat v tabulce řádky krát sloupce na straně jedné a relačních databází na straně druhé. S xml se potkáváme u MathML a především `CONTEX` tomu věnoval velkou pozornost.

<http://pragma-ade.com/show-man-7.htm>

<http://pragma-ade.com/general/manuals/xml-mkiv.pdf>

Zájemce o tuto problematiku odkazují na knihu Dana Lynche z roku 2020 *The Art of Digital Publishing*, <https://mathapedia.com/books/31>, kapitulu 6 The Mathematical Web.

U grafiky přichází formát svg, textový formát pracující v mezích xml. Především program Inkscape zaznamenal velkou oblibu ve světě open source, svg používá jako nativní formát s možností importu a exportu do pdf, včetně možnosti přes příkazový řádek a parametr `--export-pdf`. Ve světě `TeX`u byla grafika vždy trochu pozadu a plní trochu jiné úkoly než na které jsou grafici a animátoři zvyklí. `TikZ` umí načíst svg. Jisté usnadnění dávají balíčky `svg`, `svg-extract`, starší balíček `svg-inkscape` a [tikztosvg](#), v době psaní tohoto článku ještě nebyl zařazen do `TeXLive`.

```
$ texdoc svg svg-extract svg-inkscape
$ firefox https://ctan.org/pkg/tikztosvg
```

Nyní se nám podaří otevřít pdf přes Inkscape, nabídne nám možnosti přes knihovnu Poppler/Cairo či přes upravenou variantu knihovny Poppler. Pokud navolíme Internal import a odškrtneme Replace PDF fonts by closest-named installed fonts, dá se s obrázkem pracovat, byť texty se nedají editovat, jsou z nich křivky.

Jaromír Antoch tuto cestu podrobněji zkoušel a u některých starších příspěvků se text jakoby rozsype. Dávám to za vinu starším písmům ještě v rastrovém formátu. Asi by si to zasloužilo ještě bádání.

TikZ umí vygenerovat svg, viz kapitola 10.2.4 v manuálu verze 3.1.5b, závisí však na nástroji dvisvgm.

```
$ texdoc tikz
```

5. dvisvgm v2.9.1

Na následující testy jsem si připravil zatěžkávací dokument, pracovně soubor 100-pisma.tex. Znaký s diakritikou, rastrové emodži a kousek japonské básně jako zástupce jazyků ČJKV.

```
\documentclass{article}
\usepackage{emoji}
\usepackage{luatexja}
\begin{document}\pagestyle{empty}
\huge\noindent
0, náhlý déšť již zvířil prach a čilá laň teď běží s houfcem gazel
k úkrytům.\emoji{baby}\emoji{sparkling-heart}\emoji{speak-no-evil-monkey}
鳥啼く声す 夢覚ませ 見よ明け渡る 東を 空色染えて 沖つ辺に 帆船群れあぬ 霧の中
\end{document}
```

Jeden ze starších pokusů jak získat svg je nástroj pdf2svg. To bude pro mne srovnávací dokument.

```
$ sudo apt install pdf2svg
```

Spouštíme a dostáváme první obrázek ze čtyř dále v textu.

```
$ lualatex 100-pisma.tex
$ pdf2svg 100-pisma.pdf 100-pisma-pdf2svg.svg
```

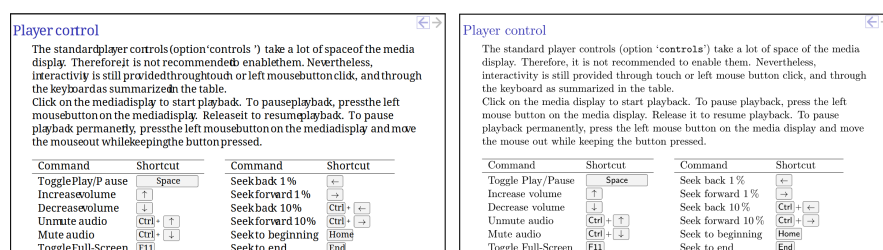
Nástroj dvisvgm má domovskou stránku <https://dvisvgm.de>.

```
$ man dvisvgm
$ info dvisvgm
```

U písem si musíme dát pozor a případně zvolit přepínač `-n` (bez zařazení písem). Zde je ukázka rozdílu při aplikaci na ukázkový dokument z <http://ctan.math.illinois.edu/macros/latex/contrib/media4svg/example/>.

```
$ dvisvgm beamer-example.tex
$ dvisvgm beamer-example.tex
$ dvisvgm --bbox=papersize --font-format=woff2 --zoom=-1 --page=-
  beamer-example.dvi
## dvisvgm -n --bbox=papersize --font-format=woff2 --zoom=-1 --page=-
  beamer-example.dvi
```

Rozdíl mezi 3. a případným 4. příkazem je viditelný. Došlo k náhradě písem a umístění glyfů nesedí. I kdyby vše sedělo, zdrojový kód svg je prakticky ručně needitovatelný.

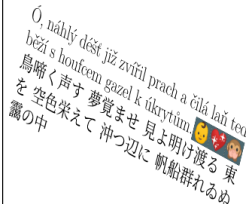


Kdyby nástroj nemohl dohledat písma GhostScriptu, užívá se parametr `--libgs`. Pokud užijeme náhradu písma, je dobré zvolit parametr `-e` na přesný výpočet bounding boxu glyfů. Je to podobné jako u nástroje `pdfcrop`. Můžeme zvolit cestu `tex→dvi→xdv→svg`, ale i `tex→pdf/ps→svg`.

Dokumentace doporučuje užít další parametry: `--font-format=woff` nebo `woff2` na nastavení formátu písma, `--zoom=-1` aby se nezasahovalo do velikosti stran, `--page=1`, - pro volbu všech stran, jinak se bere jen první strana, `--optimize` pro optimalizaci výsledného svg, a případně i `-z / --precision=1` na nastavení počtu desetinných míst. Na zobrazení svg doporučují v prohlížeči Chrome, Chromium a Opera, Firefox se zdá být pomalejší.

U našeho zatěžkávacího dokumentu spouštíme:

```
dvisvgm 100-pisma.tex
dvisvgm -n --zoom=-1 --page=- -o 100-pisma-prespdf1.svg --pdf 100-pisma.pdf
dvisvgm --font-format=woff2 --exact --zoom=-1 --page=- -o
  100-pisma-spismy.svg 100-pisma.dvi
dvisvgm -n --zoom=-1 --page=- -o 100-pisma-prespdf2.svg --pdf
  --transform="R20,w/3,2h/5 T1cm,1cm S2,3" 100-pisma.pdf
```

Ó, náhlý déšť již zvířil prach a čilá laň teď běží s houfcem gazel k úkrytům. 🐘❤️🐘 鳥啼く声す 夢覚ませ 見よ明け渡る 東 を 空色栄えて 沖つ辺に 帆船群れゐぬ 靄の中	Ó, náhlý déšť již zvířil prach a čilá laň teď běží s houfcem gazel k úkrytům. 🐘❤️🐘 鳥啼く声す 夢覚ませ 見よ明け渡る 東 を 空色栄えて 沖つ辺に 帆船群れゐぬ 靄の中
Ó, náhlý déšť již zvířil prach a čilá laň teď běží s houfcem gazel k úkrytům. 🐘❤️🐘 鳥啼く声す 夢覚ませ 見よ明け渡る 東 を 空色栄えて 沖つ辺に 帆船群れゐぬ 靄の中	

První řádek nám vysází dokument do dvi (1. obrázek i identický výsledek nástrojem pdf2svg), druhý převodem písem do vektorových křivek, třetí se pokusí o vysázení s náhradou písem a poslední příkaz je ukázka geometrické transformace celé stránky. Zde jsou náhledy. Je vidět, že nástroj má nějakou závadu u pozadí rastrových emodži, jinak je výsledek uspokojivý při převodu písem do křivek.

6. animate + dvisvgm

Ve světě JavaScriptu se dějí neskutečné věci.

Prezi

<https://www.w3.org/TR/SVG11/animate.html> anime.js svg.js d3.js <https://css-tricks.com/animate-calligraphy-with-svg/>

<https://github.com/plexus/svg-slides> <https://github.com/Moerphy/dizzy.js>
<https://sozi.baierouge.fr>

Za běžných okolností si lze pdf převést na rastrové obrázky a lze s nimi na webu dělat cokoliv. Ale jak přijde na užití hypertextových odkazů, vložení videí a animací, je lepší jít jinou cestou.

Rudolf Blaško se mne ptal, jestli by dokázal svou 2D animaci z Asymptote dostat do animovaného svg. Není tedy na škodu podívat se na možnost vygenerovat animaci z $\text{\TeX}$ u. Představím vám jednu z možných cest spolupráce $\text{\TeX}$ u a JavaScriptu.

Vezmeme druhý obrázek z článku Rudolfa Blaška, animace ve 2D připravená v Asymptote. Pracovně soubor `animacka.tex`, jakože součást knihy.

```
\documentclass{standalone}
\usepackage[inline]{asymptote}
\begin{document}
```

```
\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,6.cm);
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=0*rt*pi,t1=0,t2=t1+tinterval;
draw(circle((0,0),rv),olive+.75); real t1=8.8*pi/3; 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,kx=rp*cos(rm*t/rv),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(0,10*pi,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);
dot(wheelpoint(0),red+black);draw(wheel(0,t1,8*n),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, loop,
palindrome", multipage=false));
\end{asy}
\end{document}
```

Získáme soubor `animacka-1.asy`, když si zavoláme:

```
$ lualatex animacka.tex
```

Tento soubor podsuneme Asymptote:

```
$ asy -vv animacka-1.asy
```

Získáme především soubor `_animacka-1.pdf`.

Připravíme si pomocný soubor `jadro.tex`. Ten nám pomůže s výrobou vrstveného dvi se značkami pro `dvisvgm`.

```
\documentclass[dvisvgm]{standalone}
\usepackage[palindrome, controls=all]{animate}
\usepackage{graphicx}
\begin{document}
```

```
\animategraphics{8}{_animacka-1}{1}{8}
\end{document}
```

Spustíme:

```
$ dvisvgm jadro.tex # nebo: lualatex --output-format=dvi jadro.tex
$ dvisvgm jadro.tex
```

Vzniká nám soubor `jadro.dvi`. Ten už převedeme do `svg`.

```
$ dvisvgm --exact --zoom=-1 --page=- jadro.dvi
```

Ten již můžeme otevřít, např. přes

```
$ firefox jadro.svg
$ google-chrome jadro.svg
$ chromium jadro.svg
$ opera jadro.svg
```

Pokud bychom si naopak přáli zařadit `jadro.svg` na webovou stránku, musel by vypadal jako v tomto pracovním souboru `webovka.html`:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
  <object width="400px" type="image/svg+xml" data="jadro.svg"></object>
</body>
</html>
```

7. svganimation

Je tu ještě jiná možnost. A to získat sérii nezávislých `svg`, jeden `svg` soubor vzniklý z jednoho snímku či jedné strany `pdf` dokumentu. To bychom u naší ukázky předchozí kapitoly získali z mnohastránkového `pdf` takto:

```
$ dvisvgm --pdf --exact --zoom=-1 -o "%f-%0p" --page=- _animacka-1.pdf
```

Parametr `-o` nám zajistí název souboru bez dodatečných nul. Nástrojů bychom našli nespočet, mě zaujal projekt na Syracuse:

<https://melusine.eu.org/syracuse/G/svganimation>

Zde je několik ukázek:

<https://melusine.eu.org/syracuse/G/svganimation-exemples>

První úkol je nástroj stáhnout. Lze to přes tlačítko *tree snapshot* z

<https://melusine.eu.org/syracuse/G/git/?p=svganimation.git;a=tree>

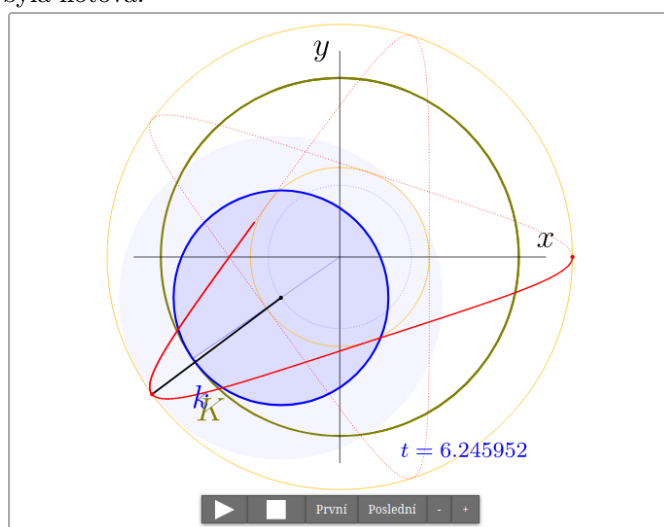
Pro automatizéry z příkazového řádku:

```
$ curl -o svganimation.tgz "https://melusine.eu.org/syracuse/G/git/\
?p=svganimation.git;a=snapshot;h=HEAD;sf=tgz"
# nebo místo curl -o užít wget -O
$ tar xvf svganimation.tgz
$ cd svganimation-HEAD-9fed6b5/ # v mém konkrétním případě
```

Nahlédl jsem na ukázky, vytvořil potřebné složky a nakopíroval 91 souborů `_animacka-1-*.svg`.

