5. ročník
Žilina 2.- 4. júla 2013
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
ossconf.soit.sk
OSSConf

OSSConf 2013: 87–94

CYKLY VO FORMÁTE LaTeX

RUDOLF BLAŠKO (SK)

Abstrakt. Typografický systém TeXa špeciálne LaTeXmá svoje nepopierateľné výhody a ostatné systémy na písanie dokumentov (komerčný `Microsoft Office`, resp. voľne šíriteľné `OpenOffice.org`, `LibreOffice`) mu len ťažko môžu konkurovať. Samozrejme aj tieto systémy majú svoje výhody, ale sú orientované skôr ako sekretárske aplikácie a nie sú príliš vhodné na písanie vedeckých a odborných publikácií, ktoré obsahujú špeciálne vzorce (napr. matematické, chemické, ...). Tieto vzorce sa tam dajú vytvoriť. Ale vynaložené úsilie nezodpovedá výslednej kvalite. V predloženom príspevku ukážeme postupy ako v LaTeX-u vytvoriť vlastné príkazy a následne zautomatizovať ich hromadné použitie pomocou príkazov na tvorenie cyklov (napr. `\foreach`, `\multido`).

Kľúčové slová. LaTeX, otvorený softvér, cyklus, `foreach`, `multido`.

CYCLES IN THE LaTeX

Abstract. Typesetting system TeXand especially LaTeXhas its undeniable advantages and other systems for documents preparation (e.g. commercial `Microsoft Office`, or freely available `OpenOffice.org` or `LibreOffice`) can hardly compete it. Of course, these systems have their advantages too, but are oriented rather to office application and are not well suited for writing scientific and technical publications that include special formulas (such as mathematical, chemical, ...). These formulas can be created even in these systems. But the effort does not correspond to the resulting quality. In the present paper we demonstrate the processes of custom commands creation in LaTeXand then automate their mass use by cycle commands (e.g. `\foreach`,`\multido`|.

Key words and phrases. LaTeX, open source, cycle, `foreach`, `multido`.

Úvod

V súčasnej dobe stále drvivá väčšina užívateľom používa komerčné a, uvedomelejší užívatelia, aj nekomerčné kancelárské programy (`Microsoft Office`, voľne šíriteľné `OpenOffice.org`, `LibreOffice`). Keďže principiálne je medzi nimi iba jediný rozdiel, a to je platenie za ich používanie, budem tieto systémy spoločne označovať `Office`. Veľkou výhodou týchto programov je veľká otvorenosť a užívateľská prívetivosť týchto programov: „Otvorím nový dokument a hneď môžem začať písať. Keď chcem niečo zmeniť iba si to jednoducho myškou naklikám.“. A v tom je väčšinou problém, málo používané funkcionality programu musí aj skúsený užívateľ hľadať. Toto hľadanie nemusí byť triviálne, najmä ak človek momentálne používa inú verziu od tej, na ktorú je zvyknutý. Keďže každý používa tieto programy systémom otvorím a píšem, skoro nikto neovláda ich zákonitosti a

väzby. Potom nastávajú situácie, že tlač vyzerá ináč ako text na monitore, miešajú a objavujú sa nechcené fonty písiem, problém je s hlavičkou a obsahom, ... Väčšina týchto problémov vyplýva z neinformovanosti užívateľa a z lákavej vidiny, že môže hneď písať a „tváriť sa ako typograf“. Tieto systémy dokážu veľa, ale keď chce niekto kvalitný výstup, potrebuje o nich niečo nemálo vedieť — a nie je to iba klikanie myškou.

Napísať jednoduchý naformátovaný text (kapitoly, odrážky, tabuľky) nie je v týchto programoch problém. Problémy nastávajú pri písaní špeciálnych výrazov (matematiké, fyzikálne, chemické a vo všeobecnosti technické vzorce). Tieto výrazy sa dajú jednoducho vytvoriť pomocou klikacích prednastavených šablón. Problém nastáva pri ich úpravach — užívateľ musí dobre ovládať myš. Nezanedbateľná je aj ich vizuálna podoba a problém zladenia rôznych fontov rovnakých premenných (napr. x_i) v texte a v uvedených výrazoch. Ale keď nie je užívateľ príliš náročný, tieto výstupy postačia.

