ZPRAVODAJ

Československého sdružení uživatelů T_EXu[illegible]

1-2
2008

OBSAH

Jaromír Kuben: Úvodník	1
Zdeněk Wagner: Errata	2
Karel Píška: Testování LM-fontů s ohledem na čs. sazbu	3
Luboš Prchal a Pavel Schlesinger: Poster v T _E Xu	44
Luboš Prchal a Pavel Schlesinger: Prezentace v T _E Xu	56
Pavel Stříž a Michal Polášek: Ukázky prezentací	63
Roman Plch, Petra Šarmanová: Interaktivní 3D grafika v HTML a PDF dokumentech	76
Zdeněk Hlávka: Velkovýroba tabulek pomocí AWK	93
První oznámení a informace: Konference: T _E Xperience 2008	96

Zpravodaj Československého sdružení uživatelů T_EXu je vydáván v tištěné podobě a distribuován zdarma členům sdružení. Po uplynutí dvanácti měsíců od tištěného vydání je poskytován v elektronické podobě (PDF) ve veřejně přístupném archivu dostupném přes <http://www.cstug.cz/>.

Své příspěvky do Zpravodaje můžete zasílat v elektronické podobě, nejlépe jako jeden archivní soubor (.zip, .arj, .tar.gz). Postupujte podle instrukcí, které najdete na stránce <http://bulletin.cstug.cz/>. Pokud nemáte přístup na Internet, můžete zaslat příspěvek na disketě na adresu:

Zdeněk Wagner
Vinohradská 114
130 00 Praha 3

Disketu formátujte nejlépe pro DOS, formáty Macintosh 1.44 MB a EXT2 jsou též přijatelné. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí L^AT_EXu), zejména v případech, kdy vás nelze kontaktovat e-mailem.

ISSN 1211-6661 (tištěná verze)

ISSN 1213-8185 (online verze)

Milí přátelé,

pokud čtete tento úvodník, zřejmě k vám dorazilo první číslo letošního Zpravodaje ζ TUGu. A protože se podařilo sehnat dostatek příspěvků, je to hned dvojčíslo. Věřím, že se vám bude jeho obsah líbit a že si každý z vás najde článek, který ho zaujme. Velkou zásluhu na jeho přípravě má nový člen výboru ζ TUGu, Pavel Stříž, kterému bych chtěl za jeho úsilí poděkovat.

Jedním z příspěvků je rozsáhlý článek Karla Pišky o testování fontů, které prováděl v rámci jednoho ze tří grantů, které ζ TUG vypsal na rok 2006. Jsem velmi rád, že se tak bude moci uzavřít i tento poslední grant a vyplatit odměna, protože vykonaná práce byla nemalá a mimo jiné vysoce oceněná tvůrci Latin Modern fontů. Především však přispěla ke zkvalitnění těchto fontů pro potřeby českých a slovenských uživatelů.

ζ TUG by do budoucna rád vypsal další granty, především takové, které by přinesly užitek naší $\text{T}_{\text{E}}\text{X}$ ovské komunitě. Víťame proto náměty a zejména tipy na řešitele. V minulosti se totiž ukázalo, že pokud nebyl zajištěn řešitel grantu, bylo jeho vypsání bezpředmětné. Jedním z témat by mohlo být pokračování testování technické kvality a posuzování typografické kvality fontů Gyre (včetně matematických fontů), které vyvíjí skupina členů polského GUSTu a na jejichž vývoj naše sdružení přispívá. Bude záležet na tom, zda dosavadní řešitelé (Karel Horák a Karel Piška) budou ochotni pokračovat v této nevědné a časově velmi náročné práci.

V minulém úvodníku jsem se zmiňoval o problémech týkajících se administrativního a technického zázemí ζ TUGu. Jisté kroky v tomto směru byly učiněny, ale ke splnění našich představ je bohužel dost daleko, vše vázne na nedostatku času dobrovolníků, kteří slíbili, že se této činnosti budou věnovat. Nezbyvá než věřit, že vše dospěje ke zdárnému konci.

Po delší odmlce se zdá, že se podaří opět zorganizovat **setkání uživatelů $\text{T}_{\text{E}}\text{X}$** . Věnujte prosím pozornost informaci, kterou najdete v tomto čísle Zpravodaje. A především se sami zúčastněte a informujte a lákejte další potenciální zájemce. Je zajištěno několik kvalitních zvaných přednášek a navíc půjde o příležitost vyměnit si zkušenosti a poznatky a dozvědět se něco nového v souvislosti s tím, co vás při používání $\text{T}_{\text{E}}\text{X}$ u a příbuzných programů trápí. Příprava takové akce není jednoduchá, organizátoři stojí vždy spoustu času a úsilí a je pak smutné, když zájem je mizivý. Přitom členů i nečlenů, kteří po nějaké takové akci volají a vyčítají ζ TUGu a výboru, že se nic neorganizuje, nebývá obvykle málo.

Ale těžko se hledají dobrovolníci, kteří jsou ochotni aktivně přispět k přípravě. Akce je tentokrát oznámena s dostatečným předstihem, takže věříme, že bude úspěšná.

Další číslo Zpravodaje, které dostanou již pouze členové mající zaplacené příspěvky za rok 2008 (termín je do 30. června), by mělo obsahovat DVD s letošní distribucí T_EXLive (pokud do té doby vyjde). V případě nedostatku kvalitních příspěvků počítáme, jak již bylo avizováno na loňské valné hromadě, s překlady kvalitních zahraničních příspěvků, popř. i výjimečně s anglickými příspěvky.

Těším se s vámi na setkání na Rusavě. Zdraví

Jaromír Kuben

Errata

ZDENĚK WAGNER

Ve Zpravodaji 1/2007 se vyskytlo několik chyb. Na konci 1. odstavce strany 64 byly přehozeny číslice v unicodové sekvenci. Správně mělo být U+0915 U+094D U+200D U+0924. Na stranách 63–64 je několikrát chybně uvedeno Chez Ami místo správného Chez Ali. Vegetace v Maroku by měla být označena spíše jako subtropická než tropická.

Ve Zpravodaji 2/2003 má být v tabulce na straně 63 kurzívou vytištěna libra, nikoliv dolar. Na straně 64 místo nesprávných názvů `cminch10` a `csinch10` má být uvedeno `cminch` a `csinch`. Na straně 65 je uvedeno, že dvojice „P.“ a „P.“ nemají ve fontu kerning. Ve skutečnosti kerning mají, a to -0.008334 pt. Na straně 84 je chybně vytištěn znak s ogonkem, správně má být `ą`.

Redakce se za chyby omlouvá a děkuje Karlu Píškovi, který chyby nalezl a ohlásil.

Zdeněk Wagner
bulletin@cstug.cz

$\text{\texttt{\textit{C}}TeX}$ a $\text{\texttt{\textit{L}}\textit{A}TeX}$ dostaly novou šanci. $\text{\texttt{\textit{C}}TUG}$ dosud neměl k dispozici pokračovatele CM-fontů ve formátech Type 1 a OpenType. Prvním takovým kandidátem, v dostatečně úplném, kvalitním a optimalizovaném provedení, jsou fonty Latin Modern. $\text{\texttt{\textit{C}}TUG}$ se proto připojil ke sponzorům jejich vývoje a následně podpořil formou grantu také testování této kolekce fontů z hlediska použití pro českou a slovenskou sazbu. Shrnutí těchto aktivit se věnuje tento dokument.

1. Obsah práce a článku

Cílem práce bylo průběžně kontrolovat, zda LM-fonty neobsahují chyby, tím napomoci jejich opravě v další verzi; soustředit pozornost na správnost glyfů a ligatur, na vybraná metrická data, zejména šířky znaků, vhodnost či nevhodnost kerningových párů; detekovat a posoudit rozdíly v metrikách LM, $\text{\texttt{\textit{C}}S}$, CM-fontů nebo také EC-fontů; ověřit technickou kvalitu fontů.

Náplní vlastního testování LM-fontů bylo generování testovacích tisků glyfů, samostatná autonomní vizuální kontrola všech glyfů, v některých případech jejich porovnání s $\text{\texttt{\textit{C}}S}$ nebo jinými fonty; generování vybraných metrických údajů v tabulkové a poté i grafické formě, srovnávání jejich hodnot mezi LM, $\text{\texttt{\textit{C}}S}$, CM a případně EC-fonty. Sledovaly se rozdíly v zobrazení znaků, obzvláště tvar a umístění akcentů, v metrikách také rozdíly a změny v šířkách znaků a v kerningových párech.

Snahou článku, který byl originálně psán pro Zpravodaj a není tudíž překladem jiných publikací, je informovat o výše zmíněných aktivitách, orientovat se hlavně na témata spojená s československou sazbou a částečně také seznámit čtenáře s některými vlastnostmi fontů, které by nemusely být všem uživatelům $\text{\texttt{\textit{T}}E\text{\textit{X}}}$ u, $\text{\texttt{\textit{L}}\textit{A}T\text{\textit{E}}\text{\textit{X}}}$ u a jim příbuzných programů všeobecně známy.

Musím upozornit čtenáře, že tento článek je právě jen součástí činností spojených s prověřováním Latin Modern fontů. Jeho úkolem není

1. ani přehled LM-fontů samotných,
2. ani $\text{\texttt{\textit{L}}\textit{A}T\text{\textit{E}}\text{\textit{X}}}$ ovou kuchařkou pro sazbu s LM-fonty,
3. ani vysvětlení fontových formátů.

jelikož vyřešení každého z těchto zadání s dostatečně širokým pokrytím tématu by pravděpodobně vyžadovalo množství práce a úsilí v rozsahu samostatného článku.

2. METAFONTové bitmapové fonty

Fonty nejsou od Boha. Proto se nedivme, že nejsou dokonalé a mohou obsahovat nedostatky, zjevné nebo podle okolností i skryté.

Přeskočíme tiskárny a další zařízení, která u nás používaly počítače pro tisk do 80. let 20. století. Vynecháme i to, jak vypadaly fonty programu *xwriter*. Krátce si připomeňme, že v době, kdy vznikaly *ℒ*-fonty, jsme měli jen jehličkové tiskárny. A jaký to byl pokrok v kvalitě tisku, když jsme poprvé mohli tisknout na laserové tiskárně s hustotou 300 dpi! V tomto rozlišení naše první písmena v abecedě s diakritikou vypadala takto: *Áá*(*csr10*), *Áá*(*csbx10*) a ve zvětšení takto:

Potřebujeme pro praktickou produkci dokumentů ještě dnes METAFONTové fonty, se kterými jsme byli tak spokojeni začátkem 90. let? Vrátím se nyní ve svých úvahách o odstavec zpět, jak vypadají bitmapové fonty v různých rozlišeních. Poznamenejme, že METAFONT produkuje grafické podoby znaků (glyfy) pouze jako bitové mapy, zvlášť pro každé zařízení, na kterém budeme tisknout. Přitom rozlišení a výsledná vyrašťovaná množina pixelů bitové mapy závisí na koncovém výstupním zařízení, bude například jiná pro běžnou tiskárnu a jiná pro osvitovou jednotku vydavatelství.

I v následujících ukázkách vystačíme pouze s malým a velkým „a“ a s čárkou.

Vidíme, jak vypadají zvětšené bitové mapy znaku *Á* z oficiální verze fontu *csbx10*, jak je vygeneruje METAFONT v rozlišeních 600, 1200, 2400, 2602 a 5333 (dpi). Dále připojím ještě pár srovnání, jak se liší samotné „a“ z *csr10* v rozlišení 1200 a 2400 a jak vypadá jejich porovnání s verzí ve formátu Type 1.

Zleva jsou rozlišení 1200 a 2400, uprostřed jejich vzájemné srovnání, poté porovnání 1200 dpi s obrysovou reprezentací Type 1 a zcela vpravo je PostScriptová verze Computer Modern CMR10 ve formátu Type 1, kterou vytvořili BlueSky a Y&Y a od roku 1997 je volně šířena pod copyrightem AMS.

3. Co jsou LM-fonty

3.1. Autorský kolektiv GUST

Hlavními postavami týmu GUST e-foundry a autory vlastních Latin Modern fontů ve formátech Type 1 a OpenType a také programového balíku **MetaType1** jsou Bogusław Jackowski a Janusz M. Nowacki. Manažerské funkce včetně komunikace plní Jerzy Ludwichowski, který se účastní nebo pomáhá při psaní článků a jiných dokumentů v angličtině, někdy vystupuje i jako přednášející na mezinárodních setkáních; Marcin Woliński vytváří podporu pro $\text{\LaTeX}$; Stanisław (Staszek) Wawrykiewicz má na starosti zařazování balíků do CTAN a $\text{\TeX}$ Live.

3.2. Co nabízejí LM-fonty

Latin Modern fonty byly vytvořeny jako rozšíření Computer Modern fontů, a to současně v několika aspektech:

- Glyphy jsou definovány v obrysové (vektorové, outlinové) reprezentaci ve formátech Type 1 a OpenType.
- LM se snaží zahrnout všechny latinkové abecedy včetně všech písmen s diakritikou, aby pokryly množinu znaků všech evropských jazyků, které používají latinku, a také některých mimoevropských.
- Kolekci doplnily chybějící skloněné varianty celé řady fontů, též ji rozšířila skupina světlých a polotučných strojopisných fontů, rovněž včetně jejich skloněných řezů.
- OpenType verze (PostScript CFF flavored) dědí definice obrysových křivek a hintů přímo z Type 1.

- Distribuce zahrnuje i podporu $\text{T}_{\text{E}}\text{X}$ u: metrická data pro 9 různých kódování a také soubory pro podporu programů $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, dvips, pdf $\text{T}_{\text{E}}\text{X}$, $\text{CON}\text{T}_{\text{E}}\text{X}\text{T}$, $\text{X}_{\text{F}}\text{T}_{\text{E}}\text{X}$ atd.
- Velká pozornost byla věnována kvalitě provedení. Technické řešení spojuje věrnou a kvalitní aproximaci METAFONT ových zdrojů a optimalizaci datových struktur.

Znaková sada umožňuje vkládat do českých a slovenských textů v originále cizí slova a jména v mnoha jazycích, aby byl zachován uniformní tvar akcentů a jednota jejich umístění. Uvedeme několik příkladů z francouzštiny a polštiny: Genève Mâcon Nîmes Samoëns Bohême tchèque française Gdańsk Grudziądz Wałęsa.

3.3. Budování LM-fontů

Autoři LM-fontů vycházejí z již existující PostScriptové verze CM-fontů, přebírají a v případě potřeby modifikují akcenty nebo nově vytvářejí další diakritická znaménka nebo jiné nové znaky, pokud ve fontu chyběly. Celé fonty, grafická podoba znaků i metrické údaje, byly převedeny do popisu v jazyce METAPOST . Programový systém MetaType1 (METAPOST s knihovnou maker a dalších programových a datových souborů) pak dokáže pomocí přehledné notace efektivně popsat, jak z písmen a akcentů skládat rozsáhlou sadu akcentovaných písmen, a v exekutivním kroku automaticky vygenerovat jak PostScriptové fonty ve formátu Type 1, tak i další přidružené fontové soubory nutné pro zpracování $\text{T}_{\text{E}}\text{X}$ em i jinými programy. Následujícím krokem je vytvoření reprezentace ve formátu OpenType. Zároveň byl vytvořen rozsáhlý aparát podpory pro $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a dalších prostředků pro automatické zařazení LM-fontů do procesů sazby.

3.4. Přejít k OpenType fontům

OpenType je univerzální formát fontů navržený pro všechny operační systémy i všechny aplikační programy. Type 1 fonty sice obsahují celý repertoár znaků, ale nejvýše 256 z nich může být dostupných v rámci některého definovaného kódování, jejich použití $\text{T}_{\text{E}}\text{X}$ em a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ em vyžaduje příslušnou nadstavbu jako metrické údaje a mechanismus dělení slov pro daná kódování, tak i mapování znaků a fontů pro ovladače jako dvips a pdf $\text{T}_{\text{E}}\text{X}$ (pdf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$), které zařídí substituci znaků odpovídajícími glyfy při produkci výsledných dokumentů ve výstupních datových formátech PS a PDF, uvnitř nich jsou pak příslušné fonty vnořeny.

Fontový formát OpenType spojuje dohromady globální data o fontu, grafické údaje o tvarech znaků, metrické údaje o rozměrech znaků a navíc i další informace pro řízení komplexního zpracování fontů, které byly v dřívějších formátech rozděleny do dvou nebo i více samostatných datových souborů, přitom ještě odlišnými způsoby v různých operačních systémech, sázecích nebo jiných aplikačních

programech. Zahrnuje i popisy pro pokročilé zpracování textu, které se dosud vykonávaly pomocí speciálních preprocesorů a postprocesorů. Ruší se omezení na objem znakových sad z počtu pouhých 256 prvků na současnou maximální hodnotu kódu znaku 10FFFF v kódování UNICODE v šestnáctkovém číselném vyjádření.

OpenType fonty povolují jen celočíselné hodnoty proměnných včetně metrických údajů, a protože LM-fonty (jakož i mnohé jiné METAFONTové fonty) tuto podmínku nesplňují, není zaručena úplná zpětná kompatibilita.

4. Naše spolupráce na vývoji LM-fontů

Odůvodněnost naší spoluúčasti (Karla Horáka, mne a dalších) potvrdily naše návrhy na úpravy nebo změny některých znaků, zejména akcentů. Za důležitá by mohla být považována odhalení chyb, hlavně těch závažných. Byly to nejen nechtěné omyly autorů LM, ale přišlo se i na defekty v původních zdrojích, které byly převzaty jako stavební kameny při procesu skládání a budování LM-fontů (což byly CM, EC, PL, $\mathcal{G}$, VN a řada dalších i různých podobách: METAFONT, METAPOST, samostatné metriky, virtuální fonty, Type 1, Type 3, aj.). Věříme, že naše připomínky (konstruktivní i negativní) přispěly k tomu, aby na CTAN a T_EXLive měli uživatelé tyto fonty ve vysoké kvalitě a s minimálním množstvím nedostatků, i když si musíme uvědomit, že „celoevropský“ font pokrývající mnoho jazyků s latinkovými abecedami (i mimo Evropu) bude vždy výsledkem kompromisů a nemůže současně vyhovovat navzájem různým lokálním požadavkům a rozdílným zvyklostem a tradicím. Při tomto uniformním sjednocení musíme zanedbat některé místní konvence jednotlivých jazyků. Z druhé strany, jednotný font, technicky kvalitní následovník CM a přitom přiměřeně přijatelný všem, jako výsledek společných aktivit, vyjde levněji než pracovat najednou na několika obdobných fontech, které by se prakticky odlišovaly od sebe jen málo.

4.1. Naše návrhy

Karel Horák přispěl řadou připomínek k úpravám tvaru a umístění českých a slovenských diakritických znamének, zejména čárek, háčků a „mäkčeňov“, mimo jiné ve dvou svých přednáškách v Bachotku (2005 a 2007), zasloužil se i o odhalení řady chyb.

Vývoj LM-fontů jsem sledoval od jejich samotných začátků a prvních výstupů v roce 2003. Nicméně první seriózní analýzy jsem začal až v roce 2005, kdy byly na březnovém EuroT_EXu vedeny diskuse o akcentech, zejména o návrzích kroužkovaných „u“, k nimž jsem již v dubnu 2005 vytvořil návrh „Ůů“ přímo v MetaType1, který reálně neznamenal opticky nějak ohromnou změnu oproti návrhu autorů. Tyto mé návrhy byly s malými modifikacemi zařazeny do verze

Předvedeme současně provedení kroužkovaných „u“, nejprve ve své originální velikosti: ŮůŮũŰűŲűŴűŶűŸű, a poté ve zvětšení:

Následující příklady ilustrující navržená „Ů“ ve fontech `lmbx10`, `lmbx5`, `lms110`, `lmti10`, `lmss10` a `lmtt10` v porovnání s PostScriptovou verzí $\mathcal{G}$ -fontů, které představují obrysové křivky, LM-fonty jsou vyplněné (obr. 1).

V následujícím výkladu uvedeme zkrácený popis testovacích procesů verzí LM-fontů z let 2005 a 2006, upozorníme na významné změny a ukážeme závažné chyby, které „musely“ být opraveny. Vysvětlíme všeobecné postupy testování metrických a grafických dat. Ukázky nyní odložíme, budeme je demonstrovat až na vybraných reálných případech spolu s konkrétními způsoby v příslušných dalších kapitolách.

Porovnání metrik bylo prováděno s několika významnými omezeními. Zadání jsme zúžili na kódování T1, pokrývající potřeby české a slovenské sazby, protože mimo jiné obsahuje všechna písmena české a slovenské abecedy včetně akcentovaných. Kromě toho zde najdeme i jiná písmena s diakritikou dovolující konzistentně vysázet např. cizí jména z řady dalších evropských jazyků. T_{EX}ové metrické soubory pro kódování T1 jsou v distribuci LM označeny jmény začínajícími „ec-lm“. Kódování T1 se také často nazývá „Cork“ podle místa konání konference TUG v Irsku v roce 1990, kde bylo dohodnuto a schváleno jako standard pro připravované evropské EC-fonty, odtud další název kódování EC.

8

Obrázek 1: Současné provedení kroužkovaných „u“ (zvětšeno)

T_EXem nejdůležitější: šířka znaku, kerningové páry a ligatury. Testování dalších položek metrických dat provedeno nebylo. Metriky pro T1, které jsou součástí L^AT_EXové podpory LM-fontů, srovnáváme s metrikami v kódování IL2, které používají G₂-fonty. Úkolem bylo zjistit, zda se shodují šířky všech znaků v obou těchto případech, které uvažujeme pro československou sazbu; také prokázat, že se shodují množiny dostupných ligatur (slitků); dále navzájem srovnat kerningové páry, podle hodnot kernů je rozdělit na shodné a rozdílné. Vyhodnocení má posoudit, zda jsou v LM údaje o kernech přijatelné. Srovnání se samozřejmě omezují na společnou podmnožinu (průnik) znaků, které se vyskytují v obou kódováních.

Nemám k dispozici nějaké rozsáhlé slovníky českých nebo slovenských slov, ani archiv vzorových textů nebo výběr testovacích slov, které by sloužily k namátkovým kontrolám, zda některé nevhodné nebo chybějící kerningové páry nepůsobí negativní efekty při sazbě.

Můj postup je jiný, spočívá v tom, že v prvním kroku generuji *všechny* kerningové páry tvořené dvojicí znaků, které oba patří do množiny znaků společné

současné pro metriky z dvou kódování IL2 ($\mathcal{G}$) a T1 (`ec-lm`). V dalších krocích kerningy vyhodnocuji, např. srovnávám a třídím. V případech, že pro dvojici sousedních znaků v textu kerningový pár neexistuje v žádném z obou kódování nebo hodnoty kerningového posunu jsou stejné, obě metriky zajistí, že se v textu vysázeném příslušnými fonty sousední znaky umístí stejně, stejně dobře nebo stejně špatně, tedy nezávisle na tom, kterou metriku použijeme. Pro podseznam stejných kerningů můžeme navíc prohlédnout, zda některé kerny nebyly navrženy nevhodně; pro obě metriky najednou, neboť jsou totožné. Pro množinu 52 fontů pro $\mathcal{G}$ a LM společných a pro všechny rozdílné kerningy (jejich celkový počet je „pouze“ 27680 z celkového počtu kerningových párů 547321 ve verzi 1.010). Dále v tomto článku ukážeme ručně vybrané případy.