```
$ cd ellipsographe/
$ mkdir animacka-1
$ cp animation.html animacka-1.html
$ cp <zdroj>/_animacka*.svg animacka-1/
```

V souboru `animacka-1.html` jsem upravil co bylo potřeba: popisky, otevření prvního souboru, rozsah animace, rychlost ap. Upravit popisky se dá v souboru `../SVGPlayerOne.js`. Zde je ukázka. Kvůli rozsahu zdrojový kód nepřikládám. Asi by stačila knihovna `jinja2` v Pythonu a obecná šablona by za chvíli byla hotová.



Koho by tato oblast zajímala víc, nechť jsou takové osobě inspirací nápady na <https://tex.stackexchange.com/questions/473936>.

8. *Hello, world!* od balíčku `media4svg` v0.4

Jeden z posledních experimentů v T_EXovém světě je balíček `media4svg`, který umožňuje při exportu do svg uložit audio a videostopy. Dokumentace je ještě

v textové formě, nikoliv v pdf. Jedná se o jistý pokus generování snímků jako u pdf přes `beamer` nebo nástroj typu `powerline`.

```
$ youtube-dl -o linus.mp4 https://www.youtube.com/watch?v=CYvJPra7Ebk  
$ ffmpeg -i linus.mp4 -vn linus.mp3
```

```
deformace i s videem jako perlička Perlička. dvisvgm -n -transform="R20,w/3,2h/5  
T1cm,1cm S2,3" export-media.dvi
```

9. Náhled na interaktivitu závěrem: `latex2html5`

Když opomineme GeoGebru a další vhodné nástroje, zde je zajímavý experiment, na který bych rád poukázal.

<https://tex.stackexchange.com/questions/145262/how-to-create-pstricks-animations-on-the-web-page-with-interactive-control-panel> <https://github.com/pyramation/LaTeX2JS>

Ono asi mělo dojít na nápad Petra Olšáka zmíněný na jedné konferenci `TeXperience`, že by měl `TeX` převést do `C++` knihoven. Škoda, že se takový nápad a podobné pokusy neuchytily, pomohlo by programátorským polyglotům v přechodech mezi `TeXem` a dalšími jazyky.

PRVNÍ KONTAKT S PROGRAMEM MANIM THE FIRST ENCOUNTER WITH MANIM

Pavel Stríž

E-mail: pavel@striz.cz

Abstrakt: Tento článek tvoří úvod do práce s programem Manim (matematické animace) založeném na Pythonu s podporou $\text{\TeX}$. Jedná se především o řešení zdrojů, tipy, triky a řešení některých problematických partií. Program vytváří a spravuje Grant Sanderson alias 3blue1brown a Ben Eater. Původně to byl soukromý projekt pomáhající jim programově vytvořit náročnější videa, nyní se jedná o otevřený software.

Klíčová slova: Animace, Manim, Python, $\text{\TeX}$.

Abstract: This article is an introduction to work with Manim software (Mathematical Animation Engine), which is a Python-based program with support of $\text{\TeX}$. The paper consists mainly of research of sources, tips, tricks and solution to some problematic parts. Software is being developed and maintained by Grant Sanderson alias 3blue1brown and Ben Eater. It was initially a private project supporting them programmatically create complex videos. Now, it's an open-source software.

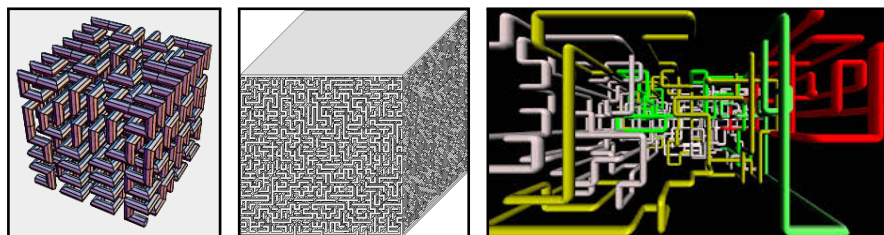
Keywords: Animation, Manim, Python, $\text{\TeX}$.

*Motto: Python se má rád, je to tam jedno *self*í za druhým!*

1. Mohou za to kvaterniony

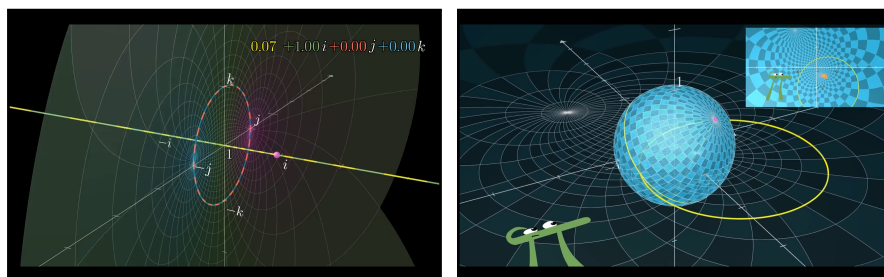
Někdy v roce 2018 jsem otevřel jeden ze starých problémů, a to jak vykreslit nekonečnou trubku, ale tak, aby se vzájemně nekřížila. Je to trochu obdoba [Hilbertovy křivky](#) (vlevo) či tvorba bludišť, např. v programu [Daedalus](#) (uprostřed). Zájemce odkazují na [demo](#) (Xubuntu 20.04, obrázek vpravo):

```
$ sudo apt install xscreensaver xscreensaver-gl  
$ xscreensaver &  
$ sleep 8; xscreensaver-command -demo 39
```



Podařilo se mi to vyřešit v **Blenderu**, ale tehdy dokumentace ke **kvaternionům** (rozšíření komplexních čísel pro 3D) byla strohá, hledal jsem doplňující materiály. Zaujalo mě video [youtube.com/watch?v=d4EgbgTm0Bg](https://www.youtube.com/watch?v=d4EgbgTm0Bg) s detaily na eater.net/quaternions, ale co víc, na **fóru** byla zmínka, že video bylo vytvořené v programu **Manim**. Tak jsem se do toho víc ponořil, neb mi název programu nic neříkal.

Zde je pár ukázek od tvůrců ze zmíněného videa:

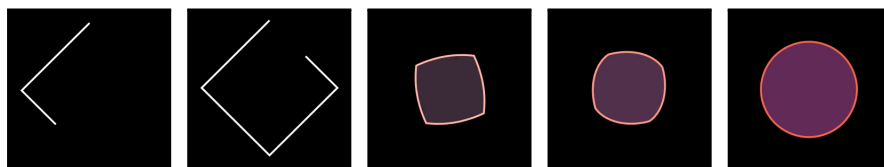


2. Hello World! aneb Manim animuje čtverec na kolečko

Jedná se o vyvíjený program na matematické animace a je docela programátorský oříšek dát vše do latě (instalaci, rozběhnutí ukázek, vlastní tvorba). Bral jsem to však za součást učení se.

Program lze nainstalovat přes **pip3** (**manimlib**), **virtualenv**, **docker**, **anaconda**, ale hlavně přímo. Detaily jsou na github.com/3b1b/manim.

```
$ git clone https://github.com/3b1b/manim.git
$ cd manim
$ sudo -H pip3 install -r requirements.txt
$ python3 manim.py example_scenes.py SquareToCircle -pl
```



Pokud vše proběhne v pořádku spustí se vám video, kdy se bílý čtverec změní na barevné kolečko a to pak zmizí.

Problém jsem měl s instalací knihovny **pycairo**, to jsem vůči manuálu vyřešil přes:

```
$ sudo apt install python3-cairo
$ sudo -H pip3 install manimlib --ignore-installed pycairo
```

Druhý řádek mi v roce 2020 nejel, stále se mi snažil vnutit `pycairo`, a to při instalaci spadne, tak jsem užil první řádek z předchozího skriptu, v `requirements.txt` jsem zakomentoval řádek s `pycairo` a doinstaloval jen další závislosti:

```
$ sudo -H pip3 install -r requirements.txt
```

Touto cestou užívám starší `pycairo` verze 1.16.2 a Manim z pracovního adresáře, vše dále představené běží.

3. Hlouběji u spuštění z příkazového řádku

Nápovědu lze získat přes `python3 -m manim --help`. Ukázky jsou realizovány přes třídy, jejich seznam jsem si nechal vypsát přes `cat <soubor.py> | grep <třída či class>`. Obvykle vidíme `Scene`, `GraphScene`, `ThreeDScene` a `SVGMOBJect`.

Vytvoření videa se realizuje přes:

```
python3 -m manim <soubor.py> [třídy] [parametry]
```

Třídy oddělujeme mezerami. Bez zadání třídy vyběhne nabídka a zadáváme čísla oddělená čárkou. Ne všechny třídy jsou takto spustitelné.

Nejběžnější volitelné parametry jsou následující:

- `-p` = preview, otevření souboru po dokončení,
- `-l` | `-m` | bez parametru = velikost videa, low | medium | high,
- `-t` = transparent, pozadí bude průhledné,
- `-c` = color, pozadí bude mít specifickou barvu,
- `-r` = resolution, rozlišení v pixelech, výška čárka délka,
- `-s` = save the last frame, uloží z videa jen poslední snímek,
- `--livestream` | `--to-twitch` = živé vysílání,
- `-h` | `--help` = nápověda.

Výsledky se ukládají do adresáře `media/video`, pak do složky dle názvu skriptu, následuje složka výška `p` rychlost, např. `480p15` znamená 480 pixelů je výška obrazu v rychlosti 15 snímků/vteřinu. Vzniká série malých videí, které se v závěru spojí do velkého souboru `mp4`.

Generované $\text{\TeX}$ ové soubory, `xdv` a `svg` lze nalézt v adresáři `media/TeX`.

4. Inspirativní zdroje

Mezi tutoriály v angličtině řadíme talkingphysics.wordpress.com/2019/01/08/getting-started-animating-with-manim-and-python-3-7 s pod-

Kdo dává přednost videotutoriálům, nechť nahlédne na youtube.com/channel/UCxiWCEdx7aY88bSEUgLOC6A s podpůrnými soubory na github.com/Elteoremadebeethoven/AnimationsWithManim.

V češtině v akad. roce 2017/2018 proběhl seminář od Mirka Olšáka, <http://www.olsak.net/mirek/manim/>, od března 2020, možná v souvislosti s koronavirem, vznikají překlady anglických videí, www.youtube.com/channel/UCIhWS2rX780XidZ87QLkxkA.

Představím některé běžné situace a jejich řešení.

Náš kód v souboru `ukazky/ahoj-svete.py` by mohl vypadat takto:

```
from manimlib.imports import *
class AhojSvete(Scene):
    def construct(self):
        svetu = TextMobject("Ahoj, světe!")
        self.play(Write(svetu))
        self.wait()
```

Spouštíme: `python3 -m manim ukazky/ahoj-svete.py -p`



5.2. Lze zařadit **TikZ**?

Lze, ale síla linek se v svg nebere v potaz. Chce to experimentovat u konkrétního kódu. Zde je ukázka animace pro divadelní hru v souboru `ukazky/modalive.py`, která nebyla použita, sloužila k testovacím účelům obrazovek a monitorů. Je vidět složitější animace, vsunutí textu do TikZu, u proměnných užívám `r""`, abych nemusel zdvojovat zpětná lomítka.