Uvažujme hypotetickú situáciu, že organizujeme nejakú akciu (napr. konferenciu) a potrebujeme vygenerovať menovite pre väčší počet ľudí reprezentatívny dokument (napri certifikát). Uvedený problém v predchádzajúcich systémoch Office je riešiteľný pomocou systému hromadnej korešpondencie, pričom je možné prepojenie aj na externé databázy a údaje sa dajú modifikovať podľa požiadaviek (napr. filtrovanie, radenie a agregovanie vstupnej databázy). Systém, bol vytvorený pre tvorbu hromadných dopisov, adresných a iných štítkov, či vizitiek. Na tlač certifikátov, alebo iných dokumentov, v ktorých potrebujeme meniť údaje podľa vstupnej databázy, sa dá samozrejme tiež použiť, ale stále je to o klikaní. Makrá sa v MSOffice dajú tvoriť ako postupnosti klávesových príkazov a kliknutí myškou. V skutočnosti sa tieto príkazy zapisujú ako volania metód jednotlivých objektov GUI príslušného programu. Zápis prebieha priamo v skriptovacom objektovom jazyku Visual Basic. Takto uložené makrá môžeme jednoducho modifikovať a dopĺňať ich o jednoduché cykly, vlastné výpočty, ap.

V typografickom systéme LaTeX je tento problém oveľa jednoduchší. Vytvoríme si vlastný príkaz (tzv. `makro`), ktorý bude obsahovať celý certifikát a namiesto mena vložíme parameter, ktorý sa pri vlastnom spustení makra nahradí menom. Takže nepotrebujeme pre každého účastníka vytvárať celý certifikát, vytvoríme ho iba raz a LaTeX za nás meno nahradí. Zanietení labužník systému Office môže namietať, že takto to funguje aj u nich. V podstate má pravdu, ale je tu jeden „nepatrný rozdiel“. V LaTeX-u je to jednoduchšie, postačí v zdrojovom súbore napísať uvedené meno ako argument príkazu. Predpokladajme, že je certifikát definovaný v makre s názvom `\OSScert`, potom stačí napísať:

```
\OSScert{Rudolf Blaško}\OSScert{Jožko Hruška}\OSScert{Michal Starec}
....
\OSScert{Iné Meno}\OSScert{Posledné Meno}
```

Prípadne môžeme použiť cyklus. V tomto prípade sa dá s úspechom použiť cyklus `\foreach`. Jeho použitie by mohlo vyzerať napr. nasledovne:

```
\foreach \MENO in {{Rudolf Blaško}, {Jožko Hruška}, {Michal Starec},
   < ostatné mená, ... >  {Iné Meno}, {Posledné Meno}}
{\OSScert{\MENO}}
```

1. Makrá systému LaTeX

Na vytvorenie vlastného makra, t. j. vlastného príkazu sa používajú príkazy `\def`, resp. `\newcommand` a ich modifikácie. Každý z nich má svoje výhody a špecifické možnosti.

Príkaz `\def` je jedným zo základných príkazov systému TeX a používa sa v tvare `\def\názov_príkazu <maska parametrov> {telo definície}`. Príkaz má lokálny charakter, to znamená, že ak ho použijeme vo vnútri nejakého prostredia, resp. ohraničujúcich zátvoriek `{...}`, mimo sa jeho význam stratí. Tento príkaz neoveruje, či už nejaké makro alebo príkaz s identickým názvom existuje, takže v kladnom prípade ho nahradí novým makrom. V názve (analogicky ako v celom TeX-u a LaTeX-u) sa rozlišujú malé a veľké písmená.

Časť `<maska parametrov>` je nepovinná a určuje spôsob práce práve definovaného makra s parametrami. Parametrov môže byť až 9 a musia nasledovať vzostupne za sebou od `#1` až po posledný napr. `#6`. Parametre môžu byť oddelené a obložené rôznymi oddeľovačmi a maska končí prvým výskytom znaku `{`.

Pri použití makra sa požaduje zápis parametrov presne v takom istom tvare, ako je uvedený v maske parametrov. Príkaz sa môže použiť aj bez parametrov, napr. zápis `\def\Konf{KonFERerencia začína!}` definuje nový príkaz `\Konf`, ktorý je pri každom použití nahradený vetou „KonFERerencia začína!“ Ak upravíme makro na tvar `\def\Konf{\it KonFERerencia začína!}`, potom sa daná veta vypíše kurívou „*KonFERerencia začína!*“. *Problém je iba v tom, že sa kurzívou vypíše aj nasledujúci text za príkazom. Správne definované makro má mať tvar* `\def\Konf{{\it KonFERerencia začína!}}`. To znamená, že je potrebné zdvojiť zátvorky, aby sa kurzíva obmedzila iba na danú vetu, t. j. na skupinu ohraničenú vnútornými zátvorkami, vonkajšie zátvorky sú súčasťou predpísaného tvarus príkazu `\def`.