5.2. Kontrolní tisky glyfů

První typ zobrazuje jednotlivé glyfy, tvary znaků v grafickém vyjádření. Tyto kontrolní (testovací) tisky pro *obrysové* fonty jsou obdobou zobrazení, která produkuje METAFONT. Podobné možnosti nabízejí obrázky, které vytvářejí MetaType1 a FontForge, v tomto článku ale nebudeme vysvětlovat jejich přednosti, nedostatky nebo odlišnosti vzhledem s mému přístupu. Mé obrázky generuje pdfL^AT_EX, byl zachován rozměr A4, aby je šlo přímo tisknout, v praxi obvykle jen po jednotlivých stranách. Vyobrazení glyfů jsou celostránková, barevná, obsahují zvětšený obraz glyfu, ke kterému jsou přidány: obrysové křivky, uzlové a řídicí body Bézieových křivek spojené řídicími vektory; jejich rozměry byly speciálně zmenšeny, aby nepřekážely prohlížení detailů. Částečně jsou znázorněny úsečkami základní dotažnice, vertikály vyznačují nulovou souřadnici a šířku znaku, pro fonty s nenulovým úhlem sklonu (ItalicAngle) jsou navíc doplněny odpovídající šikmé úsečky. Je doplněna nahoře a/nebo dole krátká jednořádková popiska, v níž jsou jméno fontu, jméno glyfu a jeho šířka, což je obvykle pro praktické vizuální prohlížení postačující.

Acrobat Reader umožňuje rychlé stránkování, zoom a tím i efektivní prohlížení. Soubory v PDF jsou navíc i prohledávatelné, glyfy lze hledat podle jména.

Pro každý font Type 1 nebo OpenType lze příslušný kontrolní tisk se všemi glyfy vygenerovat po určité přípravě již téměř automaticky.

5.3. Srovnávací tisky dvou glyfů

Druhý typ testovacích tisků zobrazuje výsledky podrobných srovnání dvou glyfů ze dvou verzí nebo vydání téhož fontu. Souhrnným výsledkem je soubor všech glyfů, v nichž byl zjištěn nějaký rozdíl v konturních obrysovéch křivkách, popřípadě v šířce glyfu; tento údaj je obsažen v metrických i grafických datech fontu. Tím získáme přehled o všech změnách ve vzhledu glyfu, tak i o změnách jeho umístění, tzn. o rozdílech v horizontálním posunu mezi dvěma sousedními vysázenými

znaky. V latinkové sazbě se znaky skládají vedle sebe, vertikální posun mezi sousedními znaky je nulový, což platí také pro LM-fonty.

6. Chronologie konkrétních testů a srovnávání

Navážeme na obecné popisy testů a ukážeme vybrané výsledky reálných prověrek provedených v průběhu posledních tří let.

6.1. Jak jsem k LM-fontům přišel

Dosud jsem nebyl autorem ani spoluautorem žádné publikace v češtině (ani ve slovenštině). S *ruským* L^AT_EXem jsem začal jako uživatel až začátkem 90. let 20. století. Můj hlubší aktivní zájem, hlavně o fonty, a to zejména exotické, se formuje teprve od roku 1995. Tak se stalo, že jsem C^S-fonty nikdy přímo nepotřeboval, ani METAFONTové, ani PostScriptové. Zájem o MetaType1, jako programovatelný nástroj na tvorbu fontů ve formátu Type 1, způsobil nutnost studovat zdrojové texty pro generování LM-fontů, neboť v době svého zrodu byly jediným významným příkladem jeho aplikace. To mne přivedlo ke zkoumání a poté i prověřování těchto fontů, i když jsem se zpočátku vůbec nemínil latinkovými písmy zabývat, ale chtěl MetaType1 používat pro písma exotická. V dalším kroku/etapě se pak stalo, že bylo třeba LM-fonty srovnávat s CM a C^S, a to až v roce 2005. A později, vlastně nechtěně (nebylo to zadání grantu), došlo i na CM a C^S fonty v bitmapové reprezentaci.

7. LM verze 0.99 (2005)

Obsáhlejší a úplnější prověrky započaly od verze 0.99.3 zveřejněné na podzim 2005. Po vygenerování kompletních testovacích tisků byly vizuálními kontrolám podrobovány všechny glyfy ze všech LM-fontů bez omezování se na „československé potřeby“, i když těm ostatním byla věnována menší pozornost.

Následují ukázky testovacích tisků ve dvou variantách (s výplní a bez výplně), jak vypadají ve zmenšení celé stránky s tvary znaků, popisnými texty, případně také hintovými zónami (obr. 2).

V reakci na má hlášení autoři průběžně opravili pár chyb. Uvádím zde příklad opravené šířky znaku Ø, která se v polotučném a bezpatkovém řezu musí vypočítávat jinak; v METAFONTu je zadán parametr `width_adj` jako nulový. Správná je vertikála vpravo, odpovídající větší šířce znaku. Obrázek 3 byl převzat z protokolu demonstrujícího chybu a následnou prověrku opraveného stavu.

LMRoman10-Regular: acute at 500 pt; WX 500

LMRoman10-Regular: caron at 500 pt; WX 444

LMRoman10-Regular: ecaron at 500 pt; WX 444

Obrázek 2: Ukázka testovacích tisků

Figure 1: lmbx10/LMRoman10-Bold:Oslash

Figure 2: lmbx12/LMRoman12-Bold:Oslash

Figure 3: lmbx5/LMRoman5-Bold:Oslash

Obrázek 3: Příklad opravené šířky znaku Ø

7.1. Prověrky metrik

Prvnímu podrobnějšímu srovnávání metrik, resp. šířkám znaků, které obsahuje i formát Type 1, jsem se začal věnovat až na podzim roku 2005. Překvapivě bylo zjištěno velké množství rozdílů v metrikách mezi LM a ζ -fonty. Na příčinu

rozdílů přišli sami autoři LM. Systematický rozdíl je totiž mezi metrikami CM a EC-fontů. Dokonce ani nejsou zachovány identické šířky společných znaků. Ale nikdo si toho více než 10 let nevšiml. Přitom se stačilo podívat do souboru `tfm`, přesněji provést srovnání se soubory `p1`, což je čitelná textová forma metrik, anglicky nazvaná jako „property list“.

Rozdíly způsobuje aritmetika METAFONTu a makro `gendef` z balíku EC-fontů pro výpočet jednotky `u#` a od ní odvozených rozměrových údajů. Proto se ani nemohly shodovat metriky ζ s metrikami LM v kódování T1, jejichž kerningové páry byly ve verzi LM 0.99 převzaty z EC-fontů.

Tedy jsem zjevně udělal správně, že jsem koncem roku 2005 nepřijal první nabídku grantu a nepustil se do podrobných srovnávání, zpracování a vyhodnocování veškerých dat. Kdybych té práci opravdu tenkrát obětoval svůj čas a provedl ji „včas“, tj. vypracoval pro Bacht_{EX} 2006 podrobnou zprávu s kompletní analýzou rozdílů mezi ζ a LM, museli bychom pak konstatovat následující:

„Verze 0.99.3 byla vyhozena, proto i výsledky testování můžeme celé vyhodit.“

7.2. Hry s metrickými daty

Přesto jsem se dodatečně rozhodl, že by bylo vhodné dopsat a vložit podrobnější vysvětlující komentáře, které se dobře hodí do samotného tématu testování fontů. Několik různých reprezentací hodnot číselných dat může být zdrojem diferencí a problémů. Z metrických údajů budeme v následujících pasáží studovat pouze dva: šířku znaku a velikost posunu (kern) mezi dvěma znaky, které jsou nejdůležitější pro formátování obvyklých latinských textů. Vliv dalších údajů lokálních pro jednotlivé znaky (výška, hloubka, italská korekce) nebývá při sazbě prostých textů dominantní. Zmíníme se, ve kterých fontových formátech jsou tyto parametry znaků obsaženy, jakými způsoby jsou reprezentovány jejich hodnoty, z čeho principiálně pramení potenciální rozdíly, které rozdíly považujeme za zanedbatelné a které jsou významné.

Pro posouzení kompatibility nebo neshody výsledných výstupů při srovnávání metrických dat ζ -fontů a LM-fontů v kódování T1 (a dalších příbuzných) je důležité porovnat různé datové formáty vstupující do hry v procesu tvorby a generování fontů, při formátování T_{EX}em (L^AT_{EX}em) a při produkci koncových dokumentů v PS a PDF.

V diverzitě a nejednotnosti číselných reprezentací je a priori založena nemožnost vyjádřit přesně, shodně a optimálně stejné číselné hodnoty. Cílem je dosáhnout invariantních reprezentací, pro které by vzájemné konverze zachovávaly identické hodnoty čísel (v povoleném definičním oboru). Omezíme se pouze na stručné charakteristiky datových formátů a rozdíly doložíme na relevantních příkladech.

Zásadní a praktický princip při porovnání dvou verzí jednoho fontu spočívá

v tom, že když některé hodnoty nejsou identické, automaticky to znamená, že došlo ke změnám. Proto „rozmazaná“ data, kdy hodnoty, které by měly být stejné, nejsou reprezentovány identicky, komplikují a zdržují operace srovnávání a tím i celá testování. Pokud se ve dvou fontech nebo ve dvou formátech hodnoty parametrů liší velmi málo, musíme nastavovat kritéria určující zanedbatelnost rozdílů a míry přípustnosti zaokrouhlovacích chyb.

Výklad provedeme na případech metrik z fontů Roman Regular a Roman Regular Bold. Tabulky ukazují, že v metrických datech, např. šířkách znaků, je systematický rozdíl mezi EC-fonty a staršími verzemi LM na jedné straně a na druhé straně CM, $\mathcal{C}$ S a LM ve verzi aktuální. V METAFONTu je pro CM definována jednotka $u\#$, která je vyjádřena zaokrouhleně v pseudoreálné (lze říci i pseudoceločíslné) reprezentaci. Proměnné od ní odvozené jako šířky znaků ($9u\#$, $13u\#$) nebo velikosti kernů ($k\#$, $kk\#$, $kkk\#$) jsou pak její *přesné* násobky.

Například interní hodnota, se kterou pracují T_EX a METAFONT, vyjádřená *nepřímo* desetinným rozvojem 0.833336, je opravdu přesným trojnásobkem interní hodnoty pro 0.277779, neboť skutečné interní reprezentaci s konečnou délkou mantisy vzájemně jednoznačně odpovídá právě dané *nejbližší* číslo v desetinném zápisu na daný počet míst.

Pro (METAFONTový) font `cmr10` je dosazena hodnota jednotky $u\#:=20/36pt\#$. Ale autoři EC-fontů si za základ výpočtů vybrali $1/36pt\#$. LM-fonty parametrizovaný způsob výpočtu vůbec nepotřebují, naopak je žádoucí, aby již dříve zafixované metrické údaje mohly být vyjádřeny ve všech fontových formátech ekvivalentně ve všech případech, kde je to možné.

Přehled vybraných metrických údajů pro fonty `cmr10.tfm/pfb`, `csr10.tfm/pfb`, `EC:ecrm1000.tfm/sfrm1000.pfb` a `LM:ec-lmr10.tfm/lmr10.pfb/otf` ukazuje tab. 1. Metriky CM (šířky jednotlivých znaků, např. „A“, „a“, „i“ a „m“, jsou v CM a $\mathcal{C}$ S stejné) jsou založeny na přesné reprezentaci jednotek v METAFONT, např. 0.750002 nebo 0.500002 (viz TFM). Naopak Type 1 (viz řádky označené PS) omezuje vyjádření čísel jako podíly celých čísel omezené velikosti, např. 2500/9, o řádku níže je přibližná desítková reprezentace) a v OpenType jsou povolena pouze celá čísla.

Jak vidíme v tabulce 2, šířky znaků se v CM, $\mathcal{C}$ S a LM prakticky shodují, až na rozdíly, které jsou zanedbatelné. Porovnáme-li položky v řádkách TFM, PS a AFM se pro CM, $\mathcal{C}$ S a LM, rozdíly jsou „menší než malé“. Navíc můžeme najít různé kombinace identických hodnot, hodnota některého z parametrů v LM může být totožná s CM, jiného naopak s $\mathcal{C}$ S, jako třeba šířka znaku „a“ v `lmbx10` a `csbx10` (výsledek V. Malýševa) je stejná, ale v `cmbx10` se mírně neshoduje.

Zato metrická data v EC-fontech a také v CM-super (viz SF PS) jsou systematicky odlišná. Následující tabulky 3 a 4 ukazují difference v hodnotách základních metrických jednotek a kerningů a demonstrují celkovou nekompatibilitu EC-fontů. A stará verze LM 0.99.3 už patří minulosti.

Tabulka 1: Vybrané metrické údaje pro fonty cmr10, ecr10, csr10, lmr10

	WD("A")	WD("a")	WD("i")	WD("m")
CM MF	20/36pt#*13 + 5/36pt#*2	20/36pt#*9	20/36pt#*5	20/36pt#*15
TFM	0.750002	0.500002	0.277779	0.833336
PS	750	500	2500/9	2500/3
dec)	750	500	277.77778)	833.33333)
AFM	750	500	277.778	833.333
CS TFM	0.750002	0.500002	0.277779	0.833336
PS	750	500	5000/18	15000/18
dec)	750	500	277.77778)	833.33333)
EC MF	1/36pt#*20*13 + 1/36pt#*5*2	1/36pt#*20*9	1/36pt#*20*5	1/36pt#*20*15
TFM	0.749817	4.99878	0.27771	0.83313
SF PS	53237/71	20495/41	27771/100	64151/77
dec)	749.8169)	499.87805)	277.71	833.12987)
AFM	749.817	499.878	277.71	833.13
LM MP	0.75	0.5	277.778	833.333
TFM	0.75	0.5	0.277776	0.8333
PS	750	500	2500/9	2500/3
dec)	750	500	277.77778)	833.33333)
AFM	750	500	277.77777	833.33333
OTF	750	500	278	833

8. LM verze 1.00 (duben 2006)

Metriky v kódování T1, původně převzaté z EC-fontů, byly ve verzi 1.00 kompletně přepočítány tak, aby byly zpětně kompatibilní s CM-fonty (a tedy nepřímě i s $\mathcal{C}$ -fonty). Po srovnání s předchozí verzí 0.99.3 došlo též ke změně glyfů v 24964 případech (automaticky spočítáno při srovnání všech glyfů ve všech 69 společných fontech). Kolekce byla rozšířena o tři textové fonty (na celkový počet 72) a 20 fontů matematických.

Tato verze byla připravena podle plánu k mítingu GUST a byla vystavena na internetu v den mého příjezdu do Bachotku. Pro podmnožinu znaků, které jsou v LM společné s CM, se jejich šířky shodují, až na několik málo nepodstatných případů. Bohužel bylo „vyrobeno“ i několik nových nechtěných chyb v glyfech. Dvacet matematických fontů (pouze jako Type 1) je de facto jen kopie CM-fontů, ty se ale sazby československých textů bezprostředně netýkají.

Tabulka 2: Šířky znaků ve fontech cmbx10, ecxb1000, csbx10, lmbx10

	WD("A")	WD("a")	WD("i")	WD("m")
CM MF	23/36pt#*13 + 7/36pt#*2	23/36pt#*9	23/36pt#*5	23/36pt#*15
TFM	0.86944	0.559024	0.319443	0.958328
PS	13911/16	25156/45	2875/9	2875/3
dec)	869.4375)	559.02222)	319.44444)	958.33333)
AFM	869.438	559.022	319.444	958.333
CS TFM	0.86944	0.559024	0.319443	0.958328
PS	15650/18	559	5750/18	17250/18
dec)	869.44444)	559	319.44444)	958.33333)
EC MF	1/36pt#*23*13 + 1/36pt#*7*2	1/36pt#*23*9	1/36pt#*23*5	1/36pt#*23*15
TFM	0.869232	0.558891	0.319366	0.958099
SF PS	48677/56	25709/46	9581/30	9581/10
dec)	869.23214)	558.8913)	319.36667)	958.1
AFM	869.232	558.891	319.366	958.099
LM MP	869.438	559	319.444	958.333
TFM	0.8694	0.559	0.31945	0.9583
PS	13911/16	559	2875/9	2875/3
dec)	869.4375)	559	319.44444)	958.33333)
AFM	869.4375	559	319.44444	958.33333
OTF	869	559	319	958

Tabulka 3: Velikosti jednotek ve fontech cmr10, ecrm1000, lmr10

	u#	k#	kk#	kkk#
CM MF	20/36pt#	-0.5*20/36pt#	-1.5*20/36pt#	-2*20/36pt#
TFM	0.055556	-0.027779	-0.083334	-0.111111
EC MF	1/36pt#*20	-1/36pt#*20*0.5	-1/36pt#*20*1.5	-1/36pt#*20*2
TFM	0.055542	-0.027771	-0.083313	-0.111084
LM 0.99.3	0.055542	-0.027771	-0.083313	-0.111084
LM 1.010	0.055555	-0.027779	-0.083333	-0.111112

8.1. Prověřování glyfů

V důsledku návrhů, připomínek a konzultací došlo ve verzi LM 1.00 k velkému množství změn, nejčastěji v umístění akcentů nad velkými písmeny. Uvedeme jen několik ilustračních případů (obr. 4) z celkovém počtu v porovnání s verzí 0.99.3 (vyznačena nevyplněnými obrysovémi křivkami).

Tabulka 4: Srovnání kerningových jednotek v CM-fontech a EC-fontech

Kerningy	cmr10, ecrm1000, ec-lmr10			cmbx10, ecbx1000, ec-lmbx10		
CM MF	k#	kk#	kkk#	k#	kk#	kkk#
TFM	-0.027779	-0.083334	-0.111111	-0.031944	-0.095833	-0.127777
AFM	-27.778	-83.333	-111.111	-31.942	-95.831	-127.775
EC MF	k#	kk#	kkk#	k#	kk#	kkk#
TFM	-0.027771	-0.083313	-0.111084	-0.031937	-0.09581	-0.127747
SF AFM	-27.771	-83.313	-111.084	-31.937	-95.81	-127.747
LM 0.99.3						
T1 TFM	-0.027771	-0.083313	-0.111084	-0.031937	-0.09581	-0.127747
LM 1.010						
MP	-27.778	-83.333	-111.111	-31.944	-95.833	-127.778
T1 TFM	-0.027779	-0.083333	-0.111112	-0.031944	-0.095833	-0.127779
AFM	-27.778	-83.333	-111.111	-31.944	-95.833	-127.778
OTF	-28	-83	-111	-32	-96	-128

8.1.1. Zkoumání čs. akcentovaných písmen

Ve všech fontech z kolekce LM byla speciálně posuzována vybraná podmnožina písmen s diakritikou, vesměs společných s ζ -fonty. Tabulka obsahující seznamy sestavené z velkých a malých písmen, na příkladu fontu `lmr10/LMRoman10-Regular`, je uvnitř jednou uspořádána podle akcentů a podruhé seřazena podle společných základních písmen (obr. 5).

Je ukázána (k ověření) jednotná koncepce návrhu a výškového umístění akcentů kapitálových, které jsou zploštělé a níže nad písmenem, a akcentů verzálových, které byly usazeny výše, v obou případech opticky vyrovnány okolo jakýchsi pomyslných horizontál.

Podobné tabulky byly vytvořeny a pak zkoumány pro každý textový font z kolekce LM.

8.2. Nechtěné chyby

Reálná přítomnost nežádoucích chyb ukazuje, že zevrubné testování fontů je nezbýtná činnost, kterou nelze vynechat. Po změnách některých glyfů se občas stalo, že nikdo z autorského týmu neprovedl vyčerpávající prověrku nového stavu. Ve verzi 1.00 se „podařilo vyrobit“ zjevně nezamýšlené chyby. Stalo se tak například u znaků „perthousand“ a „permyriad“ (obr. 6), kde se vinou nesprávného směru křivky vyplnily i vnitřky některých „nul“.

Pokus o drobné vylepšení, oprava defektu „šestek“ a „devítek“ ve fontech `lmtcsc10` a `lmtcso10`, se nepovedl (obr. 7). První pokus skončil fiaskem, omylem se přepsala písmena „F“ a „I“, včetně odvozených písmen s diakritikou. Po rychlém (a asi spěšném) návratu zpět se malý zub už nezarovnal (obr. 8), nicméně lze to považovat za tolerovatelnou poruchu nižšího řádu.

LMRoman10-Regular: Ascender: WX 750

LMRoman10-Regular: Ascender: WX 750

LMRoman10-Regular: Cxascender: WX 722.222

LMRoman10-Regular: Eascender: WX 680.556

LMRoman10-Regular: Rascender: WX 756.111

LMRoman10-Regular: Oxcircumflex: WX 777.778

LMRoman10-Regular: dxascender: WX 555.556

LMRoman10-Regular: txascender: WX 388.889

LMRoman10-Regular: acute: WX 500

LMRoman10-Regular: oxcircumflex: WX 500

Obrázek 4: Ilustrační příklad změn mezi verzemi 0.99.3 a 1.00

Ačkoliv jsem měl k dispozici vygenerovány úplné testovací tisky všech glyfů, některé chyby jsem přehlédl i já sám. Karel Horák pak objevil neprávne orientace směru vnitřní křivky ve znacích „onequarter“ a „threequarters“, která způsobuje, že malá „čtyřka“ je celá vyplněná ve čtyřech ze skupiny strojopisných řezů, konkrétně „dark“ (l_{mv}tk10 a l_{mv}tko10) a „light“ (l_{mt}l10 a l_{mt}lo10), přitom v ostatních 16 strojopisných fontech se tento jev nevyskytl.

1/4 3/4 1/4 3/4 1/4 3/4 1/4 3/4 1/4 3/4 1/4 3/4 1/4 3/4

Font je jako živý organismus. Dosud se mění a jeho vývoj se ještě nezastavil. Je tedy pochopitelné, že ani autoři, ani my, kdo fonty prověřují, ani další přispěvatelé, nemohou zabránit vznikaní dalších chyb a tomu, že budeme schopni odhalit všechny důležité chyby přítomné v každé dané chvíli. Také tento článek je zastavení zachycující jen částečný pohled v určitém průběžném historickém okamžiku.

Obrázek 5: Velká a malá akcentovaná písmena z fontu lmr10 používaná v $\mathcal{C}_S$ -fontech

Proověrování metrických dat LM-fontů, jehož součástí bylo i srovnávání s příbuznými fonty (CM, $\mathcal{G}$, EC), probíhalo několikrát pro jednotlivé verze, vždy v několika etapách a každá z nich v několika krocích.

První metoda je „analytická“. Zkoumá metrické soubory `tfm` (převezené do textové formy `pl`), z nichž vyextrahujeme seznamy (tabulky) studovaných veličin, konkrétně šířek znaků a kerningových párů. Další údaje, jako výšky a hloubky znaků a italicické korekce, systematickému studiu podrobeny nebyly. Musím poznamenat, že jeden kerningový pár se může v `tfm` vyskytovat i víckrát. Důležité je, že `TEX` použije *první nalezený výskyt*! Doufám, že pořadí prohledávání je jednoznačné a můj způsob třídění a nalezení tohoto prvního výskytu je správný, totiž je stejný, jak to dělá `TEX`.

Následující tabulka ukazuje výsledek porovnání šířek znaků v metrikách `ec-lm` a `CM`, bylo nalezeno jen 9 rozdílů.