```
from manimlib.imports import *
class Modalive(Scene):
    def construct(self):
        text=r"\begin{tikzpicture}\node[draw=none]{Móda
        live!};\end{tikzpicture}"; modalive=TextMobject(text);
        modalive.scale(6); obdelnik=[""]*5
        obdelnik[0]=TextMobject(text); obdelnik[0].scale(2);
        obdelnik[0].to_edge(DOWN); obdelnik[0].set_color(YELLOW);
        obdelnik[0].rotate(PI)
        obdelnik[1]=TextMobject(text); obdelnik[1].scale(2);
        obdelnik[1].to_edge(UP); obdelnik[1].set_color(YELLOW)
        obdelnik[2]=TextMobject(text); obdelnik[2].scale(2);
        obdelnik[2].to_edge(LEFT); obdelnik[2].set_color(YELLOW);
        obdelnik[2].rotate(PI/2)
        obdelnik[3]=TextMobject(text); obdelnik[3].scale(2);
        obdelnik[3].to_edge(RIGHT); obdelnik[3].set_color(YELLOW);
        obdelnik[3].rotate(-PI/2)
        self.wait(1); self.play(Write(modalive),run_time=20); self.wait(1);
        self.play(FadeOut(modalive),run_time=5); self.wait(1)
        for x in [1,3,0,2]: self.play(Write(obdelnik[x]), run_time=2)
        modalive.set_color_by_gradient(RED,BLUE); self.wait(1);
        self.play(GrowFromCenter(modalive), run_time=5); self.wait(1);
        self.play(Transform(obdelnik[0], modalive),Transform(obdelnik[1],
        modalive), run_time=3); self.wait(1); self.play(Transform(obdelnik[2],
        modalive), Transform(obdelnik[3], modalive), run_time=3)
        for x in range(4): self.remove(obdelnik[x])
        self.play(FadeOut(modalive), run_time=5); self.wait(1);
        srdce=TextMobject(r"$\varheartsuit$"); srdce.scale(20);
```

```
srdce.set_color(RED); srdce.rotate(-45);
self.play(GrowFromCenter(srdce), run_time=5); self.play(FadeOut(srdce),
run_time=2); self.wait(1); self.play(ShowCreation(modalive),
run_time=40); self.wait(1); self.play(FadeOut(modalive), run_time=5);
self.wait(1)
```

Animace: `python3 -m manim ukazky/moda-live.py -p`



5.3. Lze animovat matematiku?

Pozornější $\text{\TeX}$ isté si všimli, že v preambuli šablony je balíček `halloweenmath`, nyní jej použijeme v souboru `ukazky/rovnice.py`.

```
from manimlib.imports import *
class Rovnice(Scene):
    def construct(self):
        sum1=TextMobject(r"$$$sum_{i=0}^n {n\choose i} a^{n-i} b^i = (a+b)^n$$$")
        sum1.scale(2); sum1.to_edge(UP); sum1.set_color(WHITE);
        self.play(Write(sum1), run_time=3); self.wait(1)
        sum2=TextMobject(r"$$$mathwitch*_{i=0}^n {n\choose i}
\mathrightghost^{n-i} \mathleftghost^i \xrightwiconbroom*{ }
(\mathrightghost+\mathleftghost)^n$$$")
        sum2.scale(1.75); sum2.to_edge(DOWN); sum2.set_color(WHITE)
        self.play(ReplacementTransform(sum1.copy(), sum2), run_time=6);
        self.wait(4)
```

Animace: `python3 -m manim ukazky/rovnice.py -p`



5.4. Lze animovat kandži?

Asi nejrychlejší způsob je si v `manimlib/constants.py` zapnout `Chinese $\text{\TeX}$` :

```
TEX_USE_CTEX = True # False
```

Tím si zajistíme, že budeme užívat `xelatex` a $\text{\TeX}$ ovou šablonu v souboru `manimlib/ctex_template.tex`. Tu jsem si po odzálohování upravil do této podoby:

```
\documentclass[preview]{standalone}
\usepackage{amsmath,amssymb}
\usepackage[UTF8]{ctex}
\begin{document}
YourTextHere
\end{document}
```

Jednoduchá ukázka by mohla vypadat takto:

```
from manimlib.imports import *
class Japonstina(Scene):
    def construct(self):
        sum1=TextMobject(r"今日は何曜日ですか。")
        sum1.scale(2.5); sum1.to_edge(UP); sum1.set_color(RED);
        self.play(Write(sum1), run_time=3); self.wait(1)
        sum2=TextMobject(r"3日です。")
        sum2.scale(6); sum2.to_edge(DOWN); sum2.set_color(RED)
        self.play(ReplacementTransform(sum1.copy(),sum2),run_time=5); self.wait(2)
```

Animaci získáme přes: `python3 -m manim ukazky/japonstina.py -p`



5.5. Lze animovat výstupy z teorie grafů?

Na pomoc jsem si vzal knihovnu `manimnx` a vypnul jsem si užití C_TE_Xu:

```
TEX_USE_CTEX = False
```

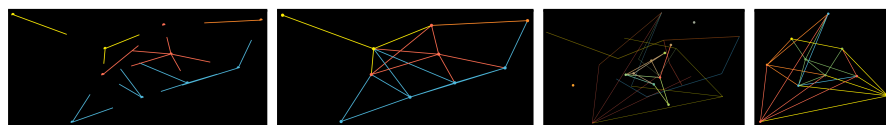
Doinstaloval jsem si potřebné:

```
$ git clone https://github.com/rajatvd/manimnx
$ sudo -H pip3 install networkx==2.3
```

V `manimnx/example.py` jsem zasáhl do jednoho řádku tímto způsobem, protože `manim.py` načítám z pracovního adresáře:

```
import manimnx.manimnx.manimnx as mnx
```

Ukázka: `python3 -m manim manimnx/example.py RandomGraphs -p`



5.6. Lze animovat diagramy?

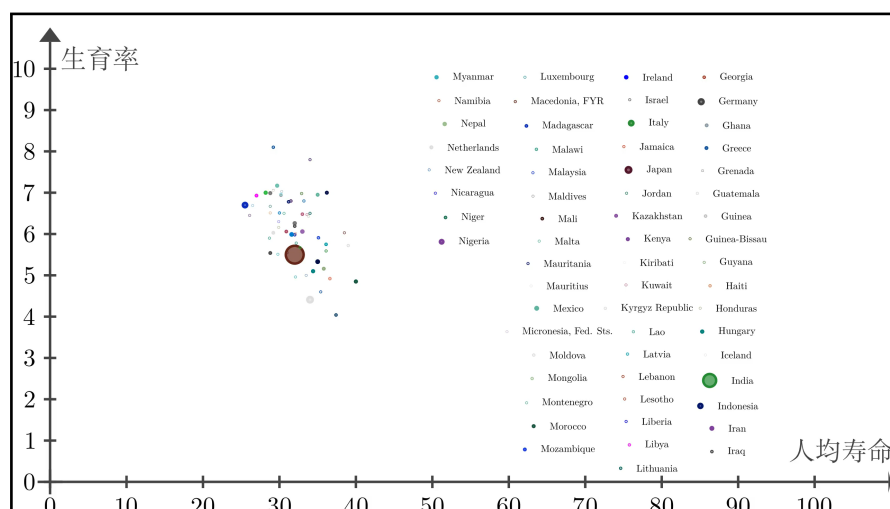
Jako poslední ukázkou jsem vybral extrémní případ knihovny **Danim**.

```
$ git clone https://github.com/graviton1221/Danim
$ sudo -H pip3 install pandas
```

V `Danim/BubbleChart/bubblechart.constant.py` jsem `\\` upravil na `/`, jak pracuji pod Linuxem. A ještě jednou jsem si zapnul $\text{\LaTeX}$.

Každý popis se generuje do zvláštního $\text{\TeX}$ ového souboru, vznik animace tedy trvá: `python3 -m manim Danim/BubbleChart/BubbleChartAnimation.py BubbleChartAnimation -p`

Ukázka z 19. vteřiny videa v inverzních barvách kvůli tisku bulletinku.



6. Co s programem Manim neumím?

Co jsem nepotřeboval a nezkoušel jsem do hloubky je živý přenos přes protokol tcp (parametr `--livestream`) s podporou pro Twitch přes parametry `--to-twitch` a `--with-key`. Hashovací klíč jsem našel na twitch.tv pod uživatelem; Settings; Channel and Videos a pak Primary Stream key (Copy nebo Show).

Práce se soubory svg je nativní, zde jsou oblíbené figurky přednášejících (PiCreature, Stickman a Linus). Lze si vytvořit vlastního průvodce. Nezkoušel jsem, byť návod existuje, <https://talkingphysics.wordpress.com/2018/08/14/working-with-svg-files-manim-series-part-12/>.



Vzniká podpora pro JavaScript, viz github.com/JazonJiao/Manim.js. Přes JavaScript bych na animace volil spíš d3js.org.

Vzniká též webové rozhraní, viz github.com/eulertour/eulerv2. To mi přišlo ve webovém prohlížeči extrémně pomalé.



Pokud chcete nahlédnout na animace zmíněné v článku, a programátorské a $\text{\TeX}$ ové věci si nezkoušet, navštivte můj skromný kanál na YouTube:

youtube.com/playlist?list=PLnD-4Ssyh8y0Qi08n9L8WJ3wV3u2p9V09



Co zmínit závěrem? Je to jiný svět, snad jsem vám hlavu nezamotal víc než je zdrávo, mám totiž před sebou víc otázek než odpovědí, i tak však přeji hezké bádání s nástrojem Manim!

- Můžeme užít [Lua \$\text{\LaTeX}\$](#) a libovolné písmo přes balíček [fontspec](#)?
- Lze získat animaci jako sérii souborů svg?
- Může se objekt pohybovat po libovolné křivce, např. po spirále?
- Je možné (de)aktivovat vyhlazování písem?
- Jak lze ideálně zařadit grafiku z programů [METAPOST](#) a [PSTricks](#)?
- Jak by to bylo se zařazením 3D objektů ([Asymptote](#), [Blender](#))?
- Lze vložit grafické výstupy z programů typu [Matplotlib](#) a [R](#)?
- Lze vložit do vznikající animace video? Například ve [videu o kvaternionech](#) je pohyb ruky, tedy to lze, ale jak ...

Je vidět, že [3blue1brown](#) je učitel tělem i duší a nutí nás přemýšlet a programátorsky hledat a experimentovat. Ne nadarmo byl 13. 3. 2020 Grant Sanderson pozván jako řečník do TEDxBerkeley (https://youtube.com/watch?v=s_L-fp8gDzY) s tématem *What Makes People Engage With Math*, minimálně jako duševní kantorská podpora v době koronaviru.

... *if you have a soul, you have to know why, right?*
— Grant Sanderson @ [00:13:04,159265](#)

TEXLIVE 2020: NOVINKY V TEXOVÉM SVĚTĚ TEXLIVE 2020: NEWS IN THE WORLD OF TEX

Pavel Stríž

E-mail: pavel@striz.cz

Motto: *In the Beginning Was the Number*
Jean-Côme Charpentier @ TEXový balíček [xlop](#)

1. Instalace

Už mnoho let používám [TEXLive](#) na [Xubuntu](#) a snažím se každý rok o novou instalaci. Vše vyzkoušet a prozkoumat.

Z webové stránky <http://tug.org/texlive/acquire-netinstall.html> či přímo si stáhnou a rozbalím instalační skript do pracovního adresáře.