Ak chceme pridať napr. názov konferencie a dátum, môžeme makro upraviť nasledovne `\def\Konf[#1|#2]{{\it KonFERerencia #2 začína #1!}}`. Po takejto definícii musíme príkaz `\Konf` používať s maskou `[názov|dátum]`. Ak nebude za slovom `\Konf` nasledovať symbol `[` (resp. ␣`[`), LaTeX vyhlási chybové hlásenie. Po zaevidovaní tohto symbolu pokračuje LaTeX v načítavaní parametra `#1` až do prvého výskytu symbolu `|`, kedy začne načítavať parameter `#2` až do prvého výskytu symbolu `]`. Príkazom `\Konf[2.7.2013|OssConf]` dostaneme vetu „*KonFERerencia OSSConf začína 2.7.2013!*“.

Makrá môžeme do seba vnárať, t. j. v definovanom makre môžeme opäť definovať nové makro. Ak používame parametre, potom sa symbol # vo vnútornom makre zdvojuje. Musíme ale rozlišovať medzi parametrami vonkajšieho a vnútorného makra. Ukážeme to na jednoduchom príklade. Definujme makro `\TRANS`:

```
\def\TRANS#1#2#3#4{\def\ptm##1##2{##1\to##2}
  $\ptm{#1}{#2}$, $\ptm{#2}{#3}$, $\ptm{#3}{#4}$, $\ptm{#4}{#1}$}
```

Zadaním príkazu `\TRANS{\alpha}{\beta}{\gamma}{\delta}` dostaneme výpis „$\alpha \to \beta$, $\beta \to \gamma$, $\gamma \to \delta$, $\delta \to \alpha$“ v matematickom móde. Ako sme videli, jednotlivé parametre sú oddelené v zátvorkách. Teoreticky by sme mohli zátvorky vynechať a použiť zápis `\TRANS\alpha\beta\gamma\delta`, ale kvôli prehľadnosti sa to nedoporučuje. Ak by sme napísali `\TRANS alpha\beta\gamma\delta` tak, by nám LaTeXvyhlásil chybu. Prvé štyri znaky `a l p h` by použil ako parametre príkazu, posledné `a` by ešte napísal, ale narazil by na príkaz `\beta`, ktorý vyžaduje matematický mód (ten je zavedený pri definícii makra `\TRANS`).

Ako sme už spomínali, príkaz `\def` má lokálny charakter, ak chceme aby definovaný príkaz mal globálny charakter, t. j. aby platil aj mimo definované prostredie, môžeme použiť príkaz `\gdef` alebo `\def` s prefixom `\global`. Uvedené príkazy sú identické, t. j. `\gdef` a `\global\def` znamenajú to isté.

Ďalším príkazom, ktorý môžeme tvoriť vlastné makrá je `\newcommand`, ktorý funguje trochu ináč ako príkaz `\def`. Tento príkaz kontroluje, či príkaz alebo makro (systémové alebo vlastné) s takýmto menom existuje. V kladnom prípade hlási LaTeX chybu. Ak napriek tomu chceme príkaz predefinovať, musíme použiť `\renewcommand`. Príkazy `\newcommand`, `\renewcommand` majú rovnakú formu zápisu `\newcommand\názov_príkazu <parametre> {telo definície}`.

Ak novovytvorený príkaz neobsahuje parametre, potom je jeho definícia rovnaká ako pri príkaze `\def`. Príkazy `\newcommand\Konf{KonFERerencia začína!}` a `\def\Konf{KonFERerencia začína!}` majú rovnaký význam a ich výstup je identický. Definíciu `\def\Konf[#1|#2]{{\it KonFERerencia #2 začína #1!}}` môžeme zapísať `\newcommand\Konf[2]{{\it KonFERerencia #2 začína #1!}}`. To znamená, že pri použití príkazu `\newcommand` nezadávame masku parametrov, ale iba ich počet v zátvorkách `[]`.

Navyše môžeme parameter `#1` implicitne definovať ako prednastavený. Jeho hodnota sa definuje taktiež v hranatých zátvorkach hneď za počtom premenných. Pri použití sa prvý parameter zadáva v hranatých zátvorkách `[]`, alebo sa vôbec neuvádza a dosadí sa zaň implicitná hodnota. Ostatné parametre počnúc parametrom `#2`, pokiaľ existujú, sa zadávajú rovnako ako pri príkaze `\def`. V príklade `\newcommand\Konf[2][2.7.2013]{{\it KonFERerencia #2 začína #1!}}` je hodnota parametra `#1` implicitne nastavená na hodnotu 2.7.2013. To znamená, že príkazy `\Konf[2.7.2013]{OssConf}` a `\Konf{OssConf}` majú rovnaký význam.