Figure 1: *lmri10/LMRoman10-Italic:perthousand*

Figure 2: *lmri12/LMRoman12-Italic:perthousand*

Figure 3: *lmri7/LMRoman7-Italic:perthousand*

Figure 4: *lmri8/LMRoman8-Italic:perthousand*

Obrázek 6: Ukázka chyby a opravy ve znacích „perthousand“ a „permyriad“

Figure 1: lmtcsc10/LMTypewriter10-CapsRegular:F

Figure 2: lmtcsc10/LMTypewriter10-CapsRegular:six

Figure 3: lmtcsc10/LMTypewriter10-CapsRegular:I

Figure 4: lmtcsc10/LMTypewriter10-CapsRegular:nine

Obrázek 7: Příklad neúspěšné změny a následné opravy

Obrázek 8: Detail znaků „6“ a „9“

dif	EC-LM	CM	olm	ocm	afm	ps	dec	otf	wlm	wcm
-0.029447	j ec-lmbxi10	cmbxti10	152	152	385	385	385	385	0.385	0.355553
-0.025560	j ec-lmri10	cmti10	152	152	332.222	2990/9	332.22223	332	0.332225	0.306665
-0.024997	j ec-lmri12	cmti12	152	152	325	325	325	325	0.325	0.300003
-0.029362	j ec-lmri7	cmti7	152	152	397.625	3181/8	397.625	398	0.39762	0.3682575
-0.027432	j ec-lmri8	cmti8	152	152	356.6	1783/5	356.6	357	0.3566	0.329168
-0.026237	j ec-lmri9	cmti9	152	152	341.054	12619/37	341.05405	341	0.341056	0.314819
-0.027766	j ec-lmu10	cmu10	152	152	361.111	3250/9	361.11111	361	0.361099	0.333333
0.008334	x ec-lmu10	cmu10	170	170	500	500	500	500	0.5	0.508334
-0.131855	sterling ec-lmu10	cmu10	277	044	826.333	2479/3	826.33333	826	0.8263	0.694445

Ze seznamu všech znaků společných pro kódování T1 a OT1 byly vybrány jen ty, jejichž šířky se v **tfm/pl** liší „více než zanedbatelně“. Hodnota rozdílu šířek v **tfm** (**CHARWD**) je přidána jako první položka **dif**. Další sloupce označují jméno znaku, dvě jména fontů v kódování EC (T1) a CM (OT1), dvojici osmičkových kódů **olm** a **ocm** (mohou být různé), pětkrát šířku znaků v LM, jak je zaznamenána v **afm**, v Type 1/ps (podíl **m n** **div** je zapsán zkráceně jako **m/n**), tentýž parametr vyjádřený desítkovým číslem **dec**, zaokrouhlená hodnota v OpenType (**otf**) a šířka z **tfm** (**wlm**). Řádek uzavírá stejný parametr z CM (**wcm**). A právě z posledních dvou hodnot byl vypočtem příslušný rozdíl šířek znaku **dif**. Ostatní reprezentace jsou vztaženy k souřadnicové soustavě glyfu 1000 × 1000, což odpovídá společné konvenci zvolené pro Type 1 i OpenType.

V dalších krocích pak položky odpovídající jednotlivým fontům navzájem porovnáme, rozdělíme na skupiny a podskupiny, abychom vybrali podseznamy pro aplikaci druhé metody.

Úvodní trojice v tabulce kerningových párů definují jména fontů a osmičkové kódy páru znaků, následují hodnoty kernů v obou fontech v číselné a symbolické

notaci (1k=k#, 3k=kk#, 4k=kkk# apod.). Na konci řádku jsou velikosti kernu v afm a otf a PostScriptová jména obou znaků.

Takto vypadá vybraná ukážka části tabulky se shodnými kerningy:

CS	EC-LM	cs	lm	afm	otf
csr10 101 103 ec-lmr10 101 103	-0.027779	1k	-0.027779	1k -27.778	-28 A C
csr10 101 107 ec-lmr10 101 107	-0.027779	1k	-0.027779	1k -27.778	-28 A G
csr10 101 117 ec-lmr10 101 117	-0.027779	1k	-0.027779	1k -27.778	-28 A O
csr10 101 121 ec-lmr10 101 121	-0.027779	1k	-0.027779	1k -27.778	-28 A Q
csr10 101 124 ec-lmr10 101 124	-0.083334	3k	-0.083333	3k -83.333	-83 A T
csr10 101 125 ec-lmr10 101 125	-0.027779	1k	-0.027779	1k -27.778	-28 A U
csr10 101 126 ec-lmr10 101 126	-0.111112	4k	-0.111112	4k -111.111	-111 A V
csr10 101 127 ec-lmr10 101 127	-0.111112	4k	-0.111112	4k -111.111	-111 A W
csr10 101 131 ec-lmr10 101 131	-0.083334	3k	-0.083333	3k -83.333	-83 A Y
csr10 101 164 ec-lmr10 101 164	-0.027779	1k	-0.027779	1k -27.778	-28 A t
csr10 101 166 ec-lmr10 101 166	-0.111112	4k	-0.111112	4k -111.111	-111 A v
csr10 101 167 ec-lmr10 101 167	-0.083334	3k	-0.083333	3k -83.333	-83 A w
csr10 101 253 ec-lmr10 101 224	-0.083334	3k	-0.083333	3k -83.333	-83 A Tcaron
csr10 101 273 ec-lmr10 101 264	-0.027779	1k	-0.027779	1k -27.778	-28 A tcaron
csr10 101 310 ec-lmr10 101 203	-0.027779	1k	-0.027779	1k -27.778	-28 A Ccaron
csr10 101 323 ec-lmr10 101 323	-0.027779	1k	-0.027779	1k -27.778	-28 A Oacute
csr10 101 324 ec-lmr10 101 324	-0.027779	1k	-0.027779	1k -27.778	-28 A Ocircumflex
csr10 101 326 ec-lmr10 101 326	-0.027779	1k	-0.027779	1k -27.778	-28 A Odieresis
csr10 101 331 ec-lmr10 101 227	-0.027779	1k	-0.027779	1k -27.778	-28 A Uring
csr10 101 332 ec-lmr10 101 332	-0.027779	1k	-0.027779	1k -27.778	-28 A Uacute
csr10 101 334 ec-lmr10 101 334	-0.027779	1k	-0.027779	1k -27.778	-28 A Udieresis
csr10 101 335 ec-lmr10 101 335	-0.083334	3k	-0.083333	3k -83.333	-83 A Yacute

Rozdíly mezi kerny považujeme za zanedbatelné, když se liší jen na posledním platném místě.

Obdobně ukážeme několik úseků tabulky, kde jsou kerningy odlišné:

CM	CS	EC-LM	cs	lm	afm	otf
cmr10 - 0k	csr10 101 143	ec-lmr10 101 143	- 0k	-0.027779	1k -27.778	-28 A c
cmr10 - 0k	csr10 101 144	ec-lmr10 101 144	- 0k	-0.027779	1k -27.778	-28 A d
cmr10 - 0k	csr10 101 145	ec-lmr10 101 145	- 0k	-0.027779	1k -27.778	-28 A e
cmr10 - 0k	csr10 101 157	ec-lmr10 101 157	- 0k	-0.027779	1k -27.778	-28 A o
cmr10 - 0k	csr10 101 161	ec-lmr10 101 161	- 0k	-0.027779	1k -27.778	-28 A q
cmr10 - 0k	csr10 101 165	ec-lmr10 101 165	- 0k	-0.027779	1k -27.778	-28 A u
cmr10 - 0k	csr10 101 171	ec-lmr10 101 171	- 0k	-0.083333	3k -83.333	-83 A y
cmr10 - -	csr10 101 350	ec-lmr10 101 243	- 0k	-0.027779	1k -27.778	-28 A ccaron
cmr10 - -	csr10 101 357	ec-lmr10 101 244	- 0k	-0.027779	1k -27.778	-28 A dcaron

```

cmr10 - - csr10 101 354 ec-lmr10 101 245 - 0k -0.027779 1k -27.778 -28 A ecaron
cmr10 - - csr10 101 371 ec-lmr10 101 267 - 0k -0.027779 1k -27.778 -28 A uring
cmr10 - 0k csr10 101 036 ec-lmr10 101 327 - 0k -0.027779 1k -27.778 -28 A OE
cmr10 - 0k csr10 101 037 ec-lmr10 101 330 - 0k -0.027779 1k -27.778 -28 A Oslash
cmr10 - - csr10 101 351 ec-lmr10 101 351 - 0k -0.027779 1k -27.778 -28 A eacute
cmr10 - - csr10 101 363 ec-lmr10 101 363 - 0k -0.027779 1k -27.778 -28 A oacute
cmr10 - - csr10 101 364 ec-lmr10 101 364 - 0k -0.027779 1k -27.778 -28 A ocircumflex
cmr10 - - csr10 101 366 ec-lmr10 101 366 - 0k -0.027779 1k -27.778 -28 A odieresis
cmr10 - 0k csr10 101 033 ec-lmr10 101 367 - 0k -0.027779 1k -27.778 -28 A oe
cmr10 - 0k csr10 101 034 ec-lmr10 101 370 - 0k -0.027779 1k -27.778 -28 A oslash
cmr10 - - csr10 101 372 ec-lmr10 101 372 - 0k -0.027779 1k -27.778 -28 A uacute
cmr10 - - csr10 101 374 ec-lmr10 101 374 - 0k -0.027779 1k -27.778 -28 A udieresis
cmr10 - - csr10 101 375 ec-lmr10 101 375 - 0k -0.083333 3k -83.333 -83 A yacute

```

Oproti předchozímu příkladu jsou dodatečně přidány (na začátek) tři sloupce s informací o kerningu v CM-fontech, abychom mohli porovnat kerningové páry ve všech třech fontech: LM, ζ i CM. Nepřítomnost kerningového páru v metrice je označena jako „- 0k“; „cmr10 - -“, je uvedeno, když v CM některý ze znaků neexistuje. Všeobecně „-“ znamená absenci znaku nebo kerningového páru.

```

cmr10 -0.027779 1k csr10 113 117 ec-lmr10 113 117 -0.027779 1k - 0k - - K O
cmr10 - 0k csr10 113 157 ec-lmr10 113 157 - 0k -0.027779 1k -27.778 -28 K o
cmr10 - 0k csr10 113 166 ec-lmr10 113 166 - 0k -0.083333 3k -83.333 -83 K v
cmr10 - 0k csr10 113 167 ec-lmr10 113 167 - 0k -0.027779 1k -27.778 -28 K w
cmr10 - - csr10 113 323 ec-lmr10 113 323 -0.027779 1k - 0k - - K Oacute
cmr10 - - csr10 113 324 ec-lmr10 113 324 -0.027779 1k - 0k - - K Ocircumflex
cmr10 - - csr10 113 326 ec-lmr10 113 326 -0.027779 1k - 0k - - K Odieresis
cmr10 - - csr10 113 363 ec-lmr10 113 363 - 0k -0.027779 1k -27.778 -28 K oacute
cmr10 - - csr10 113 364 ec-lmr10 113 364 - 0k -0.027779 1k -27.778 -28 K ocircumflex
cmr10 - - csr10 113 366 ec-lmr10 113 366 - 0k -0.027779 1k -27.778 -28 K odieresis
cmr10 - 0k csr10 113 033 ec-lmr10 113 367 - 0k -0.027779 1k -27.778 -28 K oe
cmr10 - 0k csr10 113 034 ec-lmr10 113 370 - 0k -0.027779 1k -27.778 -28 K oslash
cmr10 - 0k csr10 117 035 ec-lmr10 117 306 - 0k -0.027779 1k -27.778 -28 O AE
cmr10 -0.083334 3k csr10 120 054 ec-lmr10 120 054 -0.027779 1k -0.083333 3k -83.333 -83 P comma
cmr10 -0.083334 3k csr10 120 056 ec-lmr10 120 056 -0.027779 1k -0.083333 3k -83.333 -83 P period

```

CM	CS	EC=LM	cs	lm	afm	otf
cmr10 - 0k	csr10 150 126	ec-lmr10 150 126	- 0k	-0.083333 3k	-83.333	-83 h V
cmr10 - 0k	csr10 150 127	ec-lmr10 150 127	- 0k	-0.083333 3k	-83.333	-83 h W
cmr10 - 0k	csr10 153 126	ec-lmr10 153 126	- 0k	-0.083333 3k	-83.333	-83 k V
cmr10 - 0k	csr10 153 127	ec-lmr10 153 127	- 0k	-0.083333 3k	-83.333	-83 k W
cmr10 -0.055555 2k	csr10 153 141	ec-lmr10 153 141	-0.027779 1k	- 0k	-55.556	-56 k a
cmr10 - -	csr10 153 270	ec-lmr10 153 340	-0.027779 1k	- 0k	-55.556	-56 k agrave
cmr10 - -	csr10 153 341	ec-lmr10 153 341	-0.027779 1k	- 0k	-55.556	-56 k aacute

```

cmr10 - - csr10 153 344 ec-lmr10 153 344 -0.027779 1k - 0k - - k adieresis
cmr10 - 0k csr10 153 033 ec-lmr10 153 367 - 0k -0.027779 1k -27.778 -28 k oe
cmr10 - 0k csr10 153 034 ec-lmr10 153 370 - 0k -0.027779 1k -27.778 -28 k oslash
cmr10 - 0k csr10 155 126 ec-lmr10 155 126 - 0k -0.083333 3k -83.333 -83 m V
cmr10 - 0k csr10 155 127 ec-lmr10 155 127 - 0k -0.083333 3k -83.333 -83 m W
cmr10 - 0k csr10 156 126 ec-lmr10 156 126 - 0k -0.083333 3k -83.333 -83 n V
cmr10 - 0k csr10 156 127 ec-lmr10 156 127 - 0k -0.083333 3k -83.333 -83 n W
-----
cmr10 -0.055555 2k csr10 166 141 ec-lmr10 166 141 -0.027779 1k -0.055555 2k -55.556 -56 v a
cmr10 - - csr10 166 270 ec-lmr10 166 340 -0.027779 1k -0.055555 2k -55.556 -56 v agrave
cmr10 - - csr10 166 341 ec-lmr10 166 341 -0.027779 1k -0.055555 2k -55.556 -56 v aacute
cmr10 - - csr10 166 344 ec-lmr10 166 344 -0.027779 1k -0.055555 2k -55.556 -56 v adieresis
cmr10 - 0k csr10 166 032 ec-lmr10 166 346 - 0k -0.027779 1k -27.778 -28 v ae
cmr10 - 0k csr10 166 033 ec-lmr10 166 367 - 0k -0.027779 1k -27.778 -28 v oe
cmr10 - 0k csr10 166 034 ec-lmr10 166 370 - 0k -0.027779 1k -27.778 -28 v oslash
-----
CM          CS          EC-LM          cs          lm          afm      otf
-----

```

Ve verzi LM 1.106 byly kerningové páry "k":"a", *grave*, *acute* zrušeny i v položkách *afm* a *otf* na stav:

```

cmr10 -0.055555 2k csr10 153 141 ec-lmr10 153 141 -0.027779 1k - 0k - - k a
cmr10 - - csr10 153 270 ec-lmr10 153 340 -0.027779 1k - 0k - - k agrave
cmr10 - - csr10 153 341 ec-lmr10 153 341 -0.027779 1k - 0k - - k aacute

```

Druhá metoda je „praktická“. Výše uvedené textové tabulky sice soustřeďují nezbytná data o srovnání, ale pro posouzení jsou těžko použitelné. Proto na připravené seznamy znaků a kerningových párů aplikujeme přímo $\text{\TeX}$ (formálně $\text{\LaTeX}$ nebo $\text{\pdfLaTeX}$) a zadané kerningové páry dostaneme v grafická podobě. Kombinujeme varianty bez rámečků nebo s rámečky, kde je přímo vidět posun definovaný kernem, obdélníčky znázorňují relativní poměry velikostí kernů, ty poskytují vzájemné srovnání posunů ve dvou nebo více fontech.

Je asi nejnázornější rozdělit kerningové páry do podskupin a podle tabulek *automaticky* vygenerovat obrázky se *všemi* kerningovými páry, které jsou ve dvou fontech různé. Samozřejmě musíme je zúžit na podmnožinu znaků nacházejících se v obou fontech (v našem případě LM a $\mathcal{S}$). Bohužel, pro potřeby tohoto článku připravit příklady, které by demonstrovaly v dostatečné míře „většinu důležitých“ případů, v rozumném objemu, nezabíraly stovky stran jako úplný seznam, znamenalo větší množství ruční práce. Neboť právě jeden ze stanovených cílů práce bylo nejprve shromáždit a poté vydělit případy, kdy došlo ke změnám kernů oproti CM a $\mathcal{S}$, aby čtenář a potenciální budoucí uživatel LM-fontů mohl posoudit, co se změnilo, co bylo opraveno, vylepšeno, nebo i to, co se nepovedlo nejlépe.

Napřed předvedeme na vybraných příkladech, jak vypadají testovací tisky (obr. 9 a 10). V následujících případech porovnání kerningových párů se pro CM a ζ berou originální metriky, pro LM to jsou metriky *ec-lm* s kódováním T1. Tvary znaků ζ odpovídají bitovým mapám vygenerovaným z oficiální METAFONTové verze, zatímco glyfy CM zobrazuje jejich PostScriptová reprezentace, od níž se LM právě odvozují; tj. např. písmena v CM a LM až na několik málo výjimek v nejmenších detailech jsou definována totožnými obrysovými křivkami. A potenciální rozdílnost je pouze ve zde právě zkoumaných kerningových párech.

Kerningové páry můžeme rozdělit na následující podskupiny podle odlišnosti v LM, ζ a CM:

1. pár vyloučen z LM, v CM a ζ byl kern stejný
2. přidán do LM, v ζ (ani v CM) nebyl
3. v LM a CM je kern stejný, změnil se v ζ
4. v LM byl kern změněn, v CM nebyl nebo je shodný s ζ
5. v LM a CM není, je pouze v ζ
6. pár vyloučen z LM, přitom v ζ a CM jsou různé
7. v LM je velikost kernu dvojnásobná oproti ζ nebo CM (tento případ vydělujeme speciálně, jelikož je velice častý)

Uvedeme několik příkladů ilustrujících případy z jednotlivých podskupin.

Kerny společné pro CM a ζ :
`ligtable "K": "0" kern k#;`
 (pro všechny fonty) a
`ligtable "k": "a" kern k#;`
 (pro bezpatkové fonty).
 V LM byly oba kerningové páry zcela zrušeny.

cmr10 <■	cmss10 <■
csr10 <■	csss10 <■
ec-lmr10	ec-lmss10
cmbx10 <■	cmssi10 <■
csbx10 <■	csssi10 <■
ec-lmbx10	ec-lmss10
cmsl10 <■
cssl10 <■
ec-lmro10

cmr10 <■
csr10 <■
ec-lmr10
cmbx10 <■
csbx10 <■
ec-lmbx10
cmsl10 <■
cssl10 <■
ec-lmro10

V CM je definováno pro patkové typy takto:

`ligtable "k": if serifs: "v": "a" kern -u#, fi`
 v ζ byl kern zmenšen; LM tento kerningový pár ruší. Příliš malý odstup mezi "k" a "a" v Computer Modern označil B. Jackowski na své přednášce na EuroTeXu 2005 za přítomnosti D. Knutha (přeformulováno mými slovy) za „vlastnost“ CM-fontů, ale jelikož ve svých LM-fontech by to považovali za chybu, v LM „tento bug opravili“.

KOKO cmb10 <■	KOKO cmr7 <■
KOKO csb10 <■	KOKO csr7 <■
KOKO ec-lmb10	KOKO ec-lmr7
KOKO cmbx10 <■	KOKO cmr8 <■
KOKO csbx10 <■	KOKO csr8 <■
KOKO ec-lmbx10	KOKO ec-lmr8
KOKO cmbx12 <■	KOKO cmr9 <■
KOKO csbx12 <■	KOKO csr9 <■
KOKO ec-lmbx12	KOKO ec-lmr9
KOKO cmbx5 <■	KOKO cmsl10 <■
KOKO csbx5 <■	KOKO cssl10 <■
KOKO ec-lmbx5	KOKO ec-lmr10
KOKO cmbx6 <■	KOKO cmsl12 <■
KOKO csbx6 <■	KOKO cssl12 <■
KOKO ec-lmbx6	KOKO ec-lmr12
KOKO cmbx7 <■	KOKO cmsl8 <■
KOKO csbx7 <■	KOKO cssl8 <■
KOKO ec-lmbx7	KOKO ec-lmr8
KOKO cmbx8 <■	KOKO cmsl9 <■
KOKO csbx8 <■	KOKO cssl9 <■
KOKO ec-lmbx8	KOKO ec-lmr9
KOKO cmbx9 <■	KOKO cmss10 <■
KOKO csbx9 <■	KOKO csss10 <■
KOKO ec-lmbx9	KOKO ec-lmss10
KOKO cmbxsl10 <■	kaka cmss10 <■
KOKO csbxsl10 <■	kaka csss10 <■
KOKO ec-lmbxo10	kaka ec-lmss10
KOKO cmdunh10 <■	KOKO cmss12 <■
KOKO csdunh10 <■	KOKO csss12 <■
KOKO ec-lmdunh10	KOKO ec-lmss12
KOKO cmr10 <■	kaka cmss12 <■
KOKO csr10 <■	kaka csss12 <■
KOKO ec-lmr10	kaka ec-lmss12
KOKO cmr12 <■	KOKO cmss17 <■
KOKO csr12 <■	KOKO csss17 <■
KOKO ec-lmr12	KOKO ec-lmss17
KOKO cmr17 <■	kaka cmss17 <■
KOKO csr17 <■	kaka csss17 <■
KOKO ec-lmr17	kaka ec-lmss17
KOKO cmr5 <■	KOKO cmss8 <■
KOKO csr5 <■	KOKO csss8 <■
KOKO ec-lmr5	KOKO ec-lmss8
KOKO cmr6 <■	kaka cmss8 <■
KOKO csr6 <■	kaka csss8 <■
KOKO ec-lmr6	kaka ec-lmss8

Obrázek 9: Ukázka jedné strany testovacího protokolu (ve zmenšení)

ka cmb10 <■	ka cmr17 <■
ka csb10 <■	ka csr17 <■
ka ec-lmb10	ka ec-lmr17
ka cmbx10 <■	ka cmr5 <■
ka csbx10 <■	ka csr5 <■
ka ec-lmbx10	ka ec-lmr5
ka cmbx12 <■	ka cmr6 <■
ka csbx12 <■	ka csr6 <■
ka ec-lmbx12	ka ec-lmr6
ka cmbx5 <■	ka cmr7 <■
ka csbx5 <■	ka csr7 <■
ka ec-lmbx5	ka ec-lmr7
ka cmbx6 <■	ka cmr8 <■
ka csbx6 <■	ka csr8 <■
ka ec-lmbx6	ka ec-lmr8
ka cmbx7 <■	ka cmr9 <■
ka csbx7 <■	ka csr9 <■
ka ec-lmbx7	ka ec-lmr9
ka cmbx8 <■	ka cmsl10 <■
ka csbx8 <■	ka cssl10 <■
ka ec-lmbx8	ka ec-lmro10
ka cmbx9 <■	ka cmsl12 <■
ka csbx9 <■	ka cssl12 <■
ka ec-lmbx9	ka ec-lmro12
ka cmbxsl10 <■	ka cmsl8 <■
ka csbxsl10 <■	ka cssl8 <■
ka ec-lmbxo10	ka ec-lmro8
ka cmdunh10 <■	ka cmsl9 <■
ka csdunh10 <■	ka cssl9 <■
ka ec-lmdunh10	ka ec-lmro9
ka cmr10 <■	ka cmvtt10 <■
ka csr10 <■	ka csvtt10 <■
ka ec-lmr10	ka ec-lmvtt10
ka cmr12 <■	
ka csr12 <■	
ka ec-lmr12	

Obrázek 10: Další část testovacího protokolu ukazuje část přehledu případů, kdy se kerny v CM, $\mathcal{C}\mathcal{S}$ a LM (v kódování T1) ve všech navzájem liší. Jak vidíme, vesměs se týkají patkových fontů a páru "k": "a"

$\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ cmr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■
 $\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ cmr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■
 $\mathbb{V}_\mathbb{V}$ csr10 <■
 $\mathbb{V}_\mathbb{V}$ cmr10 <■
 $\mathbb{V}_\mathbb{V}$ ec-lmr10 <■

$\mathbb{H}_\mathbb{H}$ csti10
 $\mathbb{H}_\mathbb{H}$ cmti10 <■
 $\mathbb{H}_\mathbb{H}$ ec-lmri10 <■
 $\mathbb{G}_\mathbb{G}$ csti10
 $\mathbb{G}_\mathbb{G}$ cmti10 <■
 $\mathbb{G}_\mathbb{G}$ ec-lmri10 <■
 $\mathbb{H}_\mathbb{H}$ csti10
 $\mathbb{H}_\mathbb{H}$ cmti10 <■
 $\mathbb{H}_\mathbb{H}$ ec-lmri10 <■
 $\mathbb{G}_\mathbb{G}$ csti10
 $\mathbb{G}_\mathbb{G}$ cmti10 <■
 $\mathbb{G}_\mathbb{G}$ ec-lmri10 <■

$\mathbb{P}_\mathbb{P}$ cmr10 ■>
 $\mathbb{P}_\mathbb{P}$ csr10 ■>
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 ■>
 $\mathbb{P}_\mathbb{P}$ cmr10 ■>
 $\mathbb{P}_\mathbb{P}$ csr10 ■>
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 ■>

$\mathbb{F}_\mathbb{F}$ cmr10
 $\mathbb{F}_\mathbb{F}$ csr10 <■
 $\mathbb{F}_\mathbb{F}$ ec-lmr10
 $\mathbb{F}_\mathbb{F}$ cmr10
 $\mathbb{F}_\mathbb{F}$ csr10 <■
 $\mathbb{F}_\mathbb{F}$ ec-lmr10

Kerny se změnilý v $\mathcal{C}_S$, v LM se zachovává kompatibilita s CM, kde jsou definovány kerningové páry "P":",", ". "; tedy právě jen mezi písmenem „P“ následovaným čárkou nebo tečkou, ale pro žádná jiná písmena; údajně se v CM šetřilo paměti pro metrická data.