```
$ wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
$ tar xvf install-tl-unx.tar.gz
$ cd install-tl-20200425 # změnit na aktuální časovou známku
$ ./install-tl
```

Obvykle nemám žádný problém a instaluji, u této verze se mi nepodařilo aktivovat `tlmgr update`, tak jsem si před instalací navolil adresář `~/texlive/2020`.

Po instalaci se rozšiřují či upravují systémové cesty (`MANPATH`, `INFOPATH` a především `PATH`), obvykle v souboru `~/bashrc`. Po úpravě souboru si volám `source ~/bashrc`, případně si otevřu nové terminálové okno.

Ověřujeme spustitelnost přes `which tex` nebo `tex --version`.

Aktualizace balíčků se realizuje přes `tlmgr update --self --all`.

Dokumentace balíčku se volá přes: `texdoc <balíček>`.

Je svátek 8. 5. 2020 a [TEXLive 2020](#) mi nainstaloval 3999 balíčků.

2. Novinky

Není možné podchytit všechny novinky, ale přecejenom některé balíčky vyčnívají či by mohly pomoci.

ConTEXt je samostatná kategorie, viz <https://wiki.contextgarden.net>, za LuaTEX sleduji <https://ctan.org/topic/luatex> a za L^AT_EX3 pak <https://ctan.org/topic/exp13>. Nové zprávy na ctan.org lze sledovat na [ctan-ann](https://ctan.org/ann), témata jsou roztržena na ctan.org/topics/highscore. Reálné TEXové problémy a odpovědi TEXistů hledejte na komunitním serveru <https://tex.stackexchange.com/> (zkracováno TEX.SE).

2.2. Balíček **witharrows**

Od Pantignyho vznikl ještě jeden podobně zaměřený balíček. Tento je vhodný na popis úprav matematických vztahů. V dokumentaci je řada překlepů, chce to ještě vychytat.

```
$ texdoc witharrows

%! lualatex mal-witharrows.tex
\documentclass{article} % se standalone balíček zlobil
\pagestyle{empty}
\usepackage{witharrows}
\begin{document}
\def\ee{\mathrm{e}} \def\ii{\mathrm{i}}
\begin{DispWithArrows*}[displaystyle, wrap-lines]
S_n &= \frac{1}{n} \operatorname{Re}
\left( \sum_{k=0}^{n-1} \operatorname{bigl} \operatorname{ee}^{\operatorname{ii} \frac{\pi}{2n}} \operatorname{bigr}^k \right)
\Arrow{\upshape sum of terms of a geo-metric progression of ratio
\operatorname{ee}^{\operatorname{ii} \frac{2\pi}{n}}}
&= \frac{1}{n} \operatorname{Re} \left( \frac{1 - \operatorname{bigl} \operatorname{ee}^{\operatorname{ii} \frac{\pi}{2n}} \operatorname{bigr}^n}{1 - \operatorname{ee}^{\operatorname{ii} \frac{\pi}{2n}}} \right)
\Arrow{\upshape This line has been wrapped automatically.}
&= \frac{1}{n} \operatorname{Re} \left( \frac{1 - \operatorname{ii} \{1 - \operatorname{ee}^{\operatorname{ii} \frac{\pi}{2n}}\}}{1 - \operatorname{ee}^{\operatorname{ii} \frac{\pi}{2n}}} \right)
\end{DispWithArrows*}
\end{document}
```

$$\begin{aligned}
 S_n &= \frac{1}{n} \Re \left(\sum_{k=0}^{n-1} \left(e^{i \frac{\pi}{2n}} \right)^k \right) \\
 &= \frac{1}{n} \Re \left(\frac{1 - \left(e^{i \frac{\pi}{2n}} \right)^n}{1 - e^{i \frac{\pi}{2n}}} \right) \\
 &= \frac{1}{n} \Re \left(\frac{1 - i}{1 - e^{i \frac{\pi}{2n}}} \right)
 \end{aligned}
 \begin{array}{l}
 \left. \begin{array}{l} \text{sum of terms of a geo-} \\ \text{metric progression of} \\ \text{ratio } e^{i \frac{2\pi}{n}} \end{array} \right\} \\
 \left. \begin{array}{l} \text{This line has been} \\ \text{wrapped automatically.} \end{array} \right\}
 \end{array}$$

2.3. Balíček **siunitx**

Přes balíček **nicematrix** jsem narazil na balíček **siunitx** od **Josepha Wrighta**. Užíval jsem balíčky **siunits** a **pgfplotstable**, tohle je pravděpodobný nástupce na sazbu jednotek a tabulek s čísly.

```
$ texdoc siunitx siunits pgfplotstable

%! lualatex mal-siunitx.tex
\documentclass[varwidth]{standalone}
\usepackage{cancel}
\usepackage[binary-units]{siunitx}
\DeclareSIPrePower\quartic{4}
\DeclareSIPostPower\tothefourth{4}
```

```
\begin{document}
\num{1+-2i}, \num{.3e45}, \si{kg.m.s^{-1}},
\si{\kilogram\metre\per\second}, \si{\joule\per\mole\per\kelvin},
\si[per-mode=fraction]{\cancel\kilogram\metre\per\cancel\kilogram\per\second},
\si{\kilogram\tothefourth}, \si{\quartic\metre}, \par \num{1e2/3e4},
\ang{6;7;6.5}, \ang[angle-symbol-over-decimal]{45.697},
\SI{100}{\mebi\byte}, \SI[prefixes-as-symbols=false]{30}{\kibi\bit},
\SI{1.234}{\metre}, \SI[locale = DE]{6.789}{\metre}
\end{document}
```

$1 \pm 2i$, 0.3×10^{45} , kg m s^{-1} , kg m s^{-1} , $\text{J mol}^{-1} \text{K}^{-1}$, $\frac{\text{kg m}}{\text{kg s}}$, kg^4 , m^4 ,
 $1 \times 10^2 / (3 \times 10^4)$, $6^\circ 7' 6.5''$, $45^\circ 697$, 100 MiB, 30×2^{10} bit, 1.234 m, 6,789 m

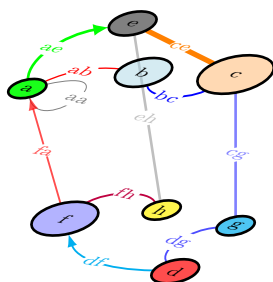
2.4. Balíček **tikz-network**

Na sazbu obrázků z teorie grafů existuje nespočet nástrojů, např. **tkz-graph**. U tohoto balíčku od **Jürgena Hackla** mne zaujaly vrstvy ve 3D. V pracovním adresáři jsem si nalinkoval pomocné soubory a ukázky se rozběhly.

```
$ mkdir data
$ cd data
$ ln -s <cesta>/texmf-dist/doc/latex/tikz-network/data/ml_{vertic,edg}es.csv .
$ cd ..

$ texdoc tikz-network

%! lualatex mal-tikz-network.tex
\documentclass{standalone}
\usepackage{tikz-network}
\begin{document}
\begin{tikzpicture}[multilayer=3d]
\Vertices{data/ml_vertices.csv}
\Edges{data/ml_edges.csv}
\end{tikzpicture}
\end{document}
```



Autorem je **Jean-Côme Charpentier**. Balíček nám pomáhá se sazbou základních aritmetických operací a schémat. Zdeněk Wagner mi psal, že autor postrádá v dokumentaci informaci, že schéma pro násobení, které se stále učíme na základních školách, vytvořil někdy v 8. století podle indických knih perský matematik **ابو عبد الله محمد ابن موسى الخوارزمي جعفر**, krátce **Al-Chorezmí**. Autorovi jsem postřeh napsal. Na pomoc s arabštinou jsem si vzal balíček **arabluatex** od **Roberta Alessiho** s renovovaným písmem **Amiri** od **Khaleda Hosnyho**.

\$ texdoc xlop arabluatex amiri

$$124 \div 7 \approx 17.71428571, \quad 124 = 7 \times 17 + 5, \quad 150 \div 7 \approx 21.\overline{428571} \dots$$

$$\begin{array}{r} \\ \\ + \\ \\ \hline \\ \\ \hline \end{array}$$

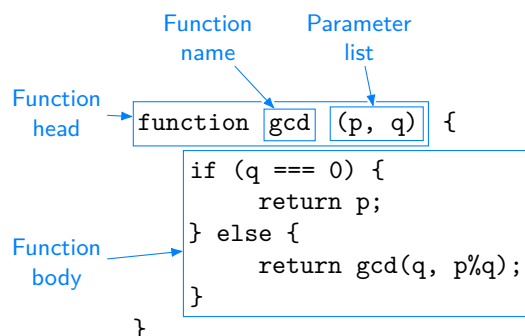
Již v dobách ranných bylo možné najít typografické vychytávky na sazbu
algoritmů, zdrojových kódů a pseudokódů. Tento balíček zvýrazňuje části
kódu s možností je popsat. TeXujeme dvakrát. Autorem je **Hồng-Phúc Bùi**.

```

%! lualatex mal-codeanatomy.tex
\documentclass{standalone}
\usepackage{codeanatomy}

```

```
\begin{document}
\begin{tikzpicture}[remember picture]
\codeBlock{%
\cPart{functionHead} {function \cPart{functionName}{gcd}
\cPart{paramList}{(p, q)}} \{ \}[2.5pt]
\ptab{\mtPoint{mostLeft}}{if (q == 0) \{ \}
\ptab{\ptab{}} return p; \}
\ptab{\} else \{ \}
\ptab{\ptab{}} return gcd(q, p%q);\extremPoint{mostRight} \}
\ptab{\mbPoint{mostBottom}}{\} \}
\}% end od \codeBlock
\fitExtrem{functionBody}{(mostLeft) (mostRight) (mostBottom)}
\codeAnnotation{functionHeadText}{-1,3}{Function\\head}
\codeAnnotation{functionBodyText}{-1,1}{Function\\body}
\codeAnnotation{functionNameText}{( 1,4){Function\\name}
\codeAnnotation{paramListText}{( 3,4){Parameter\\list}
\draw[->,annotation] (functionHeadText) -- (functionHead);
\draw[->,annotation] (functionBodyText) -- (functionBody);
\draw[->,annotation] (functionNameText) -- (functionName);
\draw[->,annotation] (paramListText) -- (paramList);
\end{tikzpicture}
\end{document}
```



2.7. Balíček **mercatormap**

V roce 2018 na konferenci OSSConf v Žilině Aleš Kozubík představil z pohledu uživatele balíček **getmap**. Ten pracuje s **OpenStreetMap**. Tehdy to byl nový balíček i pro mne a příjemné překvapení. Letos jsem organizátory předběhl, protože jsem jako první objevil tento balíček. Je to cenné hlavně z pohledu propojení dvou sekcí: T_EXové a GISácké. Autorem je **Thomas F. Sturm**.

Je potřeba mít Python3 a několik balíčků, v mém případě to bylo:

```
$ sudo apt install python3
$ sudo -H pip3 install Pillow requests
```

Přes tento balíček se mi nepodařilo získat mapy z mapy.cz ani ze serveru freemap.sk. V pozadí se očekává na dotaz png soubor, obdrží html. Český server sice API má na api.mapy.cz, ale nikoliv s touto možností. Slovenský server také umí, ale k png se musí člověk proklikat v rámci exportu mapy. Napsal jsem to vývojářům jako tip na rozšíření, kdyby se náhodou nudili, neb minimálně Sturm v dokumentaci píše, že rád nový mapový server do dokumentace svého balíčku zařadí.

Přikládám mapovou ukázkou, v poznámkách v kódu je nefunkční část rozhraní na mapy.cz, na slovenský server by to bylo obdobné, to pro případ, že by to v budoucnu fungovalo. Je potřeba $\TeX$ ovat s parametrem `--shell-escape` (Unix), případně `--enable-write18` (Microsoft Windows). Za běhu se dočasné soubory ukládají do složek `maps` a `tiles`.