2. Príkazy cyklov `\foreach` a `\multido`

V predchádzajúcej časti sme ukázali ako sa definujú v LaTeX-u vlastné príkazy a makrá. Na ich hromadné používanie s viac-menej úspešným výsledkom môžeme použiť cykly. Cykly sa dajú tvoriť aj na základnej, tzv. plain úrovni TeX-u (viď napr. [5]). Užitočnejšie je použiť nejaký hotový balíček. V tejto časti sa budeme zaoberať príkazom `\foreach`. Na jeho použitie musíme v preambule použiť `\usepackage{pgffor}`. Ak používame `TiKZ`, potom sa balíček `pgffor` načíta automaticky [1]. Príkaz sa používa v tvare `\foreach \prem {zoznam}{príkazy}`. Príkaz má povinný parameter `\prem`, ktorý postupne nadobúda všetky hodnoty z množiny `zoznam` a dosadzuje ich do príkazov `príkazy`.

Množina `zoznam` sa zadáva vymenovaním prvkov oddelených čiarkou. Pri číselných premenných môžeme použiť konštrukciu `{začiatok, ..., koniec}`. Premenná `\prem` sa postupne nahrádza číslami `začiatok`, `začiatok+1`, `začiatok+2`, `začiatok+3` až po maximálnu hodnotu, ktorá nie je väčšia ako `koniec`. Napríklad príkazom `\foreach \xx in {1.1,...,10} { \xx}` sa postupne vypíšu čísla 1.1 2.1 3.1 4.1 5.1 6.1 7.1 8.1 9.1. Rovnaký efekt by sme dosiahli tiež oveľa pracnejším príkazom `\foreach \xx in {1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1} { \xx}`.

Inou možnosťou je použiť cyklus `\multido` z rovnomenného balíčka `\multido`. Používa sa v tvare `\multido{premenné}{počet opakovaní}{príkazy}`. V povinnom parametri `premenné` sa deklaruje typ, počiatočná hodnota a prírastok premennej, ktorá je výhradne číslom a používa sa v parametri `príkazy`. Definuje sa v tvare `\prem=poč.hodnota+prírastok`. Ak je prírastok záporný, píše sa v tvare `\prem=poč.hodnota+-prírastok`. Prvé písmeno názvu `\prem` určuje jej typ:

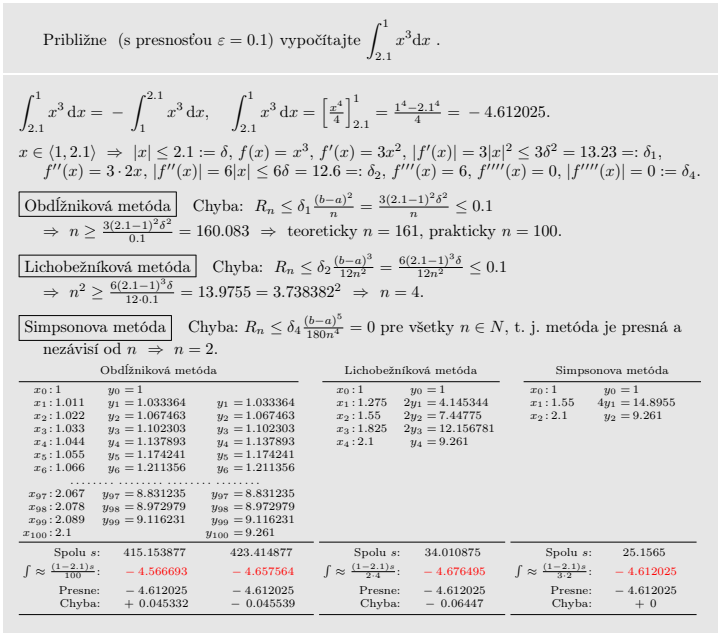
Dimension (d D) — dĺžková miera, výsledok je určený v jednotkách `sp`. Napr. `\multido{\dx=4cm+-1em}{2}{ [\dx]}` dáva výstup [7458719sp] [6745171sp].

Number (n N) — počiatočná hodnota a prírastok sú čísla s rovnakým počtom desatinných miest (maximálne 8), pričom počiatočná hodnota môže byť tiež celým číslom, výsledkom je číslo s rovnakým počtom desatinných miest, ale fixovaným. Napr. `\multido{\nx=4+1.05}{2}{ [\nx]}` dáva výstup [4] [5.05] a `\multido{\nx=4.05+1.05}{2}{ [\nx]}` dáva výstup [4.05] [5.10].