$\mathbb{H}_\mathbb{H}$ csti10
 $\mathbb{H}_\mathbb{H}$ cmti10 <■
 $\mathbb{H}_\mathbb{H}$ ec-lmri10 <■
 $\mathbb{H}_\mathbb{H}$ csti10
 $\mathbb{H}_\mathbb{H}$ cmti10 <■
 $\mathbb{H}_\mathbb{H}$ ec-lmri10 <■
 $\mathbb{H}_\mathbb{H}$ csti10
 $\mathbb{H}_\mathbb{H}$ cmti10 <■
 $\mathbb{H}_\mathbb{H}$ ec-lmri10 <■

$\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■
 $\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■
 $\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■
 $\mathbb{P}_\mathbb{P}$ csr10 <■
 $\mathbb{P}_\mathbb{P}$ ec-lmr10 <■

$\mathbb{F}_\mathbb{F}$ cmr10
 $\mathbb{F}_\mathbb{F}$ csr10 ■>
 $\mathbb{F}_\mathbb{F}$ ec-lmr10

Pouze $\mathcal{C}_S$ -fonty ruší kerny v kurzívě mezi některými verzálkami, nedochází ke zmenšení mezery za písmeny „b“, „c“, „o“, „p“. Vzdálenost mezi písmeny v LM zůstává stejná, jako byla v CM.

V LM se zmenšuje odstup vykřičníku a otazníku za apostrofem, což je účelné zejména v OpenType. Naopak, některá akcentovaná písmena budou v LM o něco dál od „T“, než je tomu v $\mathcal{C}_S$.

V $\mathcal{C}_S$ byly zavedeny nové páry jednostranně pro "F", "T", "V", "W", "Y":",", ". "; LM ponechávají stav podle CM.

	$\overline{A}A\overline{A}A$ <i>cmti10</i> <■	$\overline{A}A\overline{A}A$ <i>csti10</i> <■
$\overline{A}C\overline{A}C$ <i>cmti10</i> <■	$\overline{A}A\overline{A}A$ <i>ec-lmri10</i> <■	
$\overline{A}C\overline{A}C$ <i>csti10</i> <■	$\overline{A}m\overline{A}m$ <i>cmti10</i> <■	
$\overline{A}C\overline{A}C$ <i>ec-lmri10</i> <■	$\overline{A}m\overline{A}m$ <i>csti10</i> <■	
$\overline{A}G\overline{A}G$ <i>cmti10</i> <■	$\overline{A}m\overline{A}m$ <i>ec-lmri10</i> <■	
$\overline{A}G\overline{A}G$ <i>csti10</i> <■	$\overline{A}m\overline{A}m$ <i>cmti10</i> <■	
$\overline{A}G\overline{A}G$ <i>ec-lmri10</i> <■	$\overline{A}m\overline{A}m$ <i>csti10</i> <■	
$\overline{A}O\overline{A}O$ <i>cmti10</i> <■	$\overline{A}m\overline{A}m$ <i>ec-lmri10</i> <■	
$\overline{A}O\overline{A}O$ <i>csti10</i> <■	$\overline{A}m\overline{A}m$ <i>cmti10</i> <■	
$\overline{A}O\overline{A}O$ <i>ec-lmri10</i> <■	$\overline{A}m\overline{A}m$ <i>csti10</i> <■	
$\overline{A}Q\overline{A}Q$ <i>cmti10</i> <■	$\overline{A}m\overline{A}m$ <i>ec-lmri10</i> <■	
$\overline{A}Q\overline{A}Q$ <i>csti10</i> <■	$\overline{A}M\overline{A}M$ <i>cmti10</i> <■	
$\overline{A}Q\overline{A}Q$ <i>ec-lmri10</i> <■	$\overline{A}M\overline{A}M$ <i>csti10</i> <■	
$\overline{A}U\overline{A}U$ <i>cmti10</i> <■	$\overline{A}M\overline{A}M$ <i>ec-lmri10</i> <■	
$\overline{A}U\overline{A}U$ <i>csti10</i> <■	$\overline{A}m\overline{A}m$ <i>cmti10</i> <■	
$\overline{A}U\overline{A}U$ <i>ec-lmri10</i> <■	$\overline{A}m\overline{A}m$ <i>csti10</i> <■	
	$\overline{A}m\overline{A}m$ <i>ec-lmri10</i> <■	

LM-fonty upravují vzdálenosti za kurzívními kapitálkami. Konkrétně (záporné) kerny se zdvojnásobují a písmena následující za „A“ a „K“ budou o něco blíž.

Následující případy ukazují řadu příkladů kerningových párů, které byly nově přidány do metrik LM-fontů.

$\overline{L}L$ <i>cmr10</i>	$\overline{F}F$ <i>cmr10</i>	$\overline{X}X$ <i>csr10</i>
$\overline{L}L$ <i>csr10</i>	$\overline{F}F$ <i>csr10</i>	$\overline{X}X$ <i>ec-lmr10</i> <■
$\overline{L}L$ <i>ec-lmr10</i> <■	$\overline{F}F$ <i>ec-lmr10</i> ■>	$\overline{X}X$ <i>csr10</i>
$\overline{L}L$ <i>cmr10</i>	$\overline{F}F$ <i>cmr10</i>	$\overline{X}X$ <i>ec-lmr10</i> <■
$\overline{L}L$ <i>csr10</i>	$\overline{F}F$ <i>csr10</i>	
$\overline{L}L$ <i>ec-lmr10</i> <■	$\overline{F}F$ <i>ec-lmr10</i> ■>	
$\overline{L}L$ <i>cmr10</i>		
$\overline{L}L$ <i>csr10</i>		
$\overline{L}L$ <i>ec-lmr10</i> <■

cmr10		cmr10
csr10		csr10
<	csr10	<

cmr10	ec-lmr10 <	cmr10
csr10	csr10	csr10
<	<	<

cmr10	csr10	cmr10
csr10	ec-lmr10 <	csr10
<	csr10	<

cmr10	ec-lmr10 <	cmr10
csr10	ec-lmr10 <	csr10
<	ec-lmr10 <	<

cmr10	cmr10	cmr10
csr10	csr10	csr10
<	<	<

cmr10	cmr10
csr10	csr10
<	<	cmr10
cmr10	cmr10	csr10
csr10	csr10	<

<	<	cmr10
		csr10
	cmr10	<

<	cmr10
	cmr10	csr10
	<	<

cmr10
		csr10
		<

Upravují se vzdálenosti apostrofů a uvozovek. Také se mění odstupy za a před některými písmeny, např. za „A“ a „K“, před „V“ a „W“. Samozřejmě zde demonstují jen některé vybrané příklady.

Poté ukážeme výsledek vyhodnocení a dalšího ručního výběru – výtah ze srovnávacího přehledu změn ve zhuštěné formě. Čtenáři by to mělo umožnit posouzení vlivu kernů na „optickou vyváženost“ mezer mezi písmeny v příslušných metrikách jednotlivých fontů, většinou z řezu „Roman Regular“.

1.1 CM: KOKOKOKOKOkaka
 CS: KOKOKOKOKOkaka
 LM: KOKOKOKOKOkaka

6.1 CM: kakaka
 CS: kakaka
 LM: kakaka

3.1 CS: vavavaP,P,P,P,P,P,P,P.
 CM: vavavaP,P,P,P,P,P,P,P.
 LM: vavavaP,P,P,P,P,P,P,P.

3.2 CS: vāvāvā
 LM: vāvāvā

3.3 CS: babcbdbegbqbqacccdcgcgcq
 CM: babcbdbegbqbqacccdcgcgcq
 LM: babcbdbegbqbqacccdcgcgcq

3.4 CS: oaocodoegoqoqpapcpdpepgpopq
 CM: oaocodoegoqoqpapcpdpepgpopq
 LM: oaocodoegoqoqpapcpdpepgpopq

4.1 CM: ff!ff'ff)ff?ff]'!?'f!f'f)f?f]ajbjojpj
 CS: ff!ff'ff)ff?ff]'!?'f!f'f)f?f]ajbjojpj
 LM: ff!ff'ff)ff?ff]'!?'f!f'f)f?f]ajbjojpj

4.2 CS: FcFëFýFûFàFáFäFéFóFöFúFü
 LM: FcFëFýFûFàFáFäFéFóFöFúFü

4.3 CS: TčTěTůTýVřVůVàVáVäVéVóVöVúLTLV
 LM: TčTěTůTýVřVůVàVáVäVéVóVöVúLTLV

5.1 CS: F,F.T,T.V,V.W,W.Y,Y...
 CM: F,F.T,T.V,V.W,W.Y,Y...
 LM: F,F.T,T.V,V.W,W.Y,Y...

5.2 CS: KÓKÔKÖKàkákäbüôûpúóûôüöü
 LM: KÓKÔKÖKàkákäbüôûpúóûôüöü

1.1 V LM byly kerny vynechány, proto se vzdálenosti pro "K": "O" a bezpatkové "k": "a" mírně zvětšily.

6.1 Také patkové "k": "a" je v LM bez kernu; pro vzdálenosti platí $CM < \zeta < LM$.

3.1 ζ kern mění a vzdálenost zvětšuje, LM zůstává shodné s CM.

3.2 Páry kopírují situaci s neakcentovanými „a“.

3.3 ζ produkují sazbu kurzívy „více proloženou“.

3.4 Navíc se mi jeví, že bitmapová reprezentace ζ vypadá tučnější než vyrastrovaný obrysový font, patrně algoritmy METAFONTového a PostScriptového „hintování“ skloněných typů fungují různě.

Ve skupině 4 jsou případy, kdy v ζ (resp. CM) byly kerningové páry přítomny, ale v LM jsou hodnoty kernů odlišné. Naopak skupina 5 obsahuje situace, kdy nenulové kerny jsou pouze v metrikách ζ a žádných jiných.

Skupina 7 zahrnuje případy, ve kterých záporný kern byl v LM zvětšen na dvojnásobek. Skupina 2 ukazuje přehled vybraných příkladů, kdy LM zavádějí nové kerningové páry, které v ostatních dřívějších fontech dosud nebyly.

9. Co nebylo v zadání grantu: Čím se liší LM od ζ

Jak už bylo popsáno dříve v sekci o zkoumání čs. akcentovaných písmen, nejprve byly provedeny vizuální prověrky akcentovaných písmen samostatně pro LM pro subset znaků vymezený kódováním T1, tj. byla samostatně analyzována vybraná písmena s diakritikou ve všech LM-fontech, aby zahrnovala podmnožinu znaků vyskytujících se v ζ -fontech. Především byly zkoumány akcenty, jednotnost jejich tvaru a usazení, a to i v textových řezech LM-fontů, které v CM a ζ nejsou. Na druhé straně, LM nepodporují rozšířené spektrum stupňů, jak to dělají ζ -fonty v METAFONTových zdrojích. Autoři LM též vynechali některé obskurní a pravděpodobně zřídka používané fonty jako `cmff10`, `cmfi10` a `cmfib8`, jejichž návrh v METAFONTu já osobně nepovažuji za dotažený do konce.

Pouhé prohlížení tvaru všech znaků v LM-fontech však nedalo dostatečnou informaci potřebnou k vyhodnocení tvaru akcentů a jejich usazení.

V zadání nebylo přímé porovnání glyfů LM-fontů s oficiálními referenčními bitmapovými reprezentacemi ζ -fontů. Ani k tomu nebyla k dispozici sada potřebných nástrojů, abychom mohli dostat úplnější obraz o rozdílech v akcentech. S výjimkou omezeného počtu částečných namátkových testů nebylo provedeno regulérní srovnání. Procedury, programy a makra pro porovnání LM s ζ -fonty jsem dotvořil až v roce 2007 a potom zkoumal speciálně vybranou zúženou podmnožinu znaků společnou pro oba fonty. Tedy materiály k dostatečně širokému porovnání byly vygenerovány a následně analyzovány až dodatečně. Situaci komplikuje to, že srovnáváme font obrysový s fontem bitmapovým. Testování bitmapových fontů může přitom být podle složitosti METAFONTových zdrojů i několikrát pracnější (tj. dražší) než prověřování fontů obrysových.

Následující ukázky (obr. 11) představují případy vybrané pro zkoumání a po porovnání výsledků, převážně aby ilustrovaly již dříve nalezené potenciální rozdíly. ζ -fonty jsou zde reprezentovány vyraastrovanými bitmapovými obrazy, zatímco obrysové křivky představují kontury LM. Úzké horizontální pruhy umístěné nad základní dotažnicí vyznačují přibližné jednotné zvolené „optické výšky“ akcentů, nepředstavují přímo některé z akcentových dotažnic ani hintovacích zón, slouží jen k vzájemnému porovnání výšek umístění jednotlivých akcentů.

Při vyhodnocování rozdílů, zejména v horizontálním umístění akcentů u vybraných písmen, jsem si vzal na pomoc i fonty Lido Františka Štorma. K mému překvapení se ukázalo, když jsem poslední prověrky prováděl už potřetí, tentokrát s verzí LM 1.010, že ve většině nejspornějších případů, tam, kde se LM liší od ζ , je LM bližší Lido. Týká se to hlavně „é“, kde v Lido je čárka umístěna více vlevo, podobně jako v LM. Analogická situace je i v dalších případech, jako třeba „Č“. Zanechal jsem proto předchozí experimenty, v nichž jsem se pokoušel hýbat akcentem do stran o vzdálenost `u#` nebo `0.5u#`. Po opakovaných vyhodnoceních a konzultacích jsem usoudil, že zbyl jediný případ „Ř“, který by měl být upraven.

Obrázek 11: Ukázky rozdílů u vybraných znaků z fontů csbx10 a lmbx10

9.1. Testování verzí z roku 2007

Verze LM 1.010 již nebyla v plánu dohodnutých prověrek, navíc obsahuje několik novinek, k jejichž prověrkám nemám připraven testovací aparát. Protože nedošlo k rozhodnutí zahrnout navíc tyto práce, nebyly do programu činnosti zařazeny. Týká se to metrik `cs-lm`, jak bylo řečeno i na jiném místě, nebo změn jmen glyfů v OpenType.

Protože však bylo žádoucí vyhodnotit výsledky testů, opravit v nich různé možné chyby (kontrolor verifikující určitý produkt musí prověřovat i svou kontrolní práci), a také potvrdit, že byly opravdu opraveny chyby objevené a ohlášené dříve nebo odhalené jako nové, byly testy a srovnání metrik (v kódování T1) i glyfů aktualizovány, znovu vyhodnoceny a zkontrolovány pro verze v roce 2007. I když od vydání verze 1.010, která je poprvé na `TEXLive`, nezbývalo (jako vždy) mnoho času, bylo nezbytné z prověrek provést sice omezenou ale jistou nejdůležitější část pro prezentaci na EuroBach`TEXu` 2007.

Je také možné, že tak, jak se postupně během posledních let LM-fonty průběžně měnily, mohlo zůstat neopraveno i to, co už bylo ohlášeno v roce 2005, tehdejší hlášení už „nepasovalo“ na další verze, čili bez přeprůvěření každé nejnovější verze nelze vyloučit, že některé nedostatky dosud nezmizely.

9.2. Jiné testy

Krátce mohu shrnout i výsledky některých dalších testů, které se záležitostmi spojenými s čs. sazbou nebo podobných důležitých věcí přímo netýkaly.

Provedl jsem srovnání glyfů mezi verzemi z let 2006 a 2007. Vygeneroval jsem příslušné testovací tisky s rozdíly mezi verzemi LM 1.00 a LM 1.010. Změny jsem jen zběžně prohlédl, ale vesměs se netýkaly písmen české nebo slovenské abecedy.

Dlouhodobě se vyskytují i některé nepatrné artefakty v obrysových křivkách. Mám, ale nebudu zde uvádět příklady „mikrotypografických“ defektů, které jsou vidět jen při velkém zvětšení v PDF, ale na reálně vyrašťovaný tisk vliv nemají.

V distribuci zůstávají i některé, podle mého názoru „nepovedené“ fonty. Jedná se obvykle o fonty málo používané, jako např. `lmb10`, kde poměrně často nevycházejí dobře mezery mezi písmeny, jelikož jsou ale definice kerningů sdíleny s ostatními textovými fonty, odlišné řešení jednoho fontu by působilo nežádoucí komplikace. Rovněž fonty `lmu10` a `lmdunh10/lmduno10` jsou dost netypické. Proto nepovažuji vkládat další úsilí do těchto fontů za opodstatněnou a rozumnou investici. Jejich příklady zde také vynecháme.

9.3. Neplánovaná odhalení

Neplánovaným a neočekávaným bočním efektem srovnání LM-fontů s CM-fonty a $\mathcal{S}$ -fonty bylo odhalení několika chyb v $\mathcal{S}$ i CM-fontech, v `METAFONTu` i v Type 1. Překvapivé je, že přetrvávají už 10, 15 nebo i více let.

Předpokládám, že si pozorný čtenář v úvodní části článku všiml, že příklady „Á“ (`csbx10`) a „a“ (`cmr10`) ukazují, že není něco v pořádku v METAFONTových definicích příslušných znaků, když se s rozlišením může měnit tvar či umístění glyfu nebo jeho části.

V tučných řezech oficiální verze $\mathcal{G}$ -fontů byly vždy nesprávně čárky nad všemi velkými písmeny, s růstem rozlišení se čárka zužuje.

Na rozdíl od „a“, „c“, „æ“ v Computer Modern Regular (např. `cmr10`) po vygenerování bitových map METAFONTem v rozlišeních 1200 a 2400 dpi se můžeme podívat, když si pořídíme výtisky ve správné reálné velikosti a v obou rozlišeních na odpovídající tiskárně a prohlédneme si je třeba pod lupou. Prof. D. Knuth nevidí chybu, která by primárně závisela na rozlišení zařízení, ale rozdíly rastrování způsobuje parametr zařízení „blackier“.

V PostScriptové verzi `CMMIB5` dosud zůstala neopravená „7“, byla napřed převzata do `lmmib5` ver. 1.00, ve verzi 1.010 byla po mém hlášení opravena (obr. 12).

10. Použití LM-fontů pro sazbu češtiny a slovenštiny

K tomuto tématu zde zmíním jen pár krátkých poznámek, jelikož považuji za vhodnější, aby se vypracování uživatelského manuálu ujal odborník povolanejší než já.

Český a slovenský text můžeme L^AT_EXem vysázet LM-fonty v zásadě třemi základními způsoby, kdy použijeme

1. metriky $\mathcal{G}$ -fontů v kódování IL2 z $\mathcal{G}$ L^AT_EXu
2. metriky `ec-lm` v kódování T1 (Cork) z distribuce LM
3. metriky `cs-lm` v kódování IL2 z distribuce LM

Následný `dvips`, `pdfLATEX` nebo jiný ovladač do výstupních souborů v PS nebo PDF dosadí pomocí příslušných mapování fontů a znaků odpovídající glyfy z kolekce LM.

Poslední sada metrik s označením `cs-lm` se poprvé objevila až v distribuci LM 1.010. Zkoumání a prověřování novinek z roku 2007 už nepatří mezi zadání této mé práce.

11. Budoucnost $\mathcal{G}$ -fontů

Nedokážu odhadnout vývoj fontových technologií na desetiletí dopředu. Proto vyslovím svůj názor jen na blízká léta. Současným standardem koncových tiskových dokumentů je PDF, který obsahuje fonty ve formátu OpenType (popřípadě Type 1) nezávislém na rozlišení výstupních zařízení. Musíme sice opakovaně připomenout, že formát OpenType povoluje pouze celočíselné hodnoty metrických

Neopravená verze "sevenoldstyle"
v CMMIB5 a zároveň v lmmib5
(ver. 1.00).

Opravená a přejmenovaná
"seven_taboldstyle"
v lmmib5 (ver. 1.010).

Porovnání obou alternativ,
nevyplněné obrysové křivky
odpovídají stavu v PostScriptové
verzi CM, tzn. současně
i v LM před opravou.

Obrázek 12: Oprava Math Italic Bold „7“

a dalších numerických údajů. To by samozřejmě bránilo kompatibilitě metrik s jejich originály v `tfm`. Přesto jsem názoru, že se pro $\mathcal{G}$ TUG se v podstatě nabízejí dvě zásadní možnosti: neplánovat už žádnou podporu dalšího vývoje $\mathcal{G}$ -fontů, nebo vytvořit novou verzi $\mathcal{G}$ -fontů právě ve formátu OpenType. K tomu musím poznamenat, že já sám k tomuto úkolu nemám v současné době dost nástrojů a zkušeností, abych to dokázal udělat správně, efektivně a rychle, tj. bez chyb a v rozumném časovém termínu. Nicméně, má-li někdo jiný názor a myslí si, že tvorba rozsáhlých a přitom velmi kvalitních OpenType fontů není pracná a drahá záležitost, mohl by v případě poptávky nabídnout své služby a prokázat své schopnosti a umění. Velice rád bych viděl úspěšné výsledky jeho snažení.