```
$ texdoc mercatormap getmap
```

```
%! lualatex --shell-escape mal-mercatormap.tex
\documentclass{standalone}
\usepackage{mercatormap}
\mrcmapset{python=python3}
\mrcactivatescript
\begin{document}
\begin{tikzpicture}
\mrcmap[type=reference, latitude=49.14549, longitude=16.99913, flex reference
    scale=250000, source=topplusopen p250, target=wmsmap, tex
    width=0.9\linewidth, tex height=3cm]{mapa-bucovice}
\mrcdrawmap
\node[below, font=\bfseries\sffamily] at (mrcmap.south) {Bučovice, rodiště
    autora zprávy};
\end{tikzpicture}
%\mrcnewsupplysource{mapycz}{
% url={https://en.mapy.cz/zakladni?x={x}&y={y}&z={z}},
% attribution={mapycz},
% attribution print={mapycz},
% basename=tiles/mapycz}
\end{document}
```



Bučovice, rodiště autora zprávy

2.8. Balíček **tcolorbox**

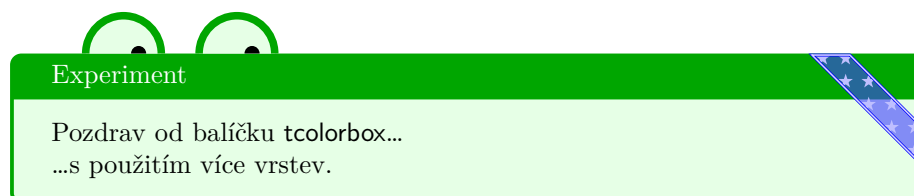
Storm v dokumentaci balíčku **mercatormap** masivně používá tento balíček, jehož je i autorem. Ačkoliv balíček znám a je vhodný především do prezentací, různých poznámek do knih a skript, na plakátky a obálky knih, přece jenom na něm autor dále pracuje a stojí za připomenutí. Já jsem si z obří dokumentace čítající přes 500 stran vytáhl žabího prince, nu, spíš obyčejnou žabu.

\$ texdoc tcolorbox

```

%! lualatex mal-tcolorbox.tex
\documentclass{standalone}
\usepackage{tikz}
\usepackage{tcolorbox}
\usetikzlibrary{patterns}
\tcbuselibrary{skins, hooks}
\tcbset{frogbox/.style={enhanced, colback=green!10, colframe=green!65!black,
  enlarge top by=5.5mm, overlay={\foreach \x in {2cm,3.5cm} {
\begin{scope}[shift={([xshift=\x]frame.north west)}]
\path[draw=green!65!black, fill=green!10, line width=1mm] (0,0) arc
(0:180:5mm);
\path[fill=black] (-0.2,0) arc (0:180:1mm);
\end{scope}}}}}
\tcbset{ribbon/.style={overlay app={\path[fill=blue!75!white, draw=blue,
double=white!85!blue, preaction={opacity=0.6, fill=blue!75!white}, line
width=0.1mm, double distance=0.2mm, pattern=fivepointed stars, pattern
color=white!75!blue] ([xshift=-0.2mm, yshift=-1.02cm]frame.north east)
-- ++(-1,1) -- ++(-0.5,0) -- ++(1.5,-1.5) -- cycle;}}}
\begin{document}
\begin{tcolorbox}[frogbox, ribbon, title=Experiment]
Pozdrav od balíčku \textsf{tcolorbox}\ldots\par\ldots s použitím více vrstev.
\end{tcolorbox}
\end{document}

```



2.9. Balíček **tikz-planets**

S úsměvem píš, že s příchodem Lua (z portugalštiny měsíc) se hodí takový balíček. Zde je ukázka vysázení fázi Měsíce.

\$ texdoc tikz-planets

```

%! lualatex mal-tikz-planets.tex

```

```
\documentclass{standalone}
\usepackage{planets}
\begin{document}
\begin{tikzpicture}[scale=0.7]
\planet[surface=moon, phase=new, centerx=0]
  \node at (0, 0) {new};
\planet[surface=moon, phase=first crescent, centerx=2]
  \node[align=center] at (2, 0) {first \\\ crescent};
\planet[surface=moon, phase=first half, centerx=4]
  \node[align=center] at (4, 0) {first \\\ half};
\planet[surface=moon, phase=waxing gibbous, centerx=6]
  \node[align=center] at (6, 0) {waxing \\\ gibbous};
\planet[surface=moon, phase=full, centerx=8]
  \node[align=center] at (8, 0) {full};
\planet[surface=moon, phase=waning gibbous, centerx=10]
  \node[align=center] at (10, 0) {waning \\\ gibbous};
\planet[surface=moon, phase=last half, centerx=12]
  \node[align=center] at (12, 0) {last \\\ half};
\planet[surface=moon, phase=last crescent, centerx=14]
  \node[align=center] at (14, 0) {last \\\ crescent};
\end{tikzpicture}
\end{document}
```



2.10. Balíček **emoji**

Program **HarfBuzz** ([GitHub](#)) umí vykreslovat písma jako třeba známější program **Pango**. První pokusy o zařazení do $\text{T}_{\text{E}}\text{X}$ u jsem viděl u **Michala Hofticha**, novější je pokus u Lua modulu od **Deepaka Joiseho**. Další testy lze nalézt v článcích v TUGboatu od **Khaleda Hosnyho** ([GitHub](#)) či v MAPS od **Kaie Eignera** ([GitHub](#)). Pro nás smrtelníky se jedná o užití barevných a exotických písem. Do $\text{LuaT}_{\text{E}}\text{X}$ u knihovnu zařadil **Luigi Scarso** a tým $\text{LuaT}_{\text{E}}\text{X}$ u.

V $\text{PlainT}_{\text{E}}\text{X}$ u se užívá `luahtex` a v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u `lualatex-dev`. To bylo nutné ještě v $\text{T}_{\text{E}}\text{X}$ Live 2019. Od $\text{T}_{\text{E}}\text{X}$ Live 2020 stačí opět užívat `lualatex`. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ový formát jsem užíval u všech zmíněných ukázek této zprávy.

Zde vstupuje do popředí balíček **emoji** od **Xiangdong Zeng** (曾祥东). Na některá písma mě navedla dokumentace, některá jsem si stáhl. První a poslední písmo je rastrové, zbytek jsou písma vektorová. Druhé písmo není v barvě. V balíčku je předvolené první písmo. Pokus o rozšíření citací o emoji zkusil **Leon Sixt** v úsměvném balíčku **emojicite**.

```
$ wget -O EmojiOneMozilla.ttf https://github.com/mozilla/positron/blob/master/
  browser/fonts/EmojiOneMozilla.ttf?raw=true
$ wget -O AppleColorEmoji.ttf https://github.com/potyt/fonts/blob/master/
  macfonts/Apple%20Color%20Emoji/Apple%20Color%20Emoji.ttf?raw=true
```

```
$ texdoc emoji emoji cite
```

```
%! lualatex mal-emoji.tex
\documentclass{standalone}
\usepackage{emoji}
\begin{document} \fontsize{19}{19}\selectfont
\def\mallist#1{%
\setemojifont{NotoColorEmoji.ttf}\emoji{#1}% Předvolené
\setemojifont{NotoEmoji-Regular.ttf}\emoji{#1}%
\setemojifont{EmojiOneMozilla.ttf}\emoji{#1}%
\setemojifont{TwemojiMozilla.ttf}\emoji{#1}%
\setemojifont{AppleColorEmoji.ttf}\emoji{#1}}%
\mallist{joy} \mallist{kiss-mark} \mallist{+1}
\end{document}
```



2.11. Balíček **pgfornament**

Na odlehčenou zmíním ještě jeden balíček, který je přepracován přes TikZ a dává tak možnost zasáhnout do různých kreseb a udělat z nich malbu. Autorem je **Alain Matthes**, ornament vlevo na další straně. Velkou inspirací ke vzniku byl balíček **pgfornament-han** z roku 2018 od LianTze Lim (林莲枝) a Chennan Zhang (张晨南), viz ornament vpravo.

```
$ texdoc pgfornament pgfornament-han
```

```
%! lualatex mal-pgfornament.tex
\documentclass[margin=1pt]{standalone}
\usepackage[dvipsnames]{xcolor}
\usepackage{pgfornament}
\usepackage{pgfornament-han}
\begin{document}
\tikzset{pgfornamentstyle/.style={draw=green!20!black, fill=orange, fill
  opacity=.5, scale=0.7, ultra thick}}%
\tikz\node{\pgfornament{3}};
\tikzset{pgfornamentstyle/.style={draw=Goldenrod, fill=Red, line width=1pt}}
\tikz\node[fill=black, circle, draw=Red, line width=2pt, inner
  sep=-8pt]{\pgfornamenthan[scale=0.25]{56}};
\end{document}
```



2.12. Za pozornost ještě stojí

Již bez ukázek upozorňuji na další nástroje a balíčky.

- **xindex** od **Herberta Voße** je na Lua_{TeX}u založený rejstříkový procesor. Je to aktivní vývojář, hlavně kolem projektu **PSTricks** a autor mnoha knih a dokumentace balíčků.
- **tex4ebook** je na Lua_{TeX}u založený balíček na převod z L^AT_EXu do elektronické knihy od českého vývojáře **Michala Hofticha**.
- **lwarp** je podobně smýšlející projekt na převod z L^AT_EXu do HTML5 od **Briana Dunny**.
- Nelze zapomenout na neustále vylepšovaný obří nástroj na přípravu seznamu literatury **biblatex** s jeho **balíčky**.
- V neposlední řadě balíček **ocgx2**, který je nástupcem balíčků **ocgx** a **ocg-p** od **Alexandra Grahna**, mj. autora balíčků **media9**, **animate** a nového experimentálního balíčku **media4svg**.
- O nástroji **dvisvgm**, který se užívá v pozadí balíčku **media4svg** či nástroje **Manim** na matematické animace, ještě uslyšíme, protože plánují vedle převodu z dvi do svg i převod pdf do svg.

3. METAPOST ztracen a nalezen

METAPOST nahradil METAFONT na kresbu. Pamatuji si své začátky nad příklady **Vincenta Zoonekynda** ([archiv](#)). Dnes je METAPOST integrován do Con_TE_Xtu přímo jako knihovna, zájemce odkazují na **Con_TE_Xt Garden**.

Jaromír Antoch se mne ptal, jestli by dokázal dostat vektorovou podobu svých kreseb na webové stránky. Když opomineme rastr, formát pdf samotný či konverzi do jiných formátů, tak stojí za pokus to vyzkoušet přímo v METAPOSTu. V minulých letech se totiž do zásahů pustil Taco Hoekwater, jeden z jeho nápadů byl rozšířit výstup do svg.

Vzal jsem si do parády ukázkou č. 32 od **Vincenta Zoonekynda**, upravil jsem ji dle návodu v dokumentaci **texdoc metapost**, str. 5, do následující podoby. Jen jsem v proměnné **outputtemplate** místo **mps** užil **svg**:

```

%! mpost zoonek.mp
prologues := 3;
outputtemplate := "%j-%c.svg";
outputformat := "svg";
beginfig(32)
  u:=1cm; pair A,B,C,D,E,F,G;
  A := (-u,u); B := (0,u); C := (u,u); D := (-u,0); E := (0,0); F := (u,0);
  draw A--D; draw A--E; draw A--F;
  draw B--D; draw B--E; draw B--F;
  draw C--D; draw C--E; draw C--F;
  dotlabel.top(btex $a$ etex, A); dotlabel.top(btex $b$ etex, B);
  dotlabel.top(btex $c$ etex, C); dotlabel.bot(btex $a'$ etex, D);
  dotlabel.bot(btex $b'$ etex, E); dotlabel.bot(btex $c'$ etex, F);
endfig;
bye.

```

Spustil jsem poté:

```

$ mpost zoonek.mp
$ inkscape zoonek-32.svg &
$ firefox zoonek-32.svg &

```

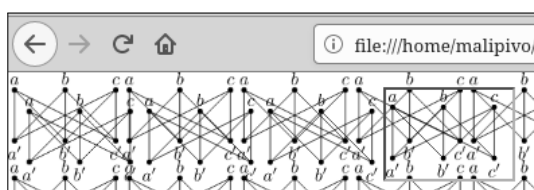
První řádek vygeneruje soubor `zoonek-32.svg`, druhý řádek soubor otevře pro případnou úpravu a poslední řádek otevře soubor přímo v prohlížeči.

Za pomoci webové [nápovědy](#) jsem zkusil vložit obrázek do webové stránky `index.htm` a tu si pak přes `firefox index.htm` otevřít. Jedná se o čtyři základní způsoby vložení svg plus pátou cestu přes kaskádový styl CSS jako opakující se obrázek v pozadí. Snad se v náhledu zorientujete. Určitě existuje nespočet dalších způsobů, nechávám hlubší bádání na čtenáři.