Integer (i I) — počiatočná hodnota a prírastok sú celé čísla, výsledkom je celé číslo. Napr. `\multido{\ix=4+1}{2}{ [\ix]}` dáva výstup [4] [5].

Real (r R) — počiatočná hodnota a prírastok sú čísla s maximálne 4 desatinnými miestami, výsledkom je reálne číslo s odrezanými nulami za desatinnou bodkou. Tento typ je rýchlejší ako Number, ale s relatívne početnými numerickými výpočtovými chybami. Napr. `\multido{\rx=4+1.05}{2}{ [\rx]}` dáva výstup [4.0] [5.05] a `\multido{\rx=4.05+1.05}{2}{ [\rx]}` dáva výstup [4.05] [5.1] a `\multido{\rx=2+3.05}{6}{ [\rx]}` dáva výstup [2.0] [5.05] [8.1] [11.15001] [14.20001] [17.25002].

Príkaz môžeme použiť aj bez premennej, napr. `\multido{}{8}{ \LaTeX{}}` a dostaneme LaTeX LaTeX LaTeX LaTeX LaTeX LaTeX LaTeX LaTeX.



Približne (s presnosťou $\varepsilon = 0.1$) vypočítajte $\int_{2.1}^{1} x^3 \mathrm{d}x$.

$\int_{2.1}^{1} x^3\,\mathrm{d}x = -\int_{1}^{2.1} x^3\,\mathrm{d}x, \quad \int_{2.1}^{1} x^3\,\mathrm{d}x = \left[\frac{x^4}{4}\right]_{2.1}^{1} = \frac{1^4-2.1^4}{4} = -4.612025.$

$x \in \langle 1, 2.1\rangle \Rightarrow |x| \le 2.1 := \delta,\ f(x) = x^3,\ f'(x) = 3x^2,\ |f'(x)| = 3|x|^2 \le 3\delta^2 = 13.23 =: \delta_1,$
$f''(x) = 3\cdot 2x,\ |f''(x)| = 6|x| \le 6\delta = 12.6 =: \delta_2,\ f'''(x) = 6,\ f''''(x) = 0,\ |f''''(x)| = 0 := \delta_4.$

Obdĺžniková metóda Chyba: $R_n \le \delta_1 \frac{(b-a)^2}{n} = \frac{3(2.1-1)^2\delta^2}{n} \le 0.1$
$\Rightarrow\ n \ge \frac{3(2.1-1)^2\delta^2}{0.1} = 160.083\ \Rightarrow$ teoreticky $n = 161$, prakticky $n = 100$.

Lichobežníková metóda Chyba: $R_n \le \delta_2 \frac{(b-a)^3}{12n^2} = \frac{6(2.1-1)^3\delta}{12n^2} \le 0.1$
$\Rightarrow\ n^2 \ge \frac{6(2.1-1)^3\delta}{12\cdot 0.1} = 13.9755 = 3.738382^2\ \Rightarrow\ n = 4.$

Simpsonova metóda Chyba: $R_n \le \delta_4 \frac{(b-a)^5}{180n^4} = 0$ pre všetky $n \in N$, t. j. metóda je presná a nezávisí od $n\ \Rightarrow\ n = 2$.

Obdĺžniková metóda

$x_0: 1$	$y_0 = 1$	
$x_1: 1.011$	$y_1 = 1.033364$	$y_1 = 1.033364$
$x_2: 1.022$	$y_2 = 1.067463$	$y_2 = 1.067463$
$x_3: 1.033$	$y_3 = 1.102303$	$y_3 = 1.102303$
$x_4: 1.044$	$y_4 = 1.137893$	$y_4 = 1.137893$
$x_5: 1.055$	$y_5 = 1.174241$	$y_5 = 1.174241$
$x_6: 1.066$	$y_6 = 1.211356$	$y_6 = 1.211356$
.....	.....	.....
$x_{97}: 2.067$	$y_{97} = 8.831235$	$y_{97} = 8.831235$
$x_{98}: 2.078$	$y_{98} = 8.972979$	$y_{98} = 8.972979$
$x_{99}: 2.089$	$y_{99} = 9.116231$	$y_{99} = 9.116231$
$x_{100}: 2.1$		$y_{100} = 9.261$
Spolu s:	415.153877	423.414877
$\int \approx \frac{(1-2.1)s}{100}$:	− 4.566693	− 4.657564
Presne:	− 4.612025	− 4.612025
Chyba:	+ 0.045332	− 0.045539