12. Nástroje pro testování a srovnávání fontů

Z programů pro srovnávání a jiné prověrky fontů byl sestaven autonomní balík, který zahrnuje zjednodušený aparát testovacích nástrojů a dovoluje jejich použití prostřednictvím samostatného volání procedur, většinou Linuxových skriptů.

Tyto nástroje jsou (relativně) primitivní, většinou poskytují jen omezené možnosti použití. Přesto mohou obsahovat mnoho chyb i systémových závislostí. Nicméně doufám, že obsahují pár původních myšlenek, které mně osobně pomohly alespoň zrychlit a doplnit testovací procesy. Jako volný produkt jsou zveřejněny na mé stránce <http://www-hep.fzu.cz/~piska/tfcp.html> a každý zájemce má právo je nebo jejich části používat, opravovat nebo upravovat podle svého přání a uvážení. Podle mého názoru by stálo za to některé programy nebo jejich části přepsat do LuaTeXu.

13. Prezentace výsledků a přímé kontakty s autory

Na konferencích EuroTeX, TUG a GUST bylo předneseno několik samostatných příspěvků. V rámci setkání, obvykle jako součást prezentací, byla i aktuální hlášení o chybách, návrhy a jiné komentáře k současnému vydání fontů. Většinou byly předány nové výsledky autorům osobně, některé z nich vytvořené či vypočtené přímo na místě. Všechny příspěvky prezentované v letech 2006–2007 v nějaké podobě obsahují informace o LM, CM, $\mathcal{G}$ a jiných fontech, o testování fontů nebo o MetaType1. Většina z nich byla doprovázena vytištěnými preprinty v konferenčních sbornících, ty byly po změnách, opravách a úpravách redaktorů a autora následně připraveny k publikování v časopise TUGboat.

Tento článek je svým obsahem zároveň výběrem i doplňkem předchozích dokumentů, přednášek a publikací, které svou šíří přesahují samotné téma prověřování LM-fontů, neboť se mohou zabývat také tvorbou a testováním fontů všeobecně.

Tyto dokumenty byly postupně zveřejňovány na webových stránkách:

- <http://www-hep.fzu.cz/~piska/lm2005.html>
- <http://www-hep.fzu.cz/~piska/lm2006.html>
- http://www-hep.fzu.cz/~piska/cmr_a.html
- <http://www-hep.fzu.cz/~piska/tfcpr.html>

Výsledky testů byly též:

- prezentovány na setkáních TUG a LUG:
Obě přednášky na konferencích BachoT_EX 2006 a EuroT_EX 2006 (o srovnávání fontů v příkladech) zahrnovaly nejnovější výsledky prověrek poslední verze LM-fontů. Analogické aktualizované informace obsahovaly také dva příspěvky pro EuroBachoT_EX 2007: o testovacích nástrojích a celkový shrnující komentář a návrhy k fontům Latin Modern.
- vytištěny v konferenčních sbornících:
Font verification and comparison in examples, *Proceedings of the XVI EuroT_EX Conference*, pp. 98–103, 5–8 July 2006, Debrecen, Hungary, 2006. Comments and suggestions about the Latin Modern fonts, pp. 111–117; Procedures for font comparison, pp. 104–110, *Proceedings of the XVII European T_EX 2007*, 28 April–2 May 2007, Bachotek, Poland, 2007.
- publikovány v časopise TUGboat:
 - TUGboat 27:1, pp. 71–75, 2006.
 - TUGboat 29:1, pp. 50–56, 2008.
 - TUGboat 29:1, pp. 57–65, 2008.

Reference

- [1] Donald E. Knuth. *The METAFONTbook*. Addison-Wesley, 1986. Volume C of *Computers and Typesetting*.
- [2] Donald E. Knuth. *Computer Modern Typefaces*. Addison-Wesley, 1986. Volume E of *Computers and Typesetting*.
- [3] Computer Modern fonts. CTAN:/fonts/cm.
- [4] CS fonts. <ftp://math.feld.cvut.cz/pub/cstex/base/csfonds.tar.gz>.
- [5] EC fonts. CTAN:/fonts/ec.
- [6] Latin Modern fonts. CTAN:/fonts/lm.
- [7] Bogusław Jackowski, Janusz M. Nowacki, Piotr Strzelczyk. Programming PostScript Type 1 Fonts Using MetaType1: Auditing, Enhancing, Creating. *Proceedings of EuroT_EX 2003*, Brest, France, 24–27 June 2003. TUGboat 24:3, pp. 575–581.
- [8] Bogusław Jackowski, Janusz M. Nowacki. Enhancing Computer Modern with accents, accents, accents. TUGboat 24:1, 2003.
- [9] Bogusław Jackowski, Janusz M. Nowacki. Latin Modern fonts: How less means more. *Proceedings of the XV EuroT_EX 2005 conference*, Pont-à-Mousson, France, March 7–11, 2005.

- [10] Bogusław Jackowski, Janusz M. Nowacki. Rodzina fontów Latin Modern. *Biuletyn Polskiej Grupy Użytkowników Systemu T_EX*, Zeszyt 23:9–12, 2006.
- [11] Jerzy Ludwichowski, Bogusław Jackowski, and Janusz Nowacki. Five years after: Report on international T_EX font projects. *TUGboat* 29:1, pp. 25–26, 2008.
- [12] Will Robertson. An exploration of the Latin Modern fonts. *TUGboat* 28:2, pp. 177–180, 2007.

Poděkování

Děkuji ζ TUGu za návrh grantu, zájem, náměty a další podněty pro provedení prací spojených s testováním LM a fontů jim příbuzným, za částečnou finanční podporu mé účasti na konferencích EuroT_EX a za možnost otisknutí tohoto článku. Děkuji také redaktorovi Zpravodaje Zdeňku Wagnerovi za profesionální přístup a mimořádnou péči a úsilí při zpracování a produkci této publikace. Jemu rovněž děkuji, že celý text článku, s výjimkou explicitních příkladů vložených většinou jako externí dokumenty v PDF, vysázel fonty Latin Modern v kódování T1 s metrikami `ec-lm`. Čtenář tak může přímo posoudit výsledek tohoto způsobu formátování včetně tvarů znaků a kerningů.

Závěr

Byla podrobně prověřována metrická a grafická data několika posledních verzí fontů Latin Modern, paralelně byly zveřejňovány výsledky prověrek v textové a původní obrazové podobě, oznamována hlášení o nalezených chybách, podávány návrhy změn a jiné komentáře. Vedle toho byly prováděny testy ζ -fontů, CM-fontů a metrických údajů EC-fontů.

Ve všech výše uvedených fontech, jejich METAFONTových, bitmapových i obrysových reprezentacích byla objevena řada závažných chyb nebo významných vzájemných nekompatibilit, některé z nich byly zveřejněny úplně poprvé. Doufám, že tento vklad přispěl k důležitým vylepšením a k odstranění dlhých nedostatků, v jednotlivostech vybočujících z jinak celkově vysoké kvality kolekce LM-fontů, zejména té části, která podporuje sazbu českých a slovenských textů, a tím spolunapomohl řešení úkolů, které si před sebe klade ζ TUG.

Nechápal jsem, že by hlavním cílem práce byl právě tento článek, ale to, že prověrky LM-fontů v rozsahu provedeného testování přispěly k úspěšnému splnění toho nejdůležitějšího výstupu pro ζ TUG — zařazení LM-fontů na T_EXLive — aby byly pro československé uživatele T_EXu (a zejména L^AT_EXu v kódování T1) k dispozici včas, v dostatečné kvalitě a s co nejmenším počtem chyb, spojených s českou a slovenskou sazbou. Nicméně, vývoj fontů se nezastavil, a proto by sledování novinek a změn ze strany ζ TUGu a jeho členů mělo i nadále pokračovat.

Summary: Latin Modern fonts testing with regard to the Czechoslovak typesetting requirements

This extended article presents grant outputs with the authors' major findings and results. It makes recommendations for changes to the LM-font creators after performing comparisons of fonts such as CM, LM, $\mathcal{C}\mathcal{S}$ and EC in the Type 1 format. The article also makes comparisons based on metric and graphical data. The tested parameters were the widths of the letters, kernings, differences in LM, $\mathcal{C}\mathcal{S}$ and CM fonts, and finally the technical quality of the glyphs.

The tools used during the testing were **FontForge** and also **MetaType1**. The testing scripts were done in bash and AWK and are published on the author's websites. The authors proposed some changes to improve the actual state and they commented on this in depth and followed it up with illustrations and tables.

They also presented their thoughts on the creation of a new OpenType font and rewriting testing scripts in Lua $\TeX$. Some findings were presented at Bacho $\TeX$ 2006, Euro $\TeX$ 2006 and the EuroBacho $\TeX$ 2007 conferences and their proceedings have been published.

1. Úvod

V následujících odstavcích se s Vámi chceme podělit o naše zkušenosti, postřehy a některé problémy, jimž jsme čelili při přípravě posterů v T_EXu, konkrétně s využitím třídy `a0poster`.

2. A0 poster

Následující odstavce jsou věnovány tvorbě (T_EXování) posteru s využitím třídy `a0poster`. Dovolte nám však, zmínit nejprve pár obecných zásad pro přípravu přehledného posteru:

- Zvolte si jedno hlavní téma, jež chcete publiku sdělit.
- Vizuálně si poster rozvrhněte – místo pro záhlaví, obrázky, text – s využitím vícesloupcové sazby (minimálně 2 sloupce při sazbě na výšku, 3 sloupce při sazbě na šířku).
- Neobávejte se volného místa – 40 % obrázku, 30 % textu a 30 % volné plochy bude vypadat lépe než poster obsahující „full-text“ článku.
- Barvy (barevný text) používejte s rozmyslem a konzistentně.
- Vkládané obrázky si připravte velké a ne příliš složité.
- Při prezentaci k posteru připněte k rozebrání jeho zmenšenou verzi (A3 nebo A4) případně doplněnou o rozsáhlejší bibliografii nebo technické detaily důkazů.

2.1. Základy

Třída `a0poster` je založena na třídě `article` a její hlavní přínos spočívá ve správném nastavení tiskového zrcadla, okrajů, velikosti písem a matematických symbolů pro přípravu posterů velikosti A3, A2, A1, A0 a „A0big“. Tato třída má tři parametry:

- orientace posteru: `landscape` / `portrait` ;
- velikost posteru: `a0b` / `a0` / `a1` / `a2` / `a3` ;
- kvalita tisku: `final` / `posterdraft`.

Standardní nastavení je `landscape`, `a0b`, `final`, pro klasický A0 poster orientovaný na výšku je tudíž nutné T_EXovský soubor zahájit příkazem:

```
\documentclass[portrait,a0,final]{a0poster}
```

Díky předefinovaným velikostem písma je možné normálně užívat standardní příkazy počínaje `tiny` (12 pt) přes `normalsize` (24.88 pt) až po `VERYHuge` (107 pt).

2.2. Sloupce, barvy, rámečky

Vícesloupcová sazba

Snadné sázení textu do více sloupců umožňuje `multicols{počet sloupců}` prostředí z balíčku `multicol` – balíček je nutné v preambuli načíst tímto příkazem `\usepackage{multicol}`. Toto prostředí funguje i uvnitř `parboxu` a je možné je používat opakovaně k sazbě vnořených vícesloupcových odstavců.

```
\parbox{11 cm}{
  \begin{multicols}{2}
    Ukázka sazby ...
    \begin{multicols}{2}
      Vnořený odstavec ...
    \end{multicols}
    Pokračování ...
  \end{multicols}
}
```

Ukázka sazby do dvou sloupců pomocí prostředí `multicol` s vnořeným odstavcem vysázeným rovněž do dvou sloupců. Pro prostředí `multicol` funguje i v rámci `parboxu`, čehož lze využít při sazbě barevných rámečků.

Vnořený odstavec vícesloupcové sazby pomocí prostředí `multicol`.

Pokračování ukázkového textu vícesloupcové sazby.

Barvy

K oživení posteru či zdůraznění podstatných částí textu dobře poslouží balíček `color`. Barevný text lze vysázet užitím příkazu `\textcolor{barva}`, přičemž jako *barvu* můžeme zadat buďto standardní `red`, `blue`, `green`, či uživatelsky „namíchaný“ odstín ve formátu RGB nebo CMYK `\definecolor{název barvy}{rgb / cmyk}{složky}`, kde *složky* udávají procentuální zastoupení jednotlivých složek RGB, resp. CMYK.

Vlastní šedou lze získat jako `\definecolor{mygrey}{rgb}{0.4,0.4,0.4}`. Obarvení celých prostředí (např. matematiky) pak lze dosáhnout užitím příkazu `\color{barva}`, který „globálně“ mění barvu písma.

```

{\color{blue}
\begin{equation}
\textcolor{red}{\bar{x}}=\frac{1}{n}\sum_{i=1}^nx_i
\end{equation}
}

```

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

Rámečky

Pokud se nám barevný text zdá nedostačující k upoutání pozornosti potenciálního zájemce o náš poster, můžeme sáhnout po balíčku **fancybox** umožňujícím sazbu textu do barevných rámečků. Barevný rámeček typu **parboxu** vytvoří příkaz `\fcolorbox{barva rámečku}{barva výplně}`, přičemž jeho základní vlastnosti (tloušťku čáry a vnitřní okraj) lze nastavit pomocí `fboxrule` a `fboxsep`, např. `\setlength{\fboxrule}{1 pt}`, resp. `\setlength{\fboxsep}{5 pt}`.

Příkaz `fcolorbox` je primárně určen k orámování slov či krátkých matematických výrazů, neboť nedokáže přizpůsobit svou velikost nastavení stránky a patřičně zalamovat vložený text. Pro sazbu složitějších „objektů“ do rámečků a „kontrolu“ jejich chování však lze s výhodou užít vnoření `parboxu` do příkazu `fcolorbox`. Pár ukázek takovéto sazby kombinující rámečky s textem ve více sloupcích je obsaženo v následujících odstavcích.

Matematika v rámečcích ve dvousloupcové sazbě

```

\begin{multicols}{2}
\begin{center}
\fcolorbox{dark}{light}{
\parbox{30 mm}{
\color{superdark}{
$$\bar{x}=\frac{1}{n}\sum_{i=1}^nx_i$$
}}}
\end{center}
Doprovodný text ...
\begin{center}
\fcolorbox{dark}{light}{
\parbox{45 mm}{
\color{superdark}{
$$s^2=\frac{1}{n-1}\sum_{i=1}^n(x_i-\bar{x})^2$$
}}}
\end{center}
\end{multicols}

```


$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Doprovodný text mezi dvěma rámečky s důležitými statistickými vztahy sloužícími k určení základních výběrových

charakteristik, průměru a směrodatné odchylky.

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Orámovaná víceloupcová sazba se stejně širokými sloupečky

```
\begin{center}
\fcolorbox{dark}{light}{
\parbox{11 cm}{
\begin{multicols}{3}
Pokusný text ...
\end{multicols}
}}
\end{center}
```

Pokusný text pro „automatickou“ třísloupcovou sazbu. Text je navíc umístěn v rámečku	<code>fcolorbox</code>	umožňujícím barevné pozadí a orámování všech tří sloupců uživatelsky definovanými barvami.	Je to však za cenu horších řádkových zlomů, jak je to zde dobře vidět.
--	------------------------	--	--

Orámovaná víceloupcová sazba s různě širokými sloupečky

```
\begin{center}
\fcolorbox{dark}{light}{
\parbox{11cm}{
\parbox[t]{75 mm}{
\textcolor{superdark}{Sloupec 1}
...
}
\hfill\parbox[t]{30 mm}{
\textcolor{superdark}{Sloupec 2}
...
}
}}
\end{center}
```

Sloupec 1

Text v prvním širším sloupci včetně sazby matematiky

$$l_{s,t}(\sigma^2, \delta^2) = \prod_{j=1}^{n/2} \frac{f_A(W_j)}{f_H(W_j)}.$$

Sloupec 2

Text ve druhém užším sloupci, tentokrát bez matematiky, zato o něco málo delší.

Hlavička posteru

```
\begin{center}
\parbox[c]{2cm}{
  \includegraphics[width=2cm]{logouk.ps}
}

\hfill \parbox[c]{5cm}{
  \begin{center}
    \textbf{
      \large \textcolor{superdark}{NADPIS POSTERU} \\\
      \smallskip \normalsize \textcolor{dark}{Jméno autora}
    } \\\
    \smallskip \textcolor{dark}{email@univerzita.cz}
  \end{center}
}

\hfill \parbox[c]{2cm}{
  \includegraphics[width=2cm]{logomff.ps}
}
\end{center}
```

NADPIS POSTERU

Jméno autora

email@univerzita.cz

2.3. Obrázky

V následujících odstavcích si představíme dva užitečné balíčky pro vkládání a práci s obrázky. Nejprve balíček **picinpar** umožňující obtékání textu okolo obrázku, tabulky, apod., poté balíček **psfrag**, který nám pomůže při sázení českých popisků, či matematiky do vkládaného postscriptového obrázku.

Na okraj poznamenejme, že pro vkládání obrázků používáme víceméně ze zvyku příkaz `\includegraphics` z balíčku `graphicx`, což v závislosti na vkusu uživatele může být nahrazeno např. příkazem `\epsfig` z balíčku `epsfig` či `\epsfbox` z balíčku `epsf`.

Obtěkání obrázků

Balíček `picinpar` definuje prostředí `window` [*nl*, *zarovnání*, *obsah*, *popisek*], a jeho dvě varianty `figwindow`, respective `tabwindow`, která lze následně umístit do textu tak, že jsou textem obtékána. Čtyři parametry určují počet řádků *nl* textu před začátkem vkládaného okénka, jeho zarovnání *l*, *c*, *r* (left, center, right) vzhledem k okolnímu textu, samotný *obsah* vkládaného okénka – obrázek, tabulka, text – a případně vysvětlující *popisek*, hodící se zejména při vkládání obrázků a tabulek.

```
\begin{window}[2,1,\includegraphics[height=3cm]{img.eps},{}]
\noindent Text obtékající obrázek při užití balíčku
\texttt{picinpar} ...
\end{window}
```

Text obtékající obrázek při užití balíčku `picinpar`. Text může obsahovat několik odstavců a může být **libovolně formátován**. Sazba jednoduché matematiky také nečiní žádné obtíže, např.

$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

Dokonce je možné vložit `fcolorbox` s vnořeným `parboxem`.

Pokračování jednoduchého textu obtékajícího obrázek, nyní pouze prostého bez jakýchkoli efektů. A ještě jedna věta, aby bylo pořádně vidět, že text

obrázek opravdu obtéká.

Dodejme, že podobných efektů lze dosáhnout i užitím balíčků `wrapfigure`, resp. `floatfigure`.

Psfrag

Balíček `psfrag` je užitečný zejména tehdy, když potřebujeme obrázek doplnit o pěkně vysázenou matematiku, české popisky či chceme sjednotit „vizáž“ vkládané grafiky. Příkaz `\psfrag{co}[posh][posv]{čím}` dokáže nahradit libovolný text *co* obsažený v postscriptovém obrázku téměř libovolným $\text{\TeX}$ ovským výrazem *čím*, přičemž nahrazení se provede při přeložení `.dvi` do `.ps`. Nepovinné

parametry *posh* a *posv* určují horizontální, resp. vertikální, umístění nahrazujícího výrazu *čím* vzhledem k nahrazovanému *co*, přičemž přípustné hodnoty pro *posh* jsou *l*, *c*, *r* (left, center, right), resp. *b*, *B*, *c*, *t* (bottom, baseline, center, top) pro *posv*.

```
\includegraphics[width=5cm]{sincos.eps}
\psfrag{xlabel}[c] []{$t$}
\psfrag{ylabel}[c] []{$\sin(t)$ a~$\cos(t)$}
\psfrag{Obrazek}[c][c]{\textbf{Ukázka použití}\texttt{PSfrag}u}}
\hfill\includegraphics[width=5cm]{sincos.eps}
```

Kromě nahrazování textu **PSfrag** umožňuje nastavením příslušných parametrů text i otáčet, libovolně posunovat a škálovat. Podrobnosti lze nalézt v krátké pěkně napsané uživatelské příručce, která je volně dostupná na stránkách CSTUGu.

2.4. Nastavení tiskového zrcadla

Každý poster-tvůrce dříve nebo později narazí na nastavování velikosti tiskového zrcadla a tomu odpovídajícím okrajům. Následující příkazy nastaví, při zachování standardního nastavení třídy **a0poster**, pěti centimetrové vnější okraje a mezeru mezi sloupci a odpovídající tiskové zrcadlo pro poster velikosti A0.

```
\setlength{\textheight}{108.82cm}
\setlength{\textwidth}{ 73.96cm}
\setlength{\hoffset}{ 2.46cm}
\setlength{\voffset}{ 2.46cm}
\setlength{\columnsep}{ 5.00cm}
```

3. Závěr

Na stránkách CSTUGu a konference Robust (stránkách profesora Antocha) jsou k dispozici „šablony“ posteru a prezentace čekající jen na Váš obsah.

Rádi uvítáme jakékoli Vaše postřehy a zkušenosti s prezentací Vaší práce ať už na konferencích, seminářích či ve výuce. Doufáme, že předcházející řádky byly pro Vás alespoň trochu inspirativní.

Šablona pro poster pro A0poster je uložena na serveru CSTUGu nebo je dostupná na <http://studium.uis.fame.utb.cz/zpravodaj/pp.rar>.

Použité a doporučené zdroje

Stránky byly platné k 7. dubnu 2008.

- <http://www.karlin.mff.cuni.cz/~antoch/>
Šablony na konferenci Robust 2006.
- <http://www.ufal.mff.cuni.cz/~antoch/robust06/postery.htm>
Postery z proběhlé konference Robust 2006.
- <http://www.cstug.cz/>
- <http://nxg.me.uk/docs/posters/>
- <http://www.ncsu.edu/project/posters/> Jak dělat vědecký poster.
- <http://www.statapol.cz/> Stránky České statistické společnosti.

Summary: Posters in T_EX

The creation of a poster in this article is done in the A0poster class. It includes examples of packages such as multicols, color, fancybox, graphicx, epsf, picinpar and psfrag. In the next section of the article the authors discuss the settings for the layout of the poster. In the last section the authors present a template for a poster to be created in A0poster class. Two real-world posters are inserted in the conclusion.

Key words: Poster Creation, A0poster, Multicol, Color, Fancybox, Picinpar, PS-Frag, Real-world Posters.

Luboš Prchal, KPMS MFF UK Praha
prchal@karlin.mff.cuni.cz

Pavel Schlesinger, ÚFAL MFF UK Praha
schlesinger@ufal.mff.cuni.cz

CHANGE POINT ANALYSIS OF ATMOSPHERIC RADIATION PROFILES

LUBOŠ PRCHAL

lubos.prchal@st.cuni.cz

Department of Statistics and Probability, Charles University, Prague

SUMMARY

The poster presents a statistical analysis of the dependance of atmospheric radioactivity on the altitude. As a theoretical model explaining the physical background of this process is not known, two parametric regression models based on Richards growth curve are proposed. We discuss several computational challenges coming from the parameter estimating procedure and we focus on the choice of software suitable for calculations. The second part is devoted to the statistical analysis of changes in variance of our measurements. It appears that the random terms cannot be modelled by a series of i.i.d. random variables. On the contrary, the functional model consisting of three segments with two unknown change points seems to be much more appropriate. Assuming normal distribution of the error terms we propose test statistic for the detection of the linear trend in the variance and show its limit distribution. To complete the work, we estimate the change points and the variance parameters using several methods and study their properties.