```

<!DOCTYPE html>
<html>
<head><meta charset="UTF-8"/>
  <style>body{background-image: url(zoonek-32.svg);}</style>
</head>
<body>
  
  <object type="image/svg+xml" data="zoonek-32.svg"></object>
  <embed type="image/svg+xml" src="zoonek-32.svg" />
  <iframe src="zoonek-32.svg" width="70px" height="50px"></iframe>
</body>
</html>

```



„Mně se to líbilo a potvrdilo mi to, že stojí, jde-li to, dělat věci nad základem, který bývá stálý, zatímco balíčky vymírají se svými tvůrci...“

Jaromír Antoch

Nesmrtelná slova. Vykreslit je do kamene!

4. Co dodat závěrem?

Tohle vše máme opět k dispozici zadarmo, se zdrojovými kódy na přípravu čehokoliv a na dosah klávesnice.

Jo, abych nezapomněl: **Donald E. Knuth** alias DEK alias 高德纳 byl v září 2019 v Brně na **Fakultě informatiky Masarykovy univerzity** u příležitosti 25. výročí založení fakulty. Je tam přednáška, fotky ad.

www.fi.muni.cz/events/2019-celebrations-of-25-years-of-fi.html

V tu dobu jsem ve stavech zoufalství sázel cosi v jakémsi $\text{T}_{\text{E}}\text{X}$ u, tak jsem přednášku, diskuzi a varhanní koncert vynechal. Možná by mi „Grand Wizard“ poradil. Kdoví!



Zleva: Jan Šustek, Jiří Rybička, DEK, Petr Sojka a Tomáš Hála

SBOHEM, DRAHÝ KARLE, SBOHEM! ADIEU, MONSIEUR KAREL, ADIEU!

Pavel Stríž

E-mail: pavel@striz.cz

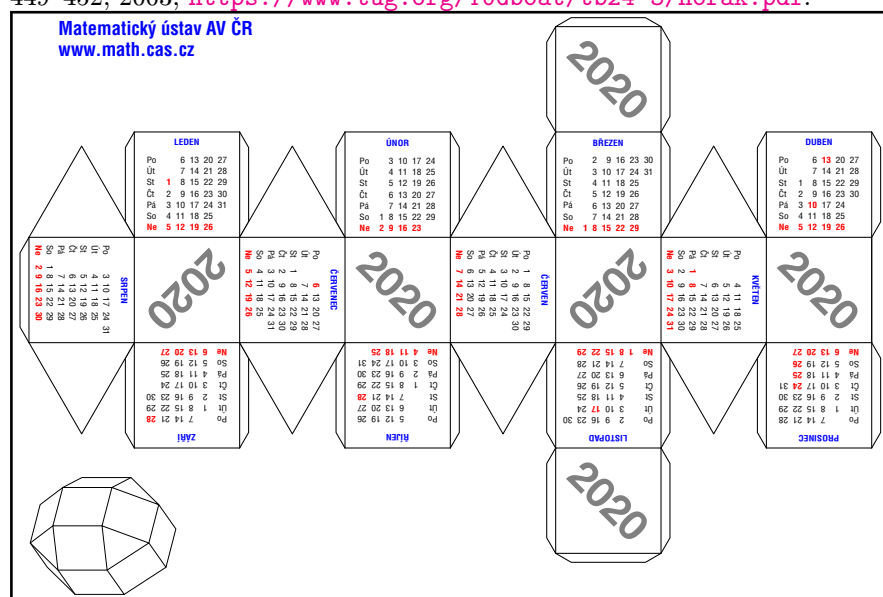
Karel Horák se s námi rozloučil 22. srpna 2020 ve svých ranných 66 letech.

Setkávání se s panem doktorem

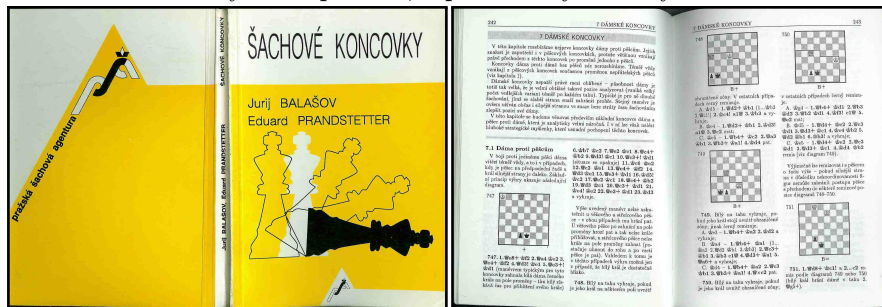
Pravděpodobně každý z vás má v okolí osobu, u které si nevzpomene, kdy a kde se s ní poprvé setkal, ať už osobně či zprostředkovaně.

To je můj případ u Karla Horáka, mj. spolužáka Jaromíra Antocha. Ať marně přemýšlím, nevím, jestli jsem jeho jméno zahlédl poprvé u některé z tirází sbírek úloh, nebo jako jeden z úspěšných soutěžících či členů komise u matematických olympiád, nebo jako jeden z předsedů Čs. sdružení uživatelů $\text{T}_\text{E}_\text{X}$, či jako jeden z účastníků Bacho $\text{T}_\text{E}_\text{X}$ ů, Euro $\text{T}_\text{E}_\text{X}$ ů či Con $\text{T}_\text{E}_\text{X}$ t mítinků.

Řada $\text{T}_\text{E}_\text{X}$ istů se s jeho prací poprvé setkala přes jeho kalendáře. Zde je jedna ukáзка. Je to dělané na koleně přes METAPOST. Začal s nimi na přelomu tisíciletí (tipuji od padesátin) a skončil u 17 stran. Lze změnit rok a vše se přepočítá a překreslí. Napsal o nich příspěvek *Geometric Diversions with $\text{T}_\text{E}_\text{X}$, METAFONT and METAPOST* do časopisu TUGBoat, Vol. 24, No. 3, 449–452, 2003, <https://www.tug.org/TUGboat/tb24-3/horak.pdf>.



Podobně výrazně se zapsal do sazby šachu, kde vytvořil potřebná makra pro Jaroslava Poláška a bylo možné se proklikávat z PostScriptu do zdrojového souboru, což nyní umí u PDF například $\text{T}_{\text{E}}\text{X}$ works. Ukázku zmínil v článku *Do šachu s $\text{T}_{\text{E}}\text{X}$ em!* ve Zpravodaji Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u, Vol. 1, No. 3, 25–26, 1991, <https://www.cstug.cz/bulletin/pdf/bul913.pdf>. Přikládám obálku a ukázku sazby z knihy Jurij Balašov, Eduard Prandstetter: *Šachové koncovky*, Pražská šachová agentura, 1991, neb Karel tvořil i obálky vedle plakátů, diplomů a výsledkových listin.



S jistotou však mohu říci, že Karel nevynechal jedinou konferenci $\text{T}_{\text{E}}\text{X}$ perience, kterou jsem organizoval za mého působení ve Zlíně. Při každém setkání se mne ptal, jestli se neplánuje další ročník. Podobně mě Karel inspiroval, jestli jsem už nepokročil se sazbou knihy českých erbů měst, o které jsem mu vyprávěl, jak ji plánuji vysazet a jaká data už mám. Těšil se na ni hodně.

Díky mé zálibě v kresbě a překreslování grafů a obrázků mi Karel nabídl práci na Matematickém ústavě AV ČR na pozici technického sazeče. Jako jeho učeň jsem nahlédl víc do Plain $\text{T}_{\text{E}}\text{X}$ u, byť jsem si u úprav šablon spíše rval vlasy. Karel obdivoval práci Donalda E. Knutha, Petra Olšáka, Zdeňka Wagnera, Karla Písky, Jana Kuly, Honzy Šustka, holandských a polských $\text{T}_{\text{E}}\text{X}$ istů. Jeho záběru u PostScriptu se málokdo vyrovnal. To bylo Karlovo! Jednou mi hrdě prozradil, že přímo ve velké tiskárně sedl ke stroji a v textovém editoru zasáhl do tiskového podkladu, aby se mohlo začít vyrábět. Jednalo se o změnu barevného prostoru v postscriptovém souboru.

A znovu jsem díky Karlovi nahlédl víc do METAPOSTu. Karel byl vždy trochu smutný, když jsem ho přemlouval k $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u či TikZu. Uznal, že pro méně zkušené uživatele $\text{T}_{\text{E}}\text{X}$ u je to lepší, ale že je potřeba stejně jít do jádra. Co mě však potěšilo, že z učně jsem se stal mistrem pro učně Karla u Lua($\text{T}_{\text{E}}\text{X}$ u). Osobně i emailově jsme řešili různé vychytávky a úpravy textů přes Lua. Ať už se jednalo o úpravy malých či zásahy do velkých souborů.

Jednu z krásných vzpomínek mám, když jsme Karla ztratili během výletu na $\text{T}_{\text{E}}\text{X}$ perience. Karla nikde a přitom všichni ostatní už byli z výletu dávno

zpátky. Z ničeho nic mi táta podává mobil, že mám hovor. A hle, Karel, že si odbočil na prohlídku zříceniny, že je tma, a že neví, jak se dostat zpět na rekreační středisko Jestřabí na Rusavě.

— Pane Horáku, vidíte někde nějaká světla?

— Vidím.

— Tak jděte za tím světlem, buď je to naše chata či začátek vesnice a z tma my vás už vyzvedneme autem. Volejte.

— Dobře, já to tedy zkusím.

Mám ještě jednu vzpomínku, se kterou se rád podělím. Karel byl kavalír, džentlmen a věděl vždy, kdy už stačí. Na ConT_EXt mítinku na Mlýně Brejlov u Týnce nad Sázavou jsem Karlovi nabízel ořechovku od táty.

— Mám tu vzorek z roku 2008. Dáte si?

— Ochutnám.

— Mám tu ještě vzorek z roku 2007.

— Neměl bych, ale to ještě ochutnám.

— Nedáte si ještě? Mám tu ještě vzorky z let 2005 a 2006. Táta každý rok trochu experimentoval.

— Ne, ne, opravdu děkuji, ale mně už stačí. Mám akorát. Ale vyřídíte tatínkovi, že je výborná, že se mu podařila.

Trochu mě mrzí, že Karel nedorazil ani jednou do Žiliny na konferenci OSSConf, kde vznikla za víc jak desetiletí zajímavá T_EXová sekce s příjemným posluchačstvem. Pro mne to bylo volné pokračování české konference T_EXperience na Slovensku. Ovšem s tou výhodou, že člověk může nahlédnout do jiných sekcí: vývoj opensource software, OSS ve vzdělávání, opendatové, openhardwarové, open GISové, ale i do 3D tisku. T_EXperience byla pro kolegy ze Slovenska vždy docela z ruky. Říkával, že se obává, že by lidé nerozuměli jemu a on jim. Byl by příjemně překvapený.