Lichobežníková metóda

$x_0: 1$	$y_0 = 1$
$x_1: 1.275$	$2y_1 = 4.145344$
$x_2: 1.55$	$2y_2 = 7.44775$
$x_3: 1.825$	$2y_3 = 12.156781$
$x_4: 2.1$	$y_4 = 9.261$
Spolu s:	34.010875
$\int \approx \frac{(1-2.1)s}{2\cdot 4}$:	− 4.676495
Presne:	− 4.612025
Chyba:	− 0.06447

Simpsonova metóda

$x_0: 1$	$y_0 = 1$
$x_1: 1.55$	$4y_1 = 14.8955$
$x_2: 2.1$	$y_2 = 9.261$
Spolu s:	25.1565
$\int \approx \frac{(1-2.1)s}{3\cdot 2}$:	− 4.612025
Presne:	− 4.612025
Chyba:	+ 0

Obr. 1. Približne (s presnosťou $\varepsilon = 0.1$) vypočítajte $\int_{2.1}^{1} x^3 \mathrm{d}x$.

3. Príklad na numerické integrovanie

Na záver článku uvedieme jednoduché makro, ktoré dokáže numericky spočítať integrál $\int_a^b x^k \mathrm{d}x$. Makro má 4 parametre `\RdroCB[n]{a}{b}{k}`, pričom prvý parameter určuje počet deliacich bodov výpočtu (maximálne $n = 100$). Ak je táto hodnota desatinné číslo z intervalu $(0, 1)$, potom sa považuje za presnosť a počet deliacich bodov sa dopočíta. Pri výpočte sa zobrazí tabuľka s deliacimi bodmi a prisluchajúcimi iteračnými hodnotami. Záver tabuľky tvorí presná hodnota integrálu, vypočítaná približná hodnota a chyba výpočtu (obr. **??**, **??**). Pri výpočte bol použitý cyklus `\foreach` spolu s balíčkom `fp` (Fixed Point package — výpočty v pevnej rádovej čiarke), ktorý dovoľuje používať matematické výpočty priamo v zdrojovom kóde LaTeX-u a s balíčkom `xifthen`, ktorý umožňuje vetviť makrá do rôznych logických častí na základe ich argumentov. K makru na riešenie úlohy prináleží makro s identickými parametrami, ktoré zadáva problém.

```
%_____________________________________________________________________________
\newcommand\droCB[4][.1]{\tjP[\riA]{-(#1)}\tjP[\riB]{(#1)-1}\def\ak{\ifthenelse}%
\def\ZLE{{\color{red}BAD:} $k=#2$, $a=#3$, $b=#4$ {\color{red}OK:} $a\ne b$}%
```

```
\tjI[\Xa]{#2}\tjP[\Xb]{(#2)-1}\tjO[\Xc]{(#4)-(#3)}%
\rbPr{\rALL}{(\Xa)*(\Xb)*((\Xc)-1)}%
\ak{\rALL=0}{\ZLE}{Približne \ak{\riA=0 \and \riB=0}{\rbPr{\rI}{max(#1,.001)}
(s~presnosťou~$\varepsilon=\rI$)}{} vypočítajte
$\displaystyle\int_{\rb{#3}}^{\rb{#4}}x^{\rbJExp{#2}}{\rm d}x$
\ak{\riB=1}{\rbPr{\gN}{trunc(min(max(#1,5),100),0)}
(použite aspoň $\gN$~deliacich bodov)}{}.}}
```

Výpočty a testovania boli riešené pomocou vlastnžch makier, napr. na testovanie celočíselnosti alebo kladnosti hodnôt boli použité príkazy (výsledkom je logická hodnota 0 alebo 1):

```
\newcommand\tjI[2][\rrT]{\rbx{#2}%
                         \FPifint{\rbV}\FPset{#1}{1}\else\FPset{#1}{0}\fi}
\newcommand\tjP[2][\rrT]{\rbx{#2}%
                         \FPifpos{\rbV}\FPset{#1}{1}\else\FPset{#1}{0}\fi}
```

Kompletný zdrojový kód obidvoch makier a všetkých pomocných testovacích a výpočtových makier je zverejnený na `http://frcatel.fri.uniza.sk/~beerb/latex/num-in.pdf`. Pri výpočte v cykle bolo potrebné si zapamätať medzivýpočty, t. j. definovať globálne premenné a zabezpečiť ich korektné sčítavanie. To bolo riešené pomocou príkazu priradenia `\let` a pomocných premenných.