I METEOROLOGICAL EXPERIMENT

Analyzed data came from Prague-Libuš upper air meteorological station of the Czech Hydrometeorological Institute, where every month the vertical profiles of beta and gamma radioactivity are measured by means of radiometric sensors. The sensors consist of a Geiger-Müller gamma and beta tubes. It is the task of the meteorological balloons platform and beta tubes to fly from the surface up to 35 km and detect short current pulses coming from the interaction between the radiation and the tube wall material.

After several "recalibrations" the complicated measurement process results into the data pairs $(x_i, y_i)_{i=1}^n$, where x represents the altitude and y the average number of pulses per second in the tube wall.

Let us note that y is proportional to the radiation intensity.

II APPLIED REGRESSION MODELS

The first aim of the analysis was to suggest a parametric regression model $Y_i = m(X_i) + \varepsilon_i$, where $m(\cdot)$ represents mean amount of radiation and ε_i a random "error" term. As described by Hubinila (2004), the models $m(\cdot)$ are based on Richards growth curve $R(x)$ and its derivative $r(x)$. Unfortunately, Richards curve itself does not describe properly the measurements in low atmospheric layers. That is why we propose two extended additive models. The first model consists of $r(x)$ and a simple linear function, meanwhile the second model of $r(x)$ and logistic growth curve being of the form

$$\begin{aligned} m_1(x) &= r(x) + cx + k; \\ m_2(x) &= r(x) + e \left(1 + \exp \{ -f(x - g) \} \right) + k. \end{aligned}$$

IV ANALYSIS OF VARIABILITY

Radiactivity measurements evidently show heteroscedasticity. The goal was to suggest a suitable parametric model, estimate its parameters and study its properties.

PARAMETRIC MODEL

It appears that the errors ε_i cannot be modelled by a series of i.i.d. random variables. To gain an idea about a suitable parametric model for the variance, we fitted squared errors using a kernel estimator as shown in the figure. Based on this estimation, the parametric functional model for $\sigma^2(x_i) = \text{var } Y_i$ in the form of

$$\sigma^2(x) = \sigma^2 + \delta^2 \left[\frac{x - x_0}{x_1 - x_0} \right] (x \in (x_0, x_1)) + \lambda (x \in (x_1, \infty))$$

seems to be much more appropriate. Having two unknown real parameters $\sigma^2 > 0$ and $\delta \geq 0$ it consists of three segments and may change its behavior in two unknown altitudes (two unknown change points x_0 and x_1).

CHANGE POINT DETECTION

A natural question appears: Is the variance constant or not? More precisely, the question is how to perform a statistical test of the hypothesis of a constant variance against the alternative of two unknown change points.

Assuming independent $Y_i \sim N(m(x_i), \sigma^2(x_i))$ and being inspired by Gupta and Ramamyaiah (2001), we propose a test statistic and show its limit distribution. Main ideas and key steps are summarized in the box below. The formal, very long and tedious proof is available on request.

$$\begin{aligned} H: \sigma^2(x) &= \sigma^2, \\ A: \exists x_0, x_1 \in \{x_1, x_2, \dots, x_n\}, \\ \sigma^2(x) &= \sigma^2, \quad x - x_0 \leq x_1, \\ &= \sigma^2 + \delta^2 \frac{x - x_0}{x_1 - x_0}, \quad x_0 < x \leq x_1, \\ &= \sigma^2 + \delta^2, \quad x_1 < x \leq x_n. \end{aligned}$$

III COMPUTATIONAL CHALLENGES

We used the classical Least Squares method to estimate the unknown parameters of suggested models $m_1(x)$ and $m_2(x)$. Although the principle seems to be quite simple, the minimization procedure must be performed numerically resulting in several different computational problems and challenges.

SETTING UP THE INITIAL PARAMETERS

Numerical methods need initial parameters estimators to start the procedure with. The key to set up appropriate initial parameters is to understand their interpretation. The choice of starting points resulting from the analysis of the parameters is described in detail by Hühnle (2004).

NUMERICAL NON-STABILITY

The numerical stability of the minimization procedure is influenced by several factors. Because of the analytic form of $r(x)$ one must pay attention on **representation of small numbers**, exclude **non-sense values of parameters** and think about **appropriate data scaling**.

CHOICE OF SOFTWARE

Almost every statistical or mathematical software offers some implementation of numerical methods suitable for the Least Squares method. However, all implementations are able to solve our computational problem with 6 (or 8) **parameters** to be estimated. Notice that we have not yet fixed the scale of the available fit of the data, but by knowing that the estimated **submodel** (e.g. $r(x)$ and the logistic curve) and their parameters are **interpretable**. To obtain the best possible parametrization we applied the optimization procedure implemented in several software packages. **Finally, we decided for Matlab, Release 13, and its lacquerfit function.**

Methods implemented in **R** and **S Plus** softwares **do not converge** at all, even if the initial parameters are set to the "optimal" solution provided by Matlab.

Matlab, Release 13, contains two suitable functions. Presented **results were obtained** by more sophisticated **lacquerfit** function from the Optimization Toolbox (right figure), while the simpler **nlinfit** function from the Statistics Toolbox convergates into uninterpretable parametrizations (left figure).

1 TRANSFORMATION

Assuming independent and normal L_i we get

$$W_i = \frac{\sigma_i^2}{2} \left(\frac{L_i^2}{\sigma_i^2} + 1 + \frac{L_i^2}{\sigma_i^2} \right) \sim \text{Exp}(\frac{1}{2\sigma_i^2})$$

2 MAXIMUM LIKELIHOOD APPROACH

For fixed change points s and t we have Ml. ratio

$$L_d(\sigma^2, \delta^2) = \prod_{i=1}^n \frac{L_i(W_i)}{2\pi \sigma_i^2}$$

Using the **same type principle** for unknown s, t we obtain the test statistic in the form of

$$T = \text{const.} \cdot \frac{\sum_{i=1}^n W_i}{\sum_{i=1}^n W_i}$$

3 MOMENTS OF T

To calculate moments of T denote

$$Z_j = W_j \left(\sum_{i=1}^n W_i \right)^{-1}$$

The vector $(Z_1, Z_2, \dots, Z_{n-1})$ has $n/2 - 1$ dimensional Dirichlet distribution with all parameters equal 1. Moments of T can be obtained in terms of Z_j for fixed n .

4 LIMIT DISTRIBUTION

The test statistic has asymptotic normal distribution, i.e.

$$V = \frac{T - ET}{\sqrt{\text{Var } T}} \sim N(0, 1)$$

and $n = 100$ is enough to use its asymptotic properties.

ESTIMATORS

As expected, we rejected the null hypothesis for all data sets. To complete, then we have to estimate the change points of the model. We suggest estimators based on the **Least Squares** method. **Weighted Least Squares** method and the L_1 approach using **Iteratively Weighted Least Squares** method and the **linear programming**. L_1 approach seems to be satisfactory for the change points estimations, on the contrary LS and WLS methods are much more appropriate for estimating the variance parameters.

The **optimal estimator** combines the L_1 -estimator for change points with the **WLS-estimator** of variance parameters σ^2 and δ^2 .

Warning !!! As shown in the figures, the change points estimators based on the Least Squares methods (blue line) are **extremely sensitive to the "outliers"**. On the contrary, L_1 approach (red line) provides much more stable estimations.

V FUTURE PLANS

The presented work should be understood as the first step in analyzing atmospheric radiation. However, a lot of interesting statistical and meteorological questions still remain without satisfactory answers. Among them, we would like to focus mainly on the following:

- how to **detect a seasonality** in our measurements when considered as functional data;
- how to **estimate quantiles** of the radiation in different altitudes to be able to set up an alarm system;
- how to **improve numerical methods** to obtain more stable parameterizations of proposed models.

Acknowledgements. Author would like to express his thanks to Prof. Jacarita Ant6nio for his generous support, valuable comments and help. The paper was supported by

[1] Gupta A.K., Bhanuwarake A. (2001). Change points with linear trend for the exponential distribution. J. Statist. Theory, Inform. 30, 381 - 396.
[2] Gupta A.K., Bhanuwarake A. (2002). Change points with linear trend for the exponential distribution. J. Statist. Theory, Inform. 31, 105 - 124.
[3] Pridal L. (2004). *Improvements to estimators for the functional data analysis* (in Czech). Master thesis, MFF UK Praha.

Aplikace genetických algoritmů v procesu plánování výroby ve společnosti Mitas, a. s.

Roman Kasal, Petr Klímeck, Jozef Říba, Pavel Stráž

Genetické algoritmy

Genetické algoritmy představují stochastickou prohledávací metodu, která je inspirována genetickými principy jako je přirozený výběr, křížení a mutace. I když se první pokusy o modelování evoluce na počítači objevily už v padesátých letech, za objevitele genetických algoritmů tak, jak je známe dnes, je považován John Holland, který položil základy této disciplíny v sedmdesátých letech.

Aplikace genetických algoritmů

Pro aplikaci genetických algoritmů rozhodla složitost výrobního procesu produkce motoplášťů ve společnosti Mitas, a. s., kde tento proces dokáže optimalizovat snad jen řízená metoda pokus-omyl. Genetické algoritmy právě tuto metodu více než vylepšily a jsou snad jediným užívaným nástrojem pro řešení velmi složitých problémů s velkým množstvím proměnných.

Aplikace genetických algoritmů bude stručně popsána v pěti číslovaných krocích.

1. Sběr potřebných informací

Časově nejnáročnější část. Obrázek (1) schematicky ukazuje nejdůležitější cíl úkolu – popis provozu pro následné řešení genetickými algoritmy.

(1) Schéma provozu

4. Využití genetických algoritmů při řešení maximalizace produkce – Evolver

K řešení problému byl velmi nápomocen již vyvinutý software Evolver od společnosti Palisade, který je určen k produktu Microsoft Excel. Dle vložených proměnných nástroj zvládnout každého výrobku řešení, záda má být vyroben a v jakém množství, a to pro každý den ze 14 plánovaných.

2. Predikce množství produktů potřebného k výrobě

- regresi analýza se sezónní složkou
- harmonická analýza
- SARIMA modelování
- umělé neuronové sítě

3. Vytvoření matematického modelu provozu v úzkém místě

- Model, jak můžeme slednout na obrázku (2), zahrnuje zejména:
- množství, které je třeba vyrobit,
 - skladové zásoby,
 - čas lisování,
 - atd.,
 - dále nejdůležitější omezení spočívající v 22,5 hodinách provozu za den, které je nutné plně využít.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) Matematický model provozu

Závěr

Genetické algoritmy se opravdu jeví jako prozatím nejlepší nástroj k výpočtu složitých problémů, které nelze řešit analyticky. Správné algoritmování problému a využití vhodné perturbací metody může ušetřit velmi mnoho času při řešení, což se ostatně potvrdilo.

Evolver během několika vteřin využít kapacity všech lisů a následně již jen optimalizoval výrobu tak, aby maximalizoval výstup s penalizací přechodu z jedné receptury na novou mezi jednotlivými dny.

5. Problém obchodního cestujícího

Problém obchodního cestujícího je sice označen jako NP problém, ale algoritmus je samozřejmě možné propůjčit i například problémům typu stanovení vhodného pořadí receptur do výroby s neukončenou Hamiltonovou kružnicí při penalizaci z přechodu mezi výrobky, kde například penalizace přechodu z výrobku A na výrobek B může být odlišná pro opakné pořadí (přechod z výrobku B na výrobek A), což je přesně náš případ. Koneckonců se toto specifikum může objevit i v realitě u problému obchodního cestujícího z důvodu jednosměrných ulic. Úkol byl řešen v Matlabu.

Diagram aplikace genetických algoritmů v problému obchodního cestujícího

Výsledná hodnota 167 minut při výměně 105 receptur, tzn. 104 změn, je v porovnání s počtem náhodou permutací pořadí receptur (1162 minut) velmi dobrý výsledek. Řešení představuje sedminásobně lepší využití času.

Kontakty

Roman Kasal kasal@fame.utb.cz
 Petr Klímek klimek@fame.utb.cz
 Jozef Říha rha@fame.utb.cz
 Pavel Stříž striz@fame.utb.cz

Ústav informatiky a statistiky
 FAME UTB Zlín
 Mostní 5139
 Zlín 760 01

1. Úvod

V následujících odstavcích se s Vámi chceme podělit o naše zkušenosti, postřehy a některé problémy, jimž jsme čelili při přípravě prezentací v T_EXu, konkrétně s využitím balíčku **Beamer**.

2. Beamer

Pokud se čas od času potřebujete vypořádat s příspěvkem na konferenci, případně s odbornou přednáškou či hodinou výuky, prostě pokud potřebujete něco odprezentovat, máte v dnešní době *kompjůtrů* možnost, kromě už poněkud zastaralých klasických slidů, sáhnout po několika specializovaných programech. Těm z vás, kteří navíc rádi T_EXujete, bychom rádi představili jednu z možných alternativ vhodnou (nejen) pro prezentace s matematickými texty. Produkt, který se vám pokusíme na následujících řádcích přiblížit, nese název Beamer (<http://latex-Beamer.sourceforge.net/>).

Ke kvalitám Beameru rozhodně patří jednoduchost instalace, snadná práce s uspokojivými výstupy i pro začátečníky, možnosti modifikace a rozšíření pro pokročilé a v neposlední řadě přiložená podrobná a čtivá dokumentace – tzv. Beameruserguide (pro další BG). Najdeme v ní čtivý tutoriál v podobě rad panu Eukleidovi ve smyšleném světě, kde si slavný matematik připravuje elektronickou prezentaci svých posledních převratných objevů. Rady a tipy zde zmíněné mají obecnou platnost pro zákonitosti dobré prezentace a poslouží čtenáři při tvorbě vlastních prezentací i v jiných programech než je Beamer. Autor programu připravil navíc šablony prezentací šitých na míru podle časové délky. Pro nejjednodušší start s Beamerem tedy zvolte editaci těchto souborů z adresáře **solutions** doplněnou o inspiraci z hotových příkladů z adresáře **examples**.

2.1. Instalace

Beamer je naštěstí již nativní součástí distribucí T_EXu pro Windows jako T_EXLive či MiK_T_EX, kterou používá asi valná většina čtenářů Zpravodaje pracujících pod operačním systémem Windows. V těchto distribucích je Beamer buď již nainstalovaný nebo jej lze přidat jednoduchým zaškrtnutím příslušného balíčku stejného (případně drobně změněného) jména v programu pro administraci distribuce.

Pro operační systémy Unix/Linux je Beamer většinou součástí distribuce (např. Debian, Ubuntu, Gentoo, Mandriva, Fedora), existuje tedy balíček a jeho instalace je pro administrátora stejně snadná jako u dvojice $\text{T}_{\text{E}}\text{X}$ Live/ $\text{M}_{\text{I}}\text{K}_{\text{E}}\text{T}_{\text{E}}\text{X}$. Pokud nemáte možnost využít výhod jakékoli distribuce, nezbývá než jít nejpracnější cestou a instalovat Beamer stáhnutím archivu (ve formátu **tar.gz** nebo **zip**) s následným rozbalením do adresáře s $\text{T}_{\text{E}}\text{X}$ em, po kterém musí navíc následovat další kroky (více viz BG, podkapitola 2.3.).

Kontrola

Kontrola bezproblémové instalace může být provedena na dvojnásobném (!) přeložení – z důvodu tvorby odkazů, obsahu atp. – některého z příkladů (např. **beamerexample1.tex**), které jsou součástí distribuce:

```
pdflatex beamerexample1.tex
pdflatex beamerexample1.tex
```

či

```
latex beamerexample1.tex
latex beamerexample1.tex
dvips beamerexample1.dvi
ps2pdf beamerexample1.ps
```

Z příkladů je vidět, že si Beamer rozumí jak s překladem pomocí příkazu **latex** tak **pdflatex**. V prvním případě je pak následně nutné navíc použít **dvips** a **ps2pdf**. Volba, kterou cestou jít, bude asi výrazně ovlivněna formátem užitých obrázků a dalšími balíčky – např. použití balíčku **psfrag** vylučuje přímé přeložení příkazem **pdflatex**, naopak pro jeho správné chování je nutné užít **dvips**.

Výsledkem je v obou případech soubor **beamerexample1.pdf**, který je připraven pro prezentaci v jakémkoli pdf prohlížeči (nejčastěji v Acrobat Readeru). Již na první pohled nového uživatele příjemně překvapí přítomnost ikon pro ovládání, přechod na následující/předchozí slide či sekci a ikony pro vyhledávání. To vše jistě zpříjemní ovládání během samotné prezentace.

2.2. Základní slide

Ze všeho nejdříve je třeba v preambuli $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u říct, že používáme Beamer:

```
\documentclass{beamer}
```

Základní samostatnou stavební jednotkou prezentace není stránka (**page**), ale nativní prostředí Beameru **frame**, které vytvoří jeden slide. Každý slide pak může

mít svůj název (`frametitle`). Uvnitř prostředí `frame` pak můžeme používat klasické $\TeX$ ovské/ $\LaTeX$ ovské příkazy a prostředí jako například `includegraphics`, `itemize` či `enumerate`. Pro přehlednost se doporučuje jednotlivé slidy shlukovat do sekcí a subsekcí stejně, jako jsme zvyklí činit se stránkami v článku. Minimalistická šablona pro jeden slide tedy může vypadat následovně:

```
\begin{frame}
  \frametitle{Nadpis pokusného slidu}
  První povídání o~Beameru

  \begin{itemize}
    \item První odrážka
    \item Druhá odrážka
    \item \ldots
  \end{itemize}

  Závěr prvního slidu o~Beameru
\end{frame}
```

S dalšími rozšířeními přichází sám Beamer v podobě nových příkazů. V praxi asi nejvíce využitelné je možnost rozdělit slide na podčásti příkazem `pause`, které se při prezentaci budou objevovat postupně. Představme si například, že popisujeme postup metody v třech následných krocích a chceme každý krok slovně více rozvést. Publiku zpravidla nechceme hned při prvním pohledu na slide zastrašit zevrubnou informací např. o třetím kroku, když teprve mluvíme o prvním. Výsledkem postupného odkrývání obsahu má být získání pozornosti a nerozptylování aktuálně méně důležitými částmi slidu:

```
\begin{enumerate}
\item 1. krok metody \pause % bude vidět od začátku slidu
\item 2. krok metody \pause % bude vidět od 2. části
\item 3. krok metody \pause % bude vidět od 3. části
\end{enumerate}
```

Protože je tato struktura velice častá, její zápis při užití prostředí `itemize` a `enumerate` nám usnadní následující „konstrukce“:

```
% jeden komentář před [<+>] zruší rozdělení slidu na podčásti
\begin{enumerate}<+>
\item 1. krok metody
\item 2. krok metody
\item 3. krok metody
\end{enumerate}
```

Ke zvýraznění důležitého slova či matematické formule slouží příkaz `alert`. V následujícím zdrojovém kódu si jeho užití vysvětlíme na pokročilejším příkladě v kombinaci s možností vnořovat výčtová prostředí a s možností změnit nepovinným parametrem u položky `item` znak pro představení:

```
\begin{itemize}[<+>]
\item \alert{Výhody}
  \begin{itemize}
    \item[+] Intuitivnost
    \item[+] Snadná interpretace
  \end{itemize}
\item \alert{Nevýhody}
  \begin{itemize}
    \item[-] Paměťové nároky
    \item[-] Časová složitost
  \end{itemize}
\end{itemize}
```

Zvláštní pozornost by měla být vždy věnována úvodu prezentace. Pro představení autorů a názvu příspěvku poslouží následující část preambule:

```
\title[kratší název]
{Klidně i~dlouhý \\
název prezentace}

\author{autorA \inst{1} \and autorB \inst{2}}

\institute{
  \inst{1}%
  Univerzita A\\
  MěstoA
  \and
  \inst{2}
  Univerzita B\\
  Město B\\
}

\date{jméno konference \\ datum}
```

První slide pak Beamer vytvoří sám pomocí

```
\begin{frame}
\titlepage
\end{frame}
```

Druhý slide bývá často věnován představení obsahu příspěvku, tedy jakési osnově, které se budeme při prezentaci držet. Zde je místo pro uvedení publika do děje a vymezení obsahu, jehož se bude příspěvek týkat. Zde nám pomůže dělení celé prezentace na sekce a podseky. Pakliže budeme dělení přísně dodržovat, můžeme druhý slide s osnovou vytvořit takto:

```
\begin{frame}
\frametitle{Osnova}
\tableofcontents%[pausesections] nebo [pausesubsections]
% Nepovinným parametrem můžeme nastavit pauzy mezi
% sekcemi nebo subsekcemi
\end{frame}
```

2.3. Modifikace

Po zvládnutí základních principů pro práci s Beamerem, které jsme se snažili vysvětlit v předchozí části, je čas představit si pokročilejší modifikace vzhledu prezentace. Nadále budeme předpokládat, že již máme napsaný obsah prezentace. Jde nám tedy o úpravy vzhledu.

Příkazy `use*theme`

Beamer přichází se sadou vnitřních proměnných, jejichž účinek bychom mohli přirovnat k užití kaskádových stylů při tvorbě HTML stránek.

Hodnoty těchto vnitřních proměnných se nastavují příkazy, které mají v názvu `use*theme{hodnota}`. Celkový vzhled prezentace ovlivňuje příkaz `usetheme`, výběr fontů pro písmo příkaz `usefonttheme` a nastavit barevné schéma lze příkazem `usecolortheme`. Vyzkoušejte:

```
\usetheme{Warsaw} či \usetheme{Singapore}
\usefonttheme{structurebold} či \usefonttheme{serif}
\usecolortheme{seahorse} či \usecolortheme{rose}
```

Do této kategorie příkazů patří také příkaz `\useinnertheme{hodnota}`, kterým můžeme měnit chování elementů „uvnitř“ slidy, např. příkazem `\useinnertheme{circles}` či `\useinnertheme{rectangles}`

nastavíme znak pro odrážky výčtového prostředí `enumerate` na kroužky či obdélníčky. Protikladem je příkaz `\useoutertheme{hodnota}`, kterým ovlivníme např. vzájemnou polohu loga, záhlaví, zápatí, navigačních ikon pro přechod na následující/předchozí stránku a v neposlední řadě taky navigace pomocí jmen sekcí a subsekcí.

Příkazy `setbeamer*`

Skupina příkazů sloužící k nastavení vlastností jednotlivých prvků prezentace, má tvar `setbeamer*{prvek}{hodnota}`. Narozdíl od příkazů `use*theme`, které jsou intuitivní a dobře popsané v BG, použití a možnosti `setbeamer*` jsou zastřeny tajemstvím. Z vlastní zkušenosti autorů se pak stává, že vnutit konkrétnímu parametru jeho hodnotu znamená vyhledat jeho definici v příslušném beamerovském souboru a tu pak formou pokusů a omylů přizpůsobit obrazu svému. Následující řádky ukazují, jak „vypnout“ záhlaví a zápatí slidu, nastavit formát `frametitle` na tučné kapitálky, nastavit okraje a změnit některé barvy:

```
\setbeamertemplate{headline}{}
\setbeamertemplate{navigation symbols}{}
\setbeamertemplate{itemize item}{
  \tiny\raise1.25pt\hbox{\donotcoloroutermaths$\blacksquare$}
}

\setbeamersize{text margin left=1cm}
\setbeamersize{text margin right=1cm}

\setbeamerfont{frametitle}{series=\bfseries,shape=\scshape}

\definecolor{mystr}{rgb}{0.15,0.15,0.75}
\definecolor{myalert}{rgb}{1,0.15,0.15}
\setbeamercolor{structure}{fg=mystr}
\setbeamercolor{item}{fg=black}
\setbeamercolor{alerted text}{fg=myalert}
```

Alternativní výstupy

Již jsme se zmínili o tom, že zdrojový text beamerovské prezentace je možno přeložit i pomocí programu `dvips`. Chceme-li předejít zbytečným komplikacím, doporučuje se tuto volbu Beameru natvrdo vnutit upřesněním pomocí nepovinného parametru:

```
\documentclass[dvips]{beamer}
```

Zajímavější hodnoty tohoto nepovinného parametru jsou **handout** a **trans**. První připraví výstup vhodný pro tisk prezentace v podobě tzv. handoutů, kdy můžeme vytisknout až čtyři slidy na jednu stránku. Druhá hodnota je podobná, potlačí se rozdělení slidů na podčásti, jeden slide však bude nadále odpovídat jedné stránce.