Předposlední email od Karla mi dokázal, že je to bojovník. Podrobně mi popsal co a kde mu z těla doktoři vedou, a že je upoután na lůžku doma a čeká mezi chemoterapiemi. Psal, že má prostor díky koronární ráně dokončit své projekty, bez ohledu na to, jak léčba zhoubného nádoru na slinivce, který zablokoval žlučovyody, dopadne. Psal jsem mu, že držím palce a můj poslední email byl náhled na animaci pro Zdeňka Wagnera o matematické gnostice. To by pro METAPOST byla trochu síla. Přikládám pár vzorků z animace.



Odepsal mi: „Pavliku, Tys hracicka :-) Zdravim, Karel“. Tak se se mnou virtuálně rozloučil, to byl od něj poslední email, náhodou vyšel na den mých narozenin.

Karel mi bude chybět, s ním odchází velký kus $\text{T}_{\text{E}}\text{X}$ ového umění. Narozdíl od Knutha, který si od komunity drží odstup, Karel vždy poradil a pomáhal. Rozloučím se s ním náhledy z jeho tvorby a překreslování, které jsem rozdělil do několika bloků dle místa jeho působení. Je to střípek z jeho obří práce da Vinciho záběru.

Vždy si rád vzpomenu na pana doktora z poslední $\text{T}_{\text{E}}\text{X}$ perience, jak nás s křídou v ruce u černé školní tabule, o jejíž instalaci nás poprosil $\text{T}_{\text{E}}\text{X}$ pert Petr Olšák, léčí ze záludností $\text{T}_{\text{E}}\text{X}$ ového světa.

Nakladatelství Prometheus  a **JČMF** ¹

<http://prometheus-nakl.cz/>

<https://www.jcmf.cz/>

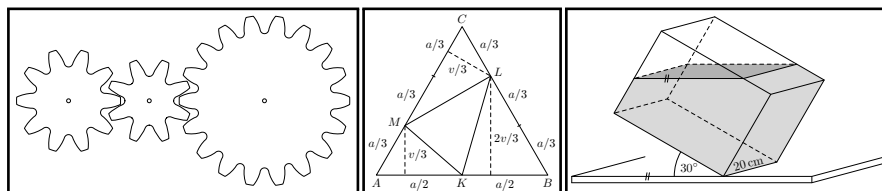
Karel sázel nejrozumnější knihy, učebnice a sbírky úloh. Snažil se vše sázet jen v Plain $\text{T}_{\text{E}}\text{X}$ u a ve svých makrech. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ neměl rád a do $\text{C}_{\text{O}}\text{N}_{\text{T}}\text{E}_{\text{X}}\text{T}$ u se nestihl zamilovat. Na obrázky používal $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{P}_{\text{O}}\text{S} \text{ T}$ s řadou udělatek, pomůcek a konverzních skriptů.

Mne zaujal **odstavce** z roku 2006 v *Rozhledech matematicko-fyzikálních*, Vol. 81, No. 2, 44–47. Jedná se o poměrně zajímavý obrázek na nakreslení umístěný v pozadí odstavce. Zároveň je to náhled na styl jeho psaní.

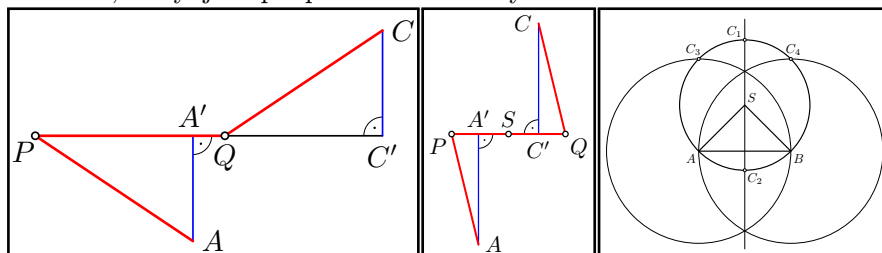
Přípravu a zdárný průběh celé akce zajišťovali organizátoři z řad členů *Mexické matematické společnosti* za podpory mexického ministerstva školství, vlády státu Yucatán, tamních univerzit a desítek sponzorů. Na-shromážděné finanční prostředky umožnily ubytovat všechny soutěžící, vedoucí družstev i členy výborů a hodnotících komisí v areálu luxusních hotelů nedaleko centra yucatánské metropole, založené španělskými dobyvateli roku 1542 na místě mayského města *Tihó*. Mexičtí hostitelé připravili výborné podmínky pro vlastní soutěž i zajímavý doprovodný program, jehož vrcholem byl celodenní výlet ke zříceninám mayského města *Chichén Itzá*. Závěr olympiády mírně narušil příchod hurikánu *Emily*, který však nakonec Méridu minul zhruba o 80 km a v samotném městě se projevil jen silnějším větrem.

¹Za pomoc jmenovitě děkuji Aleně Šolcové.

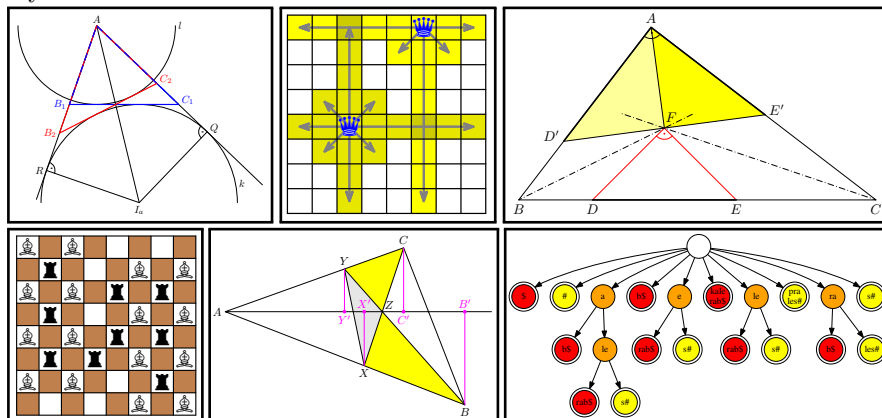
47



Vždy se pečlivě snažil průniky křivek a důležité body zvýraznit prázdným kolečkem, to byl jeho podpis. Značka kvality.



Užití barev se neštítel. Zde je několik dalších obrázků v barvě z těch samých ročníků.



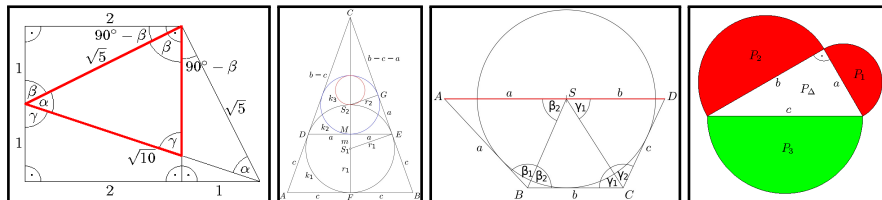
MÚ AV ČR 

<http://www.math.cas.cz/>

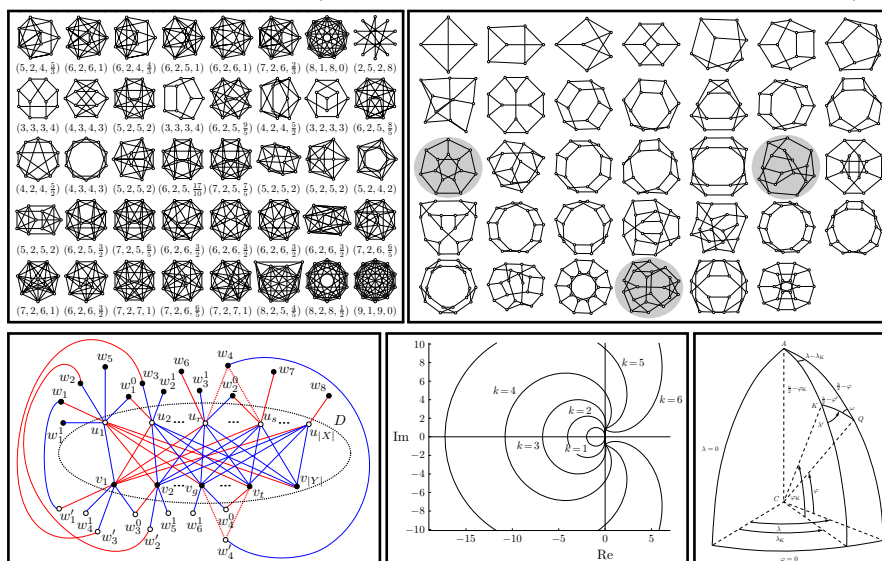
Karlova domovina byl Matematický ústav AV ČR. I zde šířil povědomí o $\text{T}_{\text{E}}\text{X}$ u, $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{P}_{\text{O}}\text{S}_{\text{T}}$ u, a jak mi potvrdili jeho kolegové, vždy každému rád poradil.

³Za pomoc jmenovitě děkuji Zdence Crkalové, Martinu Jarníkovi, Jiřímu Rákosníkovi a Tomáši Vejchodskému.

Z první ruky mohou potvrdit, že častokrát i ve dvě ráno. Zde je několik obrázků ze současného projektu *Matematika pro poučení i pro zábavu*, ze sekce Zábavných úloh, <http://matikadomu.math.cas.cz/>.

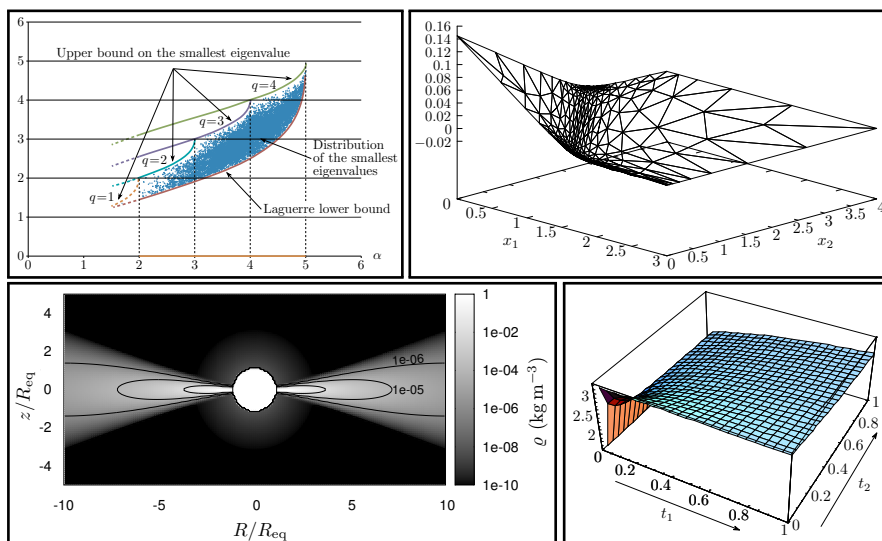


Nejtěžší sazební úkoly vyplynuly z časopisů ústavu: Czechoslovak Mathematical Journal (dále CMJ, založen 1951, <http://cmj.math.cas.cz/>), Applications of Mathematics (AM, založen 1956, <http://am.math.cas.cz/>) a Mathematica Bohemica (MB, založen 1872, <http://mb.math.cas.cz/>).



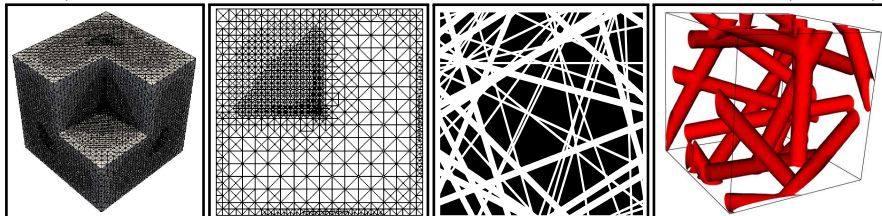
Z článků J. C. Hurajová, T. Madaras: *More on betweenness-uniform graphs*, CMJ, 68(2), 293–306, 2018 (dva obrázky); J. Yun Yue, S. Meiqin Wei, T. Yan Zhao: *Proper connection number of bipartite graphs*, CMJ, 68(2), 307–322, 2018; M. Vlasák: *Time discretizations for evolution problems*, AM, 62(2), 135–169, 2017 a T. Bayer, M. Kočandrlová: *Reconstruction of map projection, its inverse and re-projection*, AM, 63(4), 455–481, 2018.

U některých grafů nerad uznal, že jsou příliš těžké na překreslení bez vstupních dat, tak se snažil o sazební jednotu popisků v grafu či alespoň o sazbu popisků os.



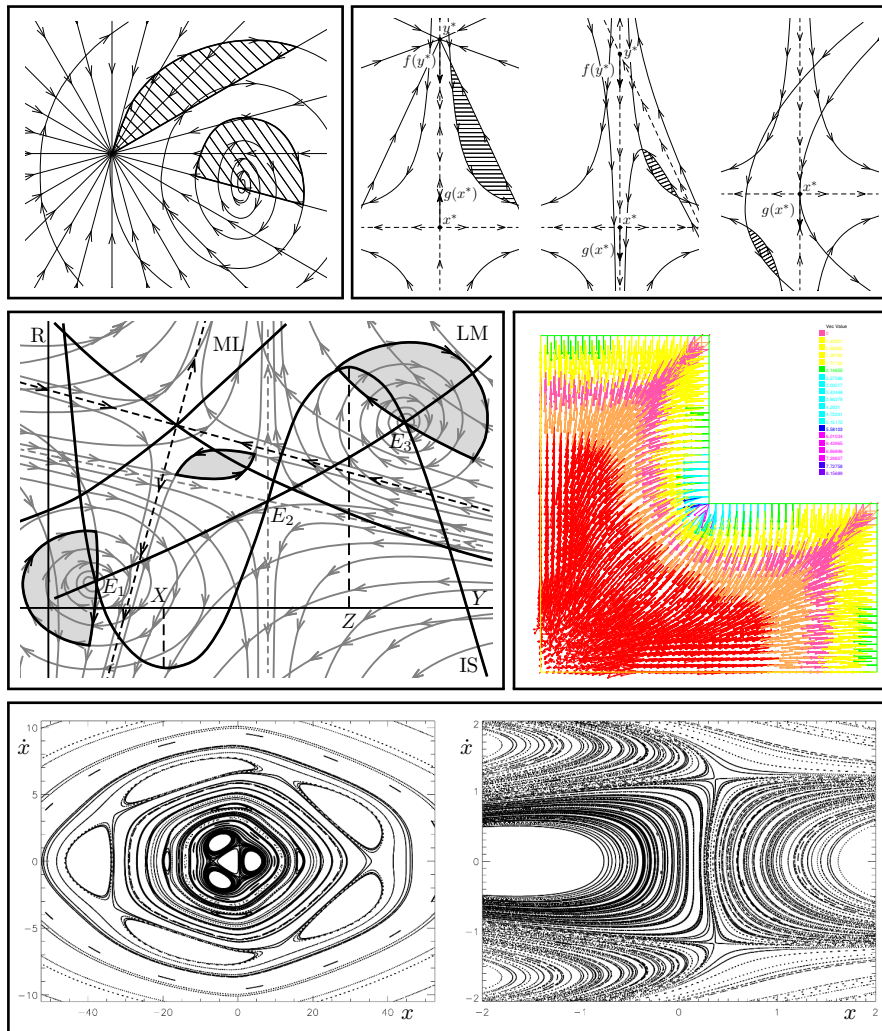
Z článků Y. Yamamoto: *On the optimality and sharpness of Laguerre's lower bound on the smallest eigenvalue of a symmetric positive definite matrix*, AM, 62(4), 319–331, 2017; J. Hozman, T. Tichý: *DG method for numerical pricing of multi-asset Asian options—the case of options with floating strike*, AM, 62(2), 171–195, 2017; P. Kurfürst, J. Krtička: *Time-dependent numerical modeling of large-scale astrophysical processes: from relatively smooth flows to explosive events with extremely large discontinuities and high Mach numbers*, AM, 62(6), 633–659, 2017 a A. Ghost, C. Kundu: *On generalized conditional cumulative past inaccuracy measure*, AM, 63(2), 167–193, 2018.

Občas se objevily podklady, kde nebylo možné či praktické zasáhnout, tak říkal, že to se nedá nic dělat. Zmínil, že autory oslovoval o lepší verzi, ale kde to šlo, autory dál nezatěžoval. Zde je ukázka ponechaného rastrového obrázku, vektorový obrázek o mnoha linkách by byl neúnosně velký (obr. vlevo). Podobně to platívá u fotek, skenů a modelů se světly a stíny (vpravo).



Po dvou obrázcích z článků Fei Xu, Hehu Xie: *A full multigrid method for semilinear elliptic equation*, AM, 62(3), 225–241, 2017 a D. Jeulin: *Iterated Boolean random varieties and application to fracture statistics models*, AM, 61(4), 363–386, 2016.

Podklady nám chodily naskenované v ruce, přes kresby v $\text{\TeX}$ u, až po nejruznější výstupy z výpočetních a simulačních nástrojů. Jednalo se o celou širší oboru. Zde je několik extra „vypečených“ ukázek pro potěšení oka čtenáře.

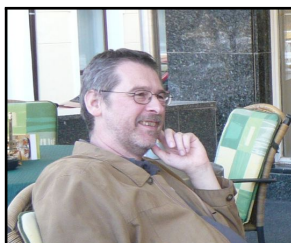