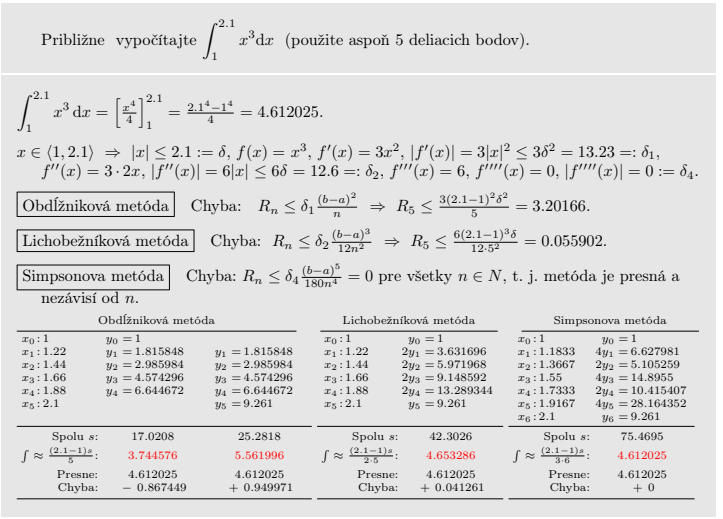
```
\foreach \gX in {\gZ, ..., \gK} %%%.............................. VLASTNY VYPOCET
{\rbPr[8]{\Go}{(\DH)+(\gX)*((\HH)-(\DH))/(\So)}\ak{\gX<\SoJ}{\xxD{\rbPr{\bo}{1}}%
 {\rbPr{\bo}{sgn(\Go)}}\rbPr[8]{\Yo}{(\bo)*abs(\Go)^(#2)}}{\rbPr[8]{\Yo}{0}}%
 \rbPr[8]{\Gl}{(\DH)+(\gX)*((\HH)-(\DH))/(\Sl)}\ak{\gX<\SlJ}{\xxD{\rbPr{\bl}{1}}%
 {\rbPr{\bl}{sgn(\Gl)}}\rbPr[8]{\Yl}{(\bl)*abs(\Gl)^(#2)}}{\rbPr[8]{\Yl}{0}}%
 \rbPr[8]{\Gs}{(\DH)+(\gX)*((\HH)-(\DH))/(\Ss)}\ak{\gX<\SsJ}{\xxD{\rbPr{\bs}{1}}%
 {\rbPr{\bs}{sgn(\Gs)}}\rbPr[8]{\Ys}{(\bs)*abs(\Gs)^(#2)}}{\rbPr[8]{\Ys}{0}}%
\rbPr{\QL}{0}\rbPr{\QP}{0}\rbPr{\Ql}{0}\rbPr{\Qs}{0}%
 \ak{\gX<\So}{\rbPr{\QL}{1}}{}\ak{\gX>0 \AND \gX<\SoJ}{\rbPr{\QP}{1}}{}%
   \rbPrG[8]{\VL}{(\hL)+(\QL)*(\Yo)}\global\let\hL\VL%
   \rbPrG[8]{\VP}{(\hP)+(\QP)*(\Yo)}\global\let\hP\VP%
 \ak{\gX<\SlJ}{\rbPr{\Ql}{2}\ak{\gX=0 \OR \gX=\Sl}{\rbPr{\Ql}{1}}{}}{}%
   \rbPrG[8]{\Vl}{(\hl)+(\Ql)*(\Yl)}\global\let\hl\Vl%
 \ak{\gX<\SsJ}{\rbPr{\Qs}{4}\ttI{(\gX)/2}{\rbPr{\Qs}{2}}{}%
   \ak{\gX=0 \OR \gX=\Ss}{\rbPr{\Qs}{1}}{}}{}%
   \rbPrG[8]{\Vs}{(\hs)+(\Qs)*(\Ys)}\global\let\hs\Vs%

\ak{\gX<\SoJ}{\xxT2{\hfill\!\!$x_{\gX}\!:$& %%%......... VLASTNE TABULKY VYPOCTOV
 $\rb[4]{\Go}$& \ak{\gX=\So}{&}{\hfill$y_{\gX}=$&
 $\rb[6]{\Yo}$}& \ak{\gX=0}{&}{\hfill$y_{\gX}=$& $\rb[6]{\Yo}$}}}{\xxT2{&&&&&}}
\hfill\ak{\gX<\SlJ}{\xxT1{\hfill\!\!$x_{\gX}\!:$&
 $\rb[4]{\Gl}$& \hfill$\rbJ{\Ql}y_{\gX}=$& $\rb[6]{(\Ql)*\Yl}$}}{\xxT1{&&&}}
\hfill\ak{\gX<\SsJ}{\xxT1{\hfill\!\!$x_{\gX}\!:$&
 $\rb[4]{\Gs}$& \hfill$\rbJ{\Qs}y_{\gX}=$& $\rb[6]{(\Qs)*\Ys}$}}{\xxT1{&&&}}

}%%%.......................................................... KONIEC \foreach
```