Dokumentační příručka a solutions

Již jsme se zmínili o příloze programu Beamer, o příručce BG. Kromě technických částí popisujících Beamer stojí jistě za zmínku zhruba prvních 40 stran. Najdeme zde čtivý tutoriál v podobě rad panu Eukleidovi ve smyšleném světě, kde si slavný matematik připravuje elektronickou prezentaci svých posledních převratných objevů. Rady a tipy zde zmíněné mají obecnou platnost pro zákonitosti dobré prezentace a poslouží čtenáři při tvorbě vlastních prezentací i v jiných programech než je Beamer. Na závěr dlužno podotknout, že autor programu připravil šablony prezentací šitých na míru podle časové délky. Pro nejjednodušší start s Beamerem tedy zvolte editaci těchto souboru z adresáře **solutions** doplněnou o inspiraci z hotových příkladů z adresáře **examples**.

Šablona pro prezentaci v Beameru je uložena na serveru ζ TUGu nebo je dostupná jako <http://studium.uis.fame.utb.cz/zpravodaj/pp.rar>.

Použité a doporučené zdroje

Stránky byly platné k 7. dubnu 2008.

- <http://www.karlin.mff.cuni.cz/~antoch/>
Šablony na Robust 2006.
- <http://latex-Beamer.sourceforge.net/>
Stránky o $\text{\LaTeX}$ ovém projektu Beamer.
- <http://www.cstug.cz/>
Stránky Československého sdružení uživatelů $\text{\TeX}$ u.
- <http://www.statapol.cz/>
Stránky České statistické společnosti.

Summary: Presentations in $\text{\TeX}$

In this article, the authors share their knowledge, notes and experience with the Beamer presentation class. The article includes installation notes and the first steps in Beamer, the **pause** command, **<+>**, **+** and **-** options, generating a title page and a table of contents. It also explains how to change the design of a presentation by setting `\use*theme{value}` and `\setbeamer*{element}{value}`.

In the conclusion of the article, the authors recommend BeamerGuide and a few internet resources for further reading. The Beamer template of the authors is published independently on their website and on the $\LaTeX$ TUG server.

Key words: Presentation Creation, Presentation Template, Beamer Class.

Luboš Prchal, KPMS MFF UK Praha
prchal@karlin.mff.cuni.cz

Pavel Schlesinger, ÚFAL MFF UK Praha
schlesinger@ufal.mff.cuni.cz

Ukázky prezentací

PAVEL STRÍŽ A MICHAL POLÁŠEK

1. Úvod

V následujících odstavcích okomentujeme jednu prezentaci postupně připravenou v balíčcích PDFSlide a PDFScreen a třídě Beamer.

Poslední ukázka budou výseky, chceme-li vybrané pasáže, z úvodního slova obhajoby disertační práce připravené v Beameru. Všechny ukázky, šablony a kódy jsou dostupné na <http://studium.uis.fame.utb.cz/zpravodaj/pp.rar> nebo na serveru $\LaTeX$ TUGu.

2. PDFSlide

Začněmež balíčkem PDFSlide. V preambuli dokumentu zavedeme standardně používané balíčky, a na řádku 7 pak vlastní balíček pdfslide.

```
1 %pdflatex pdfslide--prezentace.tex %TeXLive 2007
2 \documentclass{article}
3 \usepackage[czech]{babel}
4 \usepackage[cp1250]{inputenc}
5 \usepackage[IL2]{fontenc}
6 \usepackage{times}
7 \usepackage{pdfslide}
```

Pokračujeme zavedením informací prezentovaných na úvodním slajdu, plus vhodné definování pozadí přes příkaz `\overlay`. Nezbyývá, než zavést tělo dokumentu na řádku 13 a vygenerovat první slajd na řádku 15.

```

8 \title{Velká firma, a.~s.}
9 \author{Polášek Michal}
10 \date{22.~10.~2007}
11 \overlay{pozadi.jpg}
12
13 \begin{document}
14 \begin{center}
15 \maketitle
16 \end{center}

```

Nyní již následuje běžná sazba dalších slajdů. Každé zavedení `\section`, řádky 17, 33, 42 a další, automaticky přepínají na další obrazovku.

```

17 \section{Charakteristika firmy}
18 \begin{tabular}{ll}
19 Název:&Velká firma, a.~s.\\
20 Základní kapitál:&2.000.000~Kč\\
21 Počet zaměstnanců:&773\\
22 Sídlo:&Adresa 1, Olomouc -- Přívoz \\
23 Provoz:&Adresa 2, Přerov\\
24 \end{tabular}
25 \begin{itemize}
26 \item Koupě zboží za účelem jeho dalšího prodeje a~prodej.
27 \item Poskytování vzdělávacích aktivit.
28 \item Zprostředkovatelská činnost v~oblasti obchodu a~služeb.
29 \item Pronájem movitých věcí a~nemovitostí.
30 \item Zpracování gumárenských směsí.
31 \end{itemize}
32
33 \section{Základní poslání}
34 \textit{"Řešení pro Vás. Kvalitně, rychle a~včas."}
35 \begin{itemize}
36 \item Nabízet zajímavé výrobní portfolio.
37 \item Udržovat kvalitu produktů.
38 \item Zhodnocování kapitálu podniku a~tržní expanze.
39 \item JIT koncepce ve výrobě, skladování a~dopravě.
40 \end{itemize}
41
42 \section{Dominantní trendy}
43 \subsection{Postavení na trhu}
44 Výrobní portfolio -- z~větší části gumárenské produkty.
45 \begin{itemize}
46 \item V~rámcí Moravy i~Čech.
47 \item Zákazníci -- X, Y, Další a.~s., \ldots
48 \item 80 \% tuzemský trh, 20 \% tuzemský trh.
49 \end{itemize}
50 \begin{flushright}
51 \includegraphics[width=2cm]{pie.jpg}
52 \end{flushright}

```

Pokud používáme více úrovní nadpisů, pak je vhodný ruční přechod přes příkaz `\pagebreak`, jako je tomu na rádcích 53 či 75. Obrázky jsou vkládány na

řádcích 51, 62, 72, 83 a 92.

```
53 \pagebreak\bigskip
54 \subsection{Získávání finančních prostředků}
55 \begin{itemize}
56 \item Úvěry — nejsou problémové (jištění HIM a~NIM).
57 \item Odvětví i~podnik je ziskové $\text{\textbf{Rightarrow}}$
58 zisk pokrývá úroky.
59 \item Snadná dostupnost cizího kapitálu.
60 \end{itemize}
61 \begin{flushright}
62 \includegraphics[width=2cm]{dollar.jpg}
63 \end{flushright}
64
65 \subsection{Výrobní proces a~jeho inovace}
66 \begin{itemize}
67 \item Stálost procesů — mechanizace, technologické postupy.
68 \item Nové technologie a~stroje = nové možnosti.
69 \item Snaha o~zvyšování kapacit.
70 \end{itemize}
71 \begin{flushright}
72 \includegraphics[width=2cm]{inovace.jpg}
73 \end{flushright}
74
75 \pagebreak
76 \subsection{Získávání a~rozvoj HR}
77 \begin{itemize}
78 \item Nízká fluktuace pracovních sil.
79 \item Potřeba školeného personálu.
80 \item Nedostatek nekvalifikované pracovní síly.
81 \end{itemize}
82 \begin{flushright}
83 \includegraphics[width=2cm]{hr.jpg}
84 \end{flushright}
85
86 \subsection{Ostatní trendy}
87 \begin{itemize}
88 \item Ekologie, standardizace.
89 \end{itemize}
90 \vspace{1cm}
91 \begin{flushright}
92 \includegraphics[width=2cm]{recycle.jpg}
93 \end{flushright}
94
95 \section{Kritické faktory úspěšnosti firmy}
96 \begin{itemize}
97 \item Dobré dodavatelsko—odběratelské vztahy.
98 \item Výrobní pružnost.
99 \begin{itemize}
100 \item Využití kapacit.
101 \item Pojištění vstupů (alternativní dodavatelé).
102 \end{itemize}
102 \end{itemize}
```

```

103      \item Technické a~servisní zajištění .
104      \item Dopravní dostupnost.
105      \end{itemize}
106 \end{document}

```

V naší ukázce není příliš vhodně užité pozadí prezentace s bílým pozadím JPG obrázků, to lze samozřejmě pořešit několika cestami. To však není náplní těchto ukázek. Šikovný čtenář jistě zaexperimentuje sám.

3. PDFScreen

Nyní zkusíme tuto prezentaci překlomit do balíčku PDFScreen, aniž by nám šlo o identické zobrazení vůči balíčku PDFSlide či o zvláštní péči s pozadím obrázků. Pro úsporu místa však střední část slajdů vynecháme. Zaměříme se na rozdíly.

Zavedeme globální parametry na řádku druhém, nezbytné a standardní balíčky na řádku třetím až šestém a balíčky bonusové na řádku 7.

```

1  %pdflatex pdfscreen--prezentace.tex
2  \documentclass[a4paper,11pt]{article}
3  \usepackage[czech]{babel}
4  \usepackage[cp1250]{inputenc}
5  \usepackage[IL2]{fontenc}
6  \usepackage{times}
7  \usepackage{xspace, colortbl}

```

Jádrem nastavení balíčku PDFScreen jsou následující tři řádky. Následují identifikace autora. Poté začínáme tělo dokumentu a vysázíme první slajd.

```

8  \usepackage[screen, panelright, blue, sectionbreak]{pdfscreen}
9  \margins{.75in}{.75in}{.75in}{.75in}
10 \screensize{5.25in}{7in}
11
12 \title{Velká firma, a.~s.}
13 \author{Polášek Michal}
14 \date{22.~10.~2007}
15
16 \begin{document}
17 \maketitle

```

Nu, a nezbyvá než začít sázet jednotlivé slajdy oddělované například příkazem \section. Zde se již postupuje stejně jako v předchozím balíčku a není třeba dalších komentářů.

```

18 \section{Charakteristika firmy}
19      \begin{tabular}{ll}
20          Název:&Velká firma, a.~s.\\
21          Základní kapitál:&2.000.000~Kč\\
22          Počet zaměstnanců:&773\\
23          Sídlo:&Adresa 1, Olomouc -- Přívoz

```

```

24          Provoz:&Adresa 2, Přerov\\
25      \end{tabular}
26  \begin{itemize}
27      \item Koupě zboží za účelem jeho dalšího prodeje a~prodej.
28      \item Poskytování vzdělávacích aktivit.
29      \item Zprostředkovatelská činnost v~oblasti obchodu a~služeb.
30      \item Pronájem movitých věcí a~nemovitostí.
31      \item Zpracování gumárenských směsí.
32  \end{itemize}

```

Další slajdy jsou vypuštěny. I zde můžeme vidět přechody na další stránku pomocí příkazu `\pagebreak`.

```

33  %Další slajdy vypuštěny.
34  \pagebreak
35  \section{Kritické faktory úspěšnosti firmy}
36      \begin{itemize}
37          \item Dobré dodavatelsko—odběratelské vztahy.
38          \item Výrobní pružnost.
39          \begin{itemize}
40              \item Využití kapacit.
41              \item Pojištění vstupů (alternativní dodavatelé).
42          \end{itemize}
43          \item Technické a~servisní zajištění .
44          \item Dopravní dostupnost.
45      \end{itemize}
46  \end{document}

```

Výslednou PDF prezentaci můžete srovnat s předchozí ukázkou. V principu se vše liší jen nastavením globálních proměnných a specifických příkazů jednotlivých balíčků, které jsou popsány v dokumentaci.

4. Beamer

Nebudeme se bát a zkusíme naši prezentaci zrealizovat i pomocí třídy Beamer. Hlavička $\text{\LaTeX}$ dokumentu vypadá například takto.

Na řádce dvě zavádíme novou třídu dokumentu, na řádcích 3 až 6 pak vzhled prezentace. Následují obvyklé balíčky a identifikační údaje.

```

1  %pdflatex beamer—prezentace.tex
2  \documentclass{beamer}
3  \mode<presentation> {
4      \usetheme{Warsaw}
5      \setbeamercovered{transparent}
6  }
7
8  \usepackage[czech]{babel}
9  \usepackage[cp1250]{inputenc}
10 \usepackage{times}

```

```

11 \usepackage[IL2]{fontenc}
12
13 \title{Velká firma, a.~s.}
14 \author{Polášek Michal}
15 \date{22.~10.~2007}

```

Slajdy se generují přes prostředí **frame**. První slajd se zavádí na řádku 18.

```

16 \begin{document}
17 \begin{frame}
18 \titlepage
19 \end{frame}

```

Druhý slajd s obsahem přednášky nebo prezentace zavedeme na řádcích 20 až 23. Za pozornost stojí příkaz **\frametitle**, který umístí popis v horní části pod klikatelnou Osnovu, kterou jsme neměli důvod vypínat.

```

20 \begin{frame}
21 \frametitle{Obsah}
22 \tableofcontents
23 \end{frame}

```

Slajd číslo tři vynecháme a podíváme se rovnou na slajd číslo čtyři. Opět užíváme úrovně nadpisů. Navíc operujeme s příkazem **\pause**, řádek 31, který umožňuje postupné odkrývání částí jednoho slajdu. Často se také hodí příkaz **\alert**, který zvýrazní určitou partii, to můžeme vidět na řádku 37.

Velmi příjemné z pohledu čtenáře je umísťování textů do bloků přes prostředí **block**. První blok je na řádcích 28 až 30, druhý pak na řádcích 32 až 39.

```

24 \section{Základní poslání}
25 \begin{frame}
26 \frametitle{Filozofie podnikání a~cíle firmy}
27 \pause
28 \begin{block}{Filozofie}
29 Řešení pro Vás. Kvalitně, rychle a~včas.
30 \end{block}
31 \pause
32 \begin{block}{Hlavní cíle firmy}
33 \begin{itemize}
34 \item Nabízet zajímavé výrobní portfolio.
35 \item Udržovat kvalitu produktů.
36 \item Zhodnocování kapitálu podniku a~tržní expanze.
37 \item \alert{JIT} koncepce ve výrobě, skladování a~dopravě.
38 \end{itemize}
39 \end{block}
40 \end{frame}

```

Naši prezentaci můžeme ukončit poděkováním na slajdu posledním. Z pohledu základního překlopení nebylo nalezeno výrazných obtíží.

```

41 % Střední část slajdů vypuštěna.
42 \begin{frame}

```

```

43 \begin{center}
44 \alert{Děkuji za pozornost\ldots}
45 \end{center}
46 \end{frame}
47 \end{document}

```

Ještě jedna poznámka. Ve spodní části vpravo výsledného PDF nepřehlédneme navigaci, v úplně spodní části pak autora a nadpis přednášky, v našem případě simulovaný název firmy.

A pamatujte! Nepamatujete-li si, že v Adobe Readeru je plná obrazovka přes Ctrl+L nebo z menu View–Full Screen Mode, tak aplikujte v preambuli `\hypersetup{pdfpagemode=FullScreen}`.

5. Beamer naostro

Síla Beameru je však v tom, že disponuje obrovským potenciálem nástrojů pro tvůrce prezentace nebo přednášky, snad ze všech možných i nemožných pohledů. Dokumentace se zdá naprosto vyčerpávající.

Pokusíme se na některé další příkazy a obecné *finty* poukázat v této kapitole, kde byl Beamer použit při ostré prezentaci, která už měla vypadat i trochu na úrovni.

5.1. Poznámky ke grafice

Postupem času se nám ukazuje, že fotky a obrázky je dobré, více či méně, umělecky zkrášlit po obdélníkových okrajích, nejen tedy přes „suchý“ `\framebox`, bez nutnosti nastavovat transparentnost obrázku. Více potenciál GIMPu a jiných bitmapových editorů.

U vektorové grafiky je transparentnost zajištěna, což je situace ideální. Toho na zahřátí poslední kapitoly využijeme. Připravíme si jednoduché postscriptové obrázky pomocí METAPOSTu.

Vytvoříme soubor `obdel.mp`. Na řádku 1 a 2 si zavedeme jednotkový rozměr a tloušťku čáry, obojí lze operativně měnit a škálovat.

```

1 u:=1cm;
2 tl:=3bp;

```

Za zajímavé považujeme přípravu skupiny (group) uvnitř METAPOSTu. Zde ji zavádíme na řádku 4 a končíme s definicí na řádku 9. Smyslem je připravit grafický objekt, který se vyskytne čtyřikrát v upravené variantě.

```

3 def zaklad =
4   begingroup;
5   draw (0u,0u)––(4u,0u)––(4u,2u)––(0u,2u)––cycle
6   withpen pencircle scaled tl;

```

```

7   draw (2u,0u)--(2u,2u) withpen pencircle scaled tl;
8   draw (0u,1u)--(4u,1u) withpen pencircle scaled tl;
9   endgroup;
10  enddef;

```

V další části si přesně definujeme výsledné objekty, které jen zobrazují diskutovanou partii prvku tabulky dva krát dva. Při troše přemýšlení a nastavení podmínek a proměnné uvnitř `\beginfig()`, by šel následující kód ještě více zautomatizovat. Ale není toho třeba.

```

11  beginfig(1);
12    zaklad;
13    fill (0u,1u)--(2u,1u)--(2u,2u)--(0u,2u)--cycle withcolor black;
14  endfig;
15
16  beginfig(2);
17    zaklad;
18    fill (2u,1u)--(4u,1u)--(4u,2u)--(2u,2u)--cycle withcolor black;
19  endfig;
20
21  beginfig(3);
22    zaklad;
23    fill (0u,0u)--(2u,0u)--(2u,1u)--(0u,1u)--cycle withcolor black;
24  endfig;
25
26  beginfig(4);
27    zaklad;
28    fill (2u,0u)--(4u,0u)--(4u,1u)--(2u,1u)--cycle withcolor black;
29  endfig;
30  end

```

Přes příkaz `mpost obdel.mp` získáme čtyři postscriptové soubory `obdel.1`, `obdel.2`, `obdel.3` a `obdel.4`.

Je to takové zpestření v budoucí prezentaci. Výsledek vypadá následujícím způsobem, vektorové a dostatečně průhledné.

5.2. Prezentace

Přejdeme k Beamerování. K tomu, co máte tak rádi!

Poněvadž zdrojový kód souboru `teze-phd.tex`, cca dvacetiminutového úvodního slova, má více než 20 KB, tak opravdu jen ty nejdůležitější nové momenty.

První upozornění je na parametr `handout` v třídě dokumentu. Pomocí něj lze přepínat mezi obrazovou a tištěnou verzí prezentace. Handout, ve smyslu poznámek, je věc užitečná, když chceme mít poznámky a komentáře právě u daného

slajdu. Takové poznámky si lze vytisknout a používat během prezentace. Je to minimálně slušná pomůcka pro zapomnětlivé. *{Jeden z autorů se hlásí...}*

```
1 %pdflatex teze-phd.tex
2 \documentclass[11pt]{beamer}
3 %\documentclass[11pt,handout]{beamer}
```

Nastavíme ještě jednou standardní balíčky, nyní na řádcích 4 až 7.

Můžeme se vrhnout do nastavení stylu, to můžeme vyčíst z řádků 8 až 12. Manuál je z tohoto pohledu neocenitelný.

Prvky navigace lze operativně vypnout, řádek 13. Během prezentace se lze rychle pohybovat klávesami nebo klikáním na Osnovu. Navigační prvky lze z tohoto pohledu považovat za přepych.

Bez výraznějšího omezení ze strany Beameru si definujeme nové příkazy, nyní na řádcích 15 a 16.

```
4 \usepackage[cp1250]{inputenc}
5 \usepackage[T1]{fontenc}
6 \usepackage[english,czech]{babel}
7 \usepackage{times}
8
9 \usetheme{Darmstadt} % Upraveno o subsection.
10 \usefonttheme{structurebold}
11 \setbeamercovered{transparent=5}
12 \setbeamercovered{dynamic}
13 \setbeamertemplate{navigation symbols}{}
14
15 \def\sk{\hspace*{1.8em}}
16 \newcommand{\DDP}[1]{\color{yellow}viz DDP, str.~\#1.}
```

Poté si nadefinujeme úvodní stránku.

```
17 \title{{\normalsize Mapování a-efektivní řešení marketingově-informatických
18   úkolů nastávající společnosti založené na znalostech}}
19 \subtitle{{\small\ \ [-3pt] Podnázev:\Prínosy matematiky, pravděpodobnosti
20   \& statistiky, informatiky a-typografie skloubené v-jeden celek}}
21 \author{Ing. Pavel \textsc{Stríž}}
22 \institute[Univerzita Tomáše Bati ve Zlíně]{%
23   Fakulta managementu a-ekonomiky\
24   Univerzita Tomáše Bati ve Zlíně}
25 \date{24. dubna 2007}
```

Nastavíme vzhled osnov kapitol, jedna z možností je prezentována na řádcích 26 a 27. Můžeme zahájit dokument, řádek 30, nastavit užívaný jazyk, řádek 31, a nechat vygenerovat první slajd na řádku 33.

```
26 \AtBeginSection[]{\frame{\frametitle{Podstatné body obhajoby}
27   \tableofcontents[current]}}
28
29 %%%Začátek prezentace.
30 \begin{document}
```

```

31 \language1
32
33 \frame{\section*{OBHAJOBA DISERTAČNÍ PRÁCE}\titlepage}

```

Vedle prostředí **block** (v naší ukázce v modré) je výhodné používat prostředí **alertblock** (červené) a **exampleblock** (zelené). Ukázkové zdrojové kódy lze prostudovat níže.

```

34 \part{Část nultá}
35 \frame{\frametitle{Podstatné body obhajoby}\tableofcontents[part=1]
36 \begin{exampleblock}{Předaná ukázka}
37     Tato prezentace pro účely navazujících diskuzí.
38 \end{exampleblock}
39 }
40
41 \section[Idea práce \{1/3\}]{Informační a~znalostní společnost a~idea práce}
42 \subsection*{}
43 \frame{\frametitle{Informační společnost}
44 \begin{alertblock}{Definice pojmu}
45     {\bf Informační společnost} je charakterizována podstatným:
46 \begin{itemize}
47     \item využíváním digitálního zpracovávání,
48     \item uchovávání a
49     \item přenosu informací.
50 \end{itemize}
51 \end{alertblock}
52 \vfill
53 }

```

Při komplikovanějším a dynamickém výkladu nemusí být vždy příkaz `\pause` dostatečným. Využívají se příkazy jako `\only`, `\alert`, `\onslide`, `\uncover` a další.