Z článků B. Volná: *Chaotic behaviour of continuous dynamical system generated by Euler equation branching and its application in macroeconomic equilibrium model*, MB, 140(4), 437–445, 2015 (tři obrázky); X. Hu, P. Huang, X. Feng: *A new mixed finite element method based on the Crank-Nicolson scheme for Burgers' equation*, AM, 61(1), 27–45,

2016 a J. Málek, K. R. Rajagopal, P. Suková: *Response of a class of mechanical oscillators described by a novel system of differential-algebraic equations*, AM, 61(1), 79–102, 2016.

Byl jednou jeden pan doktor

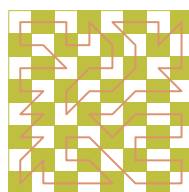
Během naší komunikace jsme řešili nejrůznější zajímavosti, a když Karel něco nevěděl, to už bylo co říct. Jeden z problémů, který mi zmínil, bylo přesázení **zdrojových kódů** zaniklého ruského časopisu Kvanta, http://www.kvant.info/zkm_main.htm. Byl tam problém s chybějícími písmi, zjištěním kódové stránky i jak získat vektorovou podobu z rastrových obrázků a Flashových animací. Shrnul jsem své postřehy na stránce <https://tex.stackexchange.com/questions/181153>. Jeho idea byla si před tiskem ročníky 1970 až 2009 přesázet do něčeho útulnějšího, následně vytisknout a vyvázat. Karel dobře věděl, kde hledat inspiraci a jak inspirovat ostatní.



Nyní se píše rok 2020 a PDF získaná $\text{\TeX}$ em stále neobsahují původní zdrojové kódy. Tohoto zlepšováku se Karel nedočkal. Rozloučím se s vámi, drazí pozůstalí, ukázkou z tohoto **časopisu** ze stran 31 a 143.

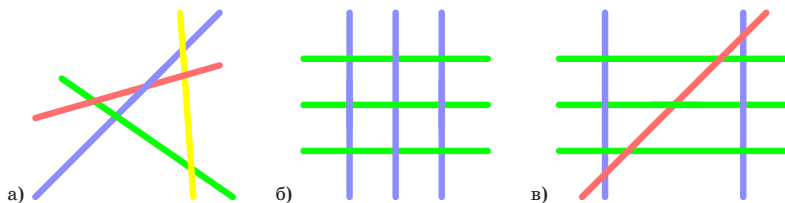
Chybí nám všem. Je to rána! Byl to borec.

220. Король обошёл шахматную доску, побывав на каждом поле ровно один раз и вернувшись последним ходом на исходное поле. (Король ходит по обычным правилам: за один ход он может перейти по горизонтали, вертикали или диагонали на любое соседнее поле.) Когда нарисовали его путь, последовательно соединив центры полей, которые он проходил, получилась замкнутая ломаная без самопересечений. Какую наименьшую и какую наибольшую длину может она иметь? (Сторона клетки равна единице.)



А.В.Климов. VII Всесоюзная олимпиада. Решение — в №5–1974. Статья И.Ф.Акулича «Проекции короля» третьего номера 2000 года. Статья Н.Б.Васильева «Вокруг формулы Пика» двенадцатого номера 1974 года

1085*: Несколько попарно скрещивающихся прямых, расположенных в пространстве, спроецировали на горизонтальную плоскость. Их проекции изображены так, чтобы в точках пересечения было видно, какая точка расположена выше, а какая ниже:



Могла ли получиться проекция, изображённая на рисунке? *С.Л.Табачников. Решение — в №5–1988. Комментарий — в статье О.Я.Виро и Ю.В.Дроботузиной «Сцепления скрещивающихся прямых» третьего номера 1988 года, в статье С.Л.Табачникова «Линейные неравенства и задачи M1085» шестого номера 1989 года и в статье А. Скопенкова «Ещё раз о скрещивающихся прямых» одиннадцатого номера 1989 года*

**VYHLÁŠENÍ SOUTĚŽE:
DATA ANALYSIS COMPETITION 2021
DATA ANALYSIS COMPETITION 2021**

Zdeněk Hlávka

E-mail: hlavka@karlin.mff.cuni.cz

CALL OF THE DATA ANALYSIS COMPETITION 2021

The International Association for Statistical Computing (IASC) announces the **Data Analysis Competition 2021**.

IASC is an Association of the International Statistical Institute (ISI) whose objectives are to promote the theory, methods, and practice of statistical computing and to foster interest and knowledge in effective and efficient statistical computing through international contacts among researchers and professionals in statistics, computer science, and related areas at universities, organizations, institutions, governments, and the general public in different countries of the world, to convert data into information and knowledge.

For the 2021 Data Competition, winners will be invited to present their work at the **Data Science, Statistics, and Visualisation Conference (DSSV-2021)** to be held at **Erasmus University, Rotterdam (July 7-9, 2021)** or at some other event organized by IASC (see <https://iasc-isi.org/events-all/>), depending on the current COVID-19 situation. The winners will also be invited to submit a manuscript for possible publication (following peer review) to IASC's **Journal of Data Science, Statistics, and Visualisation** (see <https://jdssv.org>). In addition, IASC will sponsor participation at the virtual **63rd ISI World Statistics Congress (July 12-16, 2021)** and **one year IASC membership** to all authors of submissions selected by the Committee on Data Analysis Competition; see <https://iasc-isi.org/become-a-member/> for an overview of IASC membership benefits.

The theme of the 2021 competition is around the **analysis of quality-of-life related data** and the submission should clearly describe the significance of your findings either for individuals or for the society. The primary data set may come from one or more databases but connecting information from different databases may help to obtain interesting and original conclusions.

There are currently many sources related to the spread of COVID-19, allowing to investigate topics like social inequalities in post-COVID-19 world or to compare effects of governments' actions, but we encourage you to investigate also any other quality-of-life related issues (health, education, environment, social inequalities, etc.) In your analysis, you may concentrate on individuals, a single region (e.g., your own country), a continent or even the entire world.

Your entry must be submitted as a poster in PDF format and you must clearly specify the source of your data, i.e., by listing the relevant URLs and the steps required to obtain the data.

The competition is open to everyone who is interested in presenting their poster at **WSC 2021**. You are allowed to work individually or in a small group of up to five participants on your poster. Posters will be judged according to these criteria:

1. Appropriateness of analysis.
2. Novelty of approaches used in the analysis.
3. Clarity of objectives, approaches, data management, displays, and results.
4. Significance of findings.
5. Generalizability of approaches to data sets in other areas.
6. Overall quality of poster.

Not all posters are expected to meet all criteria to the same degree. Your poster may contain links to databases, computer programs, animations, and other external sources and it may (but need not) be accompanied by a short description (maximum five pages). All materials must be submitted in PDF format.

Updates and results of previous editions of the Data Analysis Competition may be found at <https://iasc-isi.org/data-analysis-competition/>

DEADLINE FOR SUBMISSION:

Final submissions (PDF) are due on **30-th April 2021**.
Submit to: dac.iasc@email.cz

FURTHER INFORMATION:

For all inquiries contact:
Associate Professor Zdeněk Hlávka: dac.iasc@email.cz