Záver

Príkazy na tvorenie cyklov spolu s rôznymi ďalšími balíčkami značne rozširujú možnosti LaTeX-u. V článku je uvedený postup ako využiť cykly pri numerickom integrovaní. V spojení s balíčkom `TikZ` zase dovoľujú kresliť rozličné netriviálne a mnohorozvetvené útvary [1].



Približne vypočítajte $\int_1^{2.1} x^3 \mathrm{d}x$ (použite aspoň 5 deliacich bodov).

$\int_1^{2.1} x^3\,\mathrm{d}x = \left[\frac{x^4}{4}\right]_1^{2.1} = \frac{2.1^4-1^4}{4} = 4.612025.$

$x \in \langle 1, 2.1\rangle \Rightarrow |x| \le 2.1 := \delta,\ f(x) = x^3,\ f'(x) = 3x^2,\ |f'(x)| = 3|x|^2 \le 3\delta^2 = 13.23 =: \delta_1,$
$f''(x) = 3 \cdot 2x,\ |f''(x)| = 6|x| \le 6\delta = 12.6 =: \delta_2,\ f'''(x) = 6,\ f''''(x) = 0,\ |f''''(x)| = 0 := \delta_4.$

Obdĺžniková metóda Chyba: $R_n \le \delta_1 \frac{(b-a)^2}{n} \Rightarrow R_5 \le \frac{3(2.1-1)^2\delta^2}{5} = 3.20166.$

Lichobežníková metóda Chyba: $R_n \le \delta_2 \frac{(b-a)^3}{12n^2} \Rightarrow R_5 \le \frac{6(2.1-1)^3\delta}{12\cdot 5^2} = 0.055902.$

Simpsonova metóda Chyba: $R_n \le \delta_4 \frac{(b-a)^5}{180n^4} = 0$ pre všetky $n \in N$, t. j. metóda je presná a nezávisí od n.

Obdĺžniková metóda

$x_0: 1$	$y_0 = 1$	
$x_1: 1.22$	$y_1 = 1.815848$	$y_1 = 1.815848$
$x_2: 1.44$	$y_2 = 2.985984$	$y_2 = 2.985984$
$x_3: 1.66$	$y_3 = 4.574296$	$y_3 = 4.574296$
$x_4: 1.88$	$y_4 = 6.644672$	$y_4 = 6.644672$
$x_5: 2.1$		$y_5 = 9.261$
Spolu s:	17.0208	25.2818
$\int \approx \frac{(2.1-1)s}{5}$:	3.744576	5.561996
Presne:	4.612025	4.612025
Chyba:	− 0.867449	+ 0.949971

Lichobežníková metóda

$x_0: 1$	$y_0 = 1$
$x_1: 1.22$	$2y_1 = 3.631696$
$x_2: 1.44$	$2y_2 = 5.971968$
$x_3: 1.66$	$2y_3 = 9.148592$
$x_4: 1.88$	$2y_4 = 13.289344$
$x_5: 2.1$	$y_5 = 9.261$
Spolu s:	42.3026
$\int \approx \frac{(2.1-1)s}{2\cdot 5}$:	4.653286
Presne:	4.612025
Chyba:	+ 0.041261

Simpsonova metóda

$x_0: 1$	$y_0 = 1$
$x_1: 1.1833$	$4y_1 = 6.627981$
$x_2: 1.3667$	$2y_2 = 5.105259$
$x_3: 1.55$	$4y_3 = 14.8955$
$x_4: 1.7333$	$2y_4 = 10.415407$
$x_5: 1.9167$	$4y_5 = 28.164352$
$x_6: 2.1$	$y_6 = 9.261$
Spolu s:	75.4695
$\int \approx \frac{(2.1-1)s}{3\cdot 6}$:	4.612025
Presne:	4.612025
Chyba:	+ 0

Obr. 2. Približne vypočítajte $\int_1^{2.1} x^3 \mathrm{d}x$ (požte aspoň 5 deliacich bodov).

Literatúra

[1] BLAŠKO, R., KOZUBÍK, A.: *LaTeX, teória grafov a trochu umenia*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 85–90, ISBN 978-80-970457-2-2.

[2] KNUTH, D. E.: *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.

[3] KOZUBÍK, A.: *Naučím