Zde je jedna ukázka, kdy jsme si přáli skrýt černou tečku na konci výběru a nahradit ji výplní v barvě červené (`alert`). Zdrojový kód vypadá takto.

```

54 \newcommand{\ukazz}{\only<1>{.}\only<2->{\alert<2->
55     {\dotfill Záběr práce!}}}
56
57 \frame{\frametitle{Klasifikace informatika}
58 \begin{block}{Hlavní kategorie problémů, \DDP{93}}
59 \begin{itemize}
60     \item Zpracování číselných údajů\ukazz
61     \item Zpracování textových řetězců\ukazz
62     \item Práce s~grafikou (bitmapa a~vektor)\ukazz
63     \item Analýza zvukových stop.
64     \item Analýza video sekvencí.
65     \item Obecně cokoliv v~elektronické podobě.
66 \end{itemize}
67 \end{block}
68 }

```

Po jisté době člověk ocení parametrizaci i běžných příkazů výčtů, např. `\item<3->`. Kdy se daná položka výčtu zobrazí na třetím a všech ostatních vrstvách jednoho slajdu.

Pokud bychom si přáli umístit předpřipravený vektorový obrázek `obdel.1`, zde je jednoduchý postup užitím `\convertMPtoPDF`.

```

69 \setbeamercovered{transparent=50}
70 \frame[t]{\frametitle{Blok I~\hfill\convertMPtoPDF{obdel.1}{.2}{.2}}
71 \begin{block}{Výzkum trhu: tištěná podoba, \DDP{135\,--\,156}}
72 \begin{itemize}
73   \item Výpočet obtížnosti textů.
74   \item Autorské archy a-honoráře (ukázkově řešeno).
75   \item Analýza bitmapových obrázků.
76 \end{itemize}
77 \end{block}
78 }

```

Skončíme poukazování na momenty Beameru závěrečným slajdem upozorňující na to, v čem zrovna prezentace byla vyhotovena.

```

79 \section*{Závěrem}
80 \frame{\vfill
81 \begin{alertblock}{Závěrem Úvodního slova disertanta}
82 \par\hfill
83 \textbf{Pro tuto chvíli vám děkuji za pozornost!}\[-6pt]\
84 \end{alertblock}
85
86 \vfill
87 \begin{center}
88 \alert{\small{Obhajoba byla napsána v-sázecím systému \TeX\par
89 ve formátu \textsc{pdf}\LaTeXe\ v-třídě Beamer.}}
90 \end{center}
91 }

```

Dobrovolně se přiznáme, že užít zvětšování obrázku v prezentaci, ostřeji zasahovat do designu, vkládat zvukové nebo animované sekvence, zatím nebylo třeba. Pokud by to však třeba bylo, tak Beamer umí!

5.3. Slajdy k tisku

Nepočítáme-li nekonečné množství příloh (katalogy, ceníky) a vzorků (vinařské, pekařské a takové), které lze připravit a nechat kolovat u obhajob vysokoškolských kvalifikačních prací, tak se zaměříme na tisk vybraných slajdů.

Přímo Beamer doporučuje balíček `pgfpages` při změně stylu či „show only notes“ režimu. Tisk poznámek čtenář jistě zvládne, my zkusíme tisk slajdů přes balíček `pdfpages`, už mimo Beamer. To jsou ty podklady, které lze předložit k diskuzi a není zvýšený nárok na členy komise si dělat opisky z právě promítaných slajdů. To se ukázalo užitečným. Tiskovou barvou také zbytečně nešetřit.

Ačkoliv byl výběr stránek udělán ručně, není to problémem. Výhodou zůstává, že kód Beameru nenařekujeme. Výsledek přes balíček `pdfpages`, v podobě souboru `teze-phd-tisk.tex`, může vypadat takto.

```

1 %pdflatex teze-phd-tisk.tex
2 \documentclass{article}
3 \usepackage{pdfpages}
4 \begin{document}
5 \fboxrule=2pt
6 \includepdf[pages={1-4,7-9,12-15,18,19,21,24,27-39,47-50},
7 nup=1x2, frame=true, delta=30 30, scale=0.85 ]{teze-phd}
8 \end{document}

```

5.4. Úvodní list k tisku

Na závěr jeden bonbónek. Po vytištění slajdů bychom mohli chtít přidat úvodní stránku se svými iniciálami a konkrétním jménem člena komise. To má velký dopad (impact) i na ty nejzatrvelejší zkoušející. Půjdeme na to přes, hádejte, ano, přes $\TeX$.

Pro zkušenější: je to v podstatě přístup jako přes hromadnou korespondenci, s tím, že na to máme nyní jen jeden soubor, konkrétně `uvod.tex`.

Preamble dokumentu nechť je:

```

1 \documentclass[a4paper]{article}
2 \usepackage[cp1250]{inputenc}
3 \usepackage[IL2]{fontenc}
4 \usepackage[czech]{babel}
5 \pagestyle{empty}
6
7 \newcommand\sazej[2]{\vspace*{5cm}
8 \begin{center}
9 {\LARGE OBHAJOBA DISERTAČNÍ PRÁCE}\vspace{1cm}\par
10 {\LARGE Ing. Pavel \textsc{Stráž}}
11 \end{center}
12 \vfill\large{\bf Předáno:\ }#1\par\noindent#2\vspace{2cm}\newpage
13 } %Konec příkazu \sazej.

```

Zavedli jsme si běžné balíčky a připravili nový příkaz `\sazej`, který dělá to, že připraví jednu stranu A4 s příslušným jménem, titulaturou a funkcí, vždy pro jednoho konkrétního člověka v komisi.

Vyvolání, resp. vygenerování, potřebného počtu stran je už rutina přepisu jmen ze seznamu zkušební komise den před obhajobou kvalifikační práce.

```

14 \begin{document}
15 \sazej{prof. Ing. Křestní1 \textsc{Příjmení1}, CSc.}{předseda zkušební komise}
16 \sazej{doc. Ing. Křestní2 \textsc{Příjmení2}, CSc.}{člen zkušební komise}
17 \sazej{prof. Ing. Křestní3 \textsc{Příjmení3}, CSc.}{člen zkušební komise}
18 \sazej{prof. Ing. Křestní6 \textsc{Příjmení6}, Ph.D.}{člen zkušební komise}

```

```

19 \sazej{prof. Ing. Křestní7 \textsc{Příjmení7}, DrSc.}{oponent disertační práce}
20 \sazej{doc. RNDr. Křestní8 \textsc{Příjmení8}, CSc.}{oponent disertační práce}
21 \sazej{doc. Ing. Křestní9 \textsc{Příjmení9}, CSc.}{školitel doktoranda}
22 \sazej{Paní Křestní10 Příjmení10}{tajemnice}
23 \sazej{Host obhajoby disertační práce}{}
24 \end{document}

```

Už jen hodit do spirál, spirály hodit do tašky, tašku hodit do kufru auta a autem se včas dostat do právě té jedné vyhlášené místnosti vaší *Alma mater*.

A v takovém případě nelze než popřát: *Kolegové, hodně zdaru!*

Summary: Examples of Presentations

In the first part, the article presents a simple presentation created in PDFSlide and PDFScreen packages and in the Beamer class. In the second part, the article discusses the Beamer class in a real-world presentation in more detail. It starts with the creation of METAPOST graphics. After that it comments on some settings of the Beamer design, generating a title page and a section table of contents. Next follows an example of `\alert` and `\convertMPtoPDF` commands. The generated output is a PDF file. This file is converted for printing purposes using a `pdfpages` package. At the conclusion of the article a preparation of individually named title pages are prepared with printed materials which are to be given to the members of the committee, e.g. before a theses defense at a university.

Key words: Presentation Examples, PDFSlide, PDFScreen, Beamer, METAPOST, PDFPages, Individual Title Pages.

Pavel Stríž, ÚIS FaME UTB ve Zlíně
striz@fame.utb.cz

Michal Polášek, FaME UTB ve Zlíně
michal.igi@seznam.cz

Interaktivní 3D grafika v HTML a PDF dokumentech

ROMAN PLCH, PETRA ŠARMANOVÁ

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

1. Úvod

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

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

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

2. Interaktivní grafika v HTML dokumentu

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

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

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

2.1. Mapleovská knihovna JavaViewLib

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

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

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

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

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

Knihovnu načteme příkazem:

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

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

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

¹Nastavení cesty závisí na použitém operačním systému.

2.2. Export grafů

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

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

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

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

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

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

Obrázek 1: Export do MPL

Obrázek 2: Export do JVX

Příkaz

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

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

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

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

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

2.3. Zobrazení grafů pomocí JavaView

Příkaz

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

spustí program JavaView a zobrazí v něm zadaný graf funkce.

Příkaz

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

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

2.4. Ostatní příkazy

Seznam parametrů a jejich hodnot získáme příkazem:

```
> getInfo();
```

JavaViewLib State Information

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

Obrázek 3: Ukázka začlenění interaktivní grafiky do HTML dokumentu

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

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

Vypsání hodnoty konkrétního parametru dosáhneme příkazem:

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

Nastavení provedeme pomocí příkazu set:

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

Příkazem

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

nastavíme znovu původní parametry.

Příkaz

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

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

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

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

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

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

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

2.5. Presentace interaktivní grafiky na webu

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

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

Dále uvádíme příklad volání appletu v HTML dokumentu:

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

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

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

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

3. Interaktivní grafika v PDF dokumentu

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

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

²Mezinárodní standardizační instituce.

3.1. Vytvoření 3D objektu ve formátu U3D

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

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

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

Obrázek 4: Grafický objekt ve formátu U3D

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

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

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

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

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

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

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

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

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

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

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

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

3.2. Úskalí exportu z Maplu do VRML

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

Příkazy

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

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

Obrázek 5: Ukázka exportu os z Maplu do VRML

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

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

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

3.3. Vložení grafického objektu ve formátu U3D do PDF dokumentu

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

Obrázek 6: Funkce a osy vytvořené procedurou `Plotaxes`

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

Balíček načteme ve zdrojovém souboru příkazem

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

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

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

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

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

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

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

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

3.4. Optimální umístění objektu na scéně

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.5. Vytvoření externího souboru s různými pohledy na scénu

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

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

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

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

3.6. Zobrazení různých pohledů na scénu

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

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

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

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

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

Podívejte se na danou funkci z~různých pohledů:\\

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

{Pohled shora}.

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

{Pohled zespodu}.

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

Dále uvádíme zdrojový text k ukázce na obrázku 8.

```
\includemovie[toolbar,label=pict1,3Drender=SolidWireframe,  
3Dcoo=0.42 0.43 0.17, 3Droo=5.5, 3Dc2c=0.73 0.40 0.53,  
3Droll=-0.32, 3Dviews=view.txt, 3Dlights=Headlamp]  
{0.6\linewidth}{0.6\linewidth}{p1_os3.u3d}
```

Podívejte se na danou funkci z~různých pohledů --

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

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

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

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

Obrázek 8: Hypertextové odkazy na předdefinované pohledy

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

4. Závěr

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

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

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

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

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

Reference

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

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

Summary: Interactive 3D Graphics in HTML and PDF Documents

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

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

1. Úvod

Zkratka AWK znamená Aho, Weinberger, Kernighan a označuje poněkud svérázný programovací jazyk navržený pro zpracovávání textových souborů.

První verze tohoto programovacího jazyka vznikla již v roce 1977. V roce 1985 byl tento programovací jazyk dále rozšířen [?] a tato nová implementace byla vydána pod volnou licencí v roce 1996. Další informace lze nalézt na stránce Briana Kernighana: <http://cm.bell-labs.com/cm/cs/awkbook/>. Za pozornost stojí i kniha [?].

GNU awk (nebo gawk) je jiná implementace jazyka AWK, která je (téměř jistě) obsažená v každé distribuci Linuxu. Další distribuce existují pod názvy xgawk, mawk nebo TAWK (verze pro DOS a Windows) i GNU AWK (OS/2).

2. Příklad

Princip jazyka AWK si ukážeme na jednoduchém příkladě. Předpokládejme, že výsledky rozsáhlé simulační studie jsou uloženy v následujícím značně nepřehledném textovém souboru `priklad.txt`:

```
TABULKA.AA "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
"1" 1 2.0599209764691 1.1747333245979 1.03814385890582 2.0599209746917
1.10206315043 27 1.00304826425488 1.16483656626828 1.04785974714491
1.0215094731292 1.1656626828 1.03386899720460 1.00215433687505
"2" 1 1.2111642320962 1.1785509411592 1.09217031887314 1.21116423209622
1.168397750859 1.0225654118313 1.1664016471655 1.05402221156164
1.00649949971005 1.1664064716555 1.02731120933558 1.00184132881134
"3" 1 1.3220358323917 1.0628344035499 1.01922692920646 1.32203958323917
1.052451742835 1.00663451863578 1.0622641328610 1.0229410009790
1.01245982055420 1.0622661328608 1.02234509043763 1.00537251503339
"4" 1 3.0406725862321 1.12150481270856 1.0758055731613 3.04367252319
1.1156698237625 58 1.00211380906327 1.2487169003298 1.102381764515
1.01161646944822 1.24871679003298 1.10077794969210 1.00206064501333
```

Programovací jazyk AWK umožňuje tento textový výstup jednoduše přeformátovat do podoby L^AT_EXové tabulky. Následující postup funguje se standardní Linuxovou implementací `gawk`.

Nejprve vytvoříme soubor `tabulka.awk`, který obsahuje všechny potřebné instrukce:

```

BEGIN{
    getline;
    print "\\begin{tabular*}%";
    print "{\\textwidth}{1|*{" ,NF-1,"}{@{\\extracolsep\\fill}r}}";
    row=1
}

{
    printf "%.0f",row;
    for (i=2;i<=NF/2;i++) {
        printf "&%.1f",$(i+NF/2);
        printf "&{\\scriptsize (%.3f)}", $i;
    }
    print "\\\\";
    row=row+1
}

END{
    print "\\end{tabular*}";
}

```

Úvodní část programu, za příkazem **BEGIN**, obsahuje příkazy, které se provedou pouze na počátku. Zde vytiskneme hlavičku naší tabulky. Proměnná **NF** zde vytiskne počet záznamů na prvním řádku vstupního textového souboru (musíme odečíst 1, protože výsledná tabulka bude mít o jeden sloupeček méně). Vytvoříme proměnnou **row**, která bude obsahovat číslo řádku. Příkaz **getline** „přeskočí“ nezajímavý první řádek.

Hlavní část programu nalezneme v dalších složených závorkách. Tyto příkazy se vykonají postupně pro každý řádek vstupního textového souboru. Nejdříve vytiskneme číslo řádku a pak postupně tiskneme čísla z druhé poloviny textového souboru (velkým písmem a s přesností na jedno desetinné místo) a za ně vždy příslušná čísla z první poloviny textového souboru (menším písmem, v závorce a s přesností na tři desetinná místa). Vše doplňujeme potřebnými oddělovači a na konec každého řádku nezapomeneme vytisknout dvě obrácená lomítka.

Ve třetí části programu, za příkazem **END**, již vytiskneme **\end{tabular*}** a ukončíme tím tabulku.

Z příkazové řádky spustíme náš program:

```
gawk -f tabulka.awk < priklad.txt > priklad.tex
```

Získáme pěkně zformátovanou L^AT_EXovou tabulku v souboru **priklad.tex**, který obratem vložíme do našeho příspěvku pomocí **\input{priklad.tex}** příkazu:

```

\begin{table}
\input{priklad.tex}
\caption{Pěkně zformátovaná tabulka.}
\end{table}

```


1	1.2	(1.000)	1.0	(2.060)	1.0	(1.175)	1.2	(1.038)	1.0	(2.060)	1.0	(1.130)
2	1.2	(1.000)	1.1	(1.211)	1.0	(1.179)	1.2	(1.092)	1.0	(1.211)	1.0	(1.162)
3	1.1	(1.000)	1.0	(1.322)	1.0	(1.063)	1.1	(1.019)	1.0	(1.322)	1.0	(1.058)
4	1.2	(1.000)	1.1	(3.040)	1.0	(1.122)	1.2	(1.018)	1.1	(3.040)	1.0	(1.110)

Tabulka 1: Pěkně zformátovaná tabulka.

Výsledek našeho snažení se vzápětí objeví v pěkně zformátované tabulce 1.

Hlavní výhoda tohoto postupu je fakt, že pro změnu ve formátování tabulky 1 stačí tuto změnu provést pouze na jednom místě v `tabulka.awk`. Naprosto triviální je například změna počtu desetinných míst pro všechna čísla v tabulce 1.

Celý postup lze samozřejmě ještě více automatizovat a zjednodušit. Například, program `gawk` můžeme zavolat přímo z našeho statistického software už při ukládání výsledků. Pokud jsme nedočkaví, tak můžeme tabulku zároveň i přeložit `LATEX`em do PDF nebo PS. Lze si představit i jiné použití v kombinaci s příkazem `Sweave()` ve statistickém programovacím prostředí `R` dostupném na: <http://www.r-project.org/>. Pro RedHat systémy najdete RPM na serveru <http://fedoraproject.org/wiki/EPEL>.

3. Závěr

Je zřejmé, že `AWK` můžeme využít i k jiným, náročnějším úkolům. Díky své rychlosti se hodí k provádění jednoduchých operací na rozsáhlých datových souborech, k automatizaci psaní opakujících se počítačových programů, nebo k získávání potřebných informací z jasně zformátovaných rozsáhlých `LATEX`ových souborů. Jako příklad si lze představit automatické odesílání emailů autorům příspěvků v tomto sborníku, vždy s automaticky vyplněnou správnou emailovou adresou, automaticky vyplněným správným jménem autora a automaticky vyplněným správným názvem příspěvku.

Fantazii se zkrátka žádné meze nekladou.

Summary: Large-scale Production of Tables in AWK

This short article demonstrates the capabilities of `AWK` when producing `LATEX` tables and formatting them in large quantities.

Key words: `AWK`, Large-scale Table Production, Table Formatting.

Zdeněk Hlávka, KPMS MFF UK Praha
hlavka@karlin.mff.cuni.cz

Konference: T_EXperience 2008

PRVNÍ OZNÁMENÍ A INFORMACE

Vážené kolegyně, vážení kolegové,

pořadatelé, Československé sdružení uživatelů T_EXu a Fakulta managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně, Vás zvou na 1. ročník odborné konference T_EXperience 2008.

Ta se uskuteční o víkendu, **26. – 28. září 2008** (příjezd ve čtvrtek 25. 9. je možný), na Rusavě, okr. Kroměříž. Akce bude probíhat na turistické chatě, chatkách a ve srubech na jestrabi.cz.

Motto akce je: „Vyměňte si své zkušenosti s T_EXem neformálně mezi přáteli!“

Výzva: dovezte své i cizí počiny (knihy, skripta, články, ...) dělané pomocí T_EXu a příbuzných programů, uděláme hromadnou výstavku. Ze šuplíků s počiny ven!

Zvané přednášky přijali:

- Karel Píška: Fonty a jak s nimi pracují (plain)T_EX, L^AT_EX, pdfT_EX, X_YT_EX a LuaT_EX.
- Karel Horák: Typografie a nová písma v mezinárodních projektech.
- Pohovoří i Zdeněk Wagner, který téma pečlivě rozvažuje.
- Vít Zýka ukáže značkování dokumentu a praktické partie z ConT_EXtu.
- Petr Sojka: Jak jsem sázel knihu Knuthovu.

Páteční večer bude věnován Petru Olšákovi s tutorialem „Úvod do makrojazyka T_EXu“.

Sobotní večer je na programu panelová diskuze s těmi T_EX zdatnějšími na téma zkušenosti a užití T_EXu v osobním i pracovním životě, vyřešené partie a problémy řešené. Prostor však bude na prvním místě dán otázkám z pléna.

Příspěvky dalších přednášejících budou upřesněny včas v Programu konference.

Pouštířky libovolné velikosti lze dovést, můžeme vylepit po celé budově, začneme okny.

Sponzoři vítáni. Kontaktujte prosím přímo organizátory. *Studenti* taktéž vítáni.

Termín vyplnění přihlášky na cstug.cz, uhrazení poplatku na bankovní účet ζ TUGu a odeslání příspěvku v šabloně Zpravodaje na bulletin@cstug.cz (ano, určitě, i anglický abstrakt připsat) je jednotný: **10. srpna 2008**.

– Pozdní uhrazení by snad bylo možné řešit individuálně. Budte však k organizátorům šetrní.

– Pozdní přihlášení je snad možné řešit s organizátory na základě aktuální kapacity.

– Pozdní odeslání příspěvku znamená, že nemůžeme slíbit otištění v plánovaném sborníku (nakladatelství Konvoj). Můžeme po termínu uvažovat o Zpravodaji ζ TUGu.

Cena: 1550 Kč (1800 Kč příjezd ve čtvrtek), doprovodná osoba dospělá 1000 Kč (1250 Kč i se čtvrtkem) a doprovodná osoba dítě 850 Kč (1100 Kč i se čtvrtkem). Zvířectvo zadarmo.

Technika: dataprojektor, Internet snad také bude. Akce bude pravděpodobně natáčena.

Doprava: autem (parkovat lze). Či autobusem, z Rusavy-města ještě několik kilometrů pěšky.

Ubytování: dvou-, tří- a čtyřlůžkové. Nebo ve vlastním stanu. :-)

Doprovodné osoby: ano, je možné, i dospělé i děti, i psy i kočky.

Toalety a sprchy na patře (hlavní budova) nebo v blízké samostatné budově (ostatní).

Strava: domácí. Druhé večere budou jako překvapení, o kterém mlčíme jako partyzáni.

Počasí je domluvené. Teplé oblečení sebou! Táboráček a posezení u ohně. Sběr hub možný.

Dobrovolné povinný výlet: na Svatý Hostýn a přilehlou Rozhlednu na Hostýnském vrchu.

Organizátoři akce: Zdeněk Wagner <bulletin@cstug.cz>, Pavel Stríž <striz@fame.utb.cz>, fotka na: <<http://studium.uis.fame.utb.cz/>>, telefon: +420 603 380 402, na vlastní akci také +420 724 009 340 (s dopravou a nouzovými stavy rád pomůže táta, Milan Stríž) a Vít Zýka.

Zpravodaj Československého sdružení uživatelů T_EXu
ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze)

Vydalo: Československé sdružení uživatelů T_EXu
vlastním nákladem jako interní publikaci
Počet výtisků: 500
Uzávěrka: 20. 5. 2008
Odpovědný redaktor: Zdeněk Wagner
Redakční rada: Pavel Stríž, David Catto a výbor CSTUGu
Tisk a distribuce: KONVOJ, spol. s r. o., Berkova 22, 612 00 Brno,
tel. +420 549 240 233
Adresa: CSTUG, c/o FEL ČVUT, Technická 2, 166 27 Praha 6
Tel: +420 224 353 611
Fax: +420 233 332 938
Email: cstug@cstug.cz

Zřízené poštovní aliasy sdružení CSTUG:

bulletin@cstug.cz, zpravodaj@cstug.cz
korespondence ohledně Zpravodaje sdružení

board@cstug.cz
korespondence členům výboru

cstug@cstug.cz, president@cstug.cz
korespondence předsedovi sdružení

gacstug@cstug.cz
grantová agentura CSTUGu

secretary@cstug.cz, orders@cstug.cz
korespondence administrativní síle sdružení, objednávky CD-ROM

cstug-members@cstug.cz
korespondence členům sdružení

cstug-faq@cstug.cz
řešené otázky s odpověďmi navrhované k zařazení do dokumentu CSTFAQ

bookorders@cstug.cz
objednávky tištěné T_EXové literatury na dobírku

ftp server sdružení:
<ftp://ftp.cstug.cz>

www server sdružení:
<http://www.cstug.cz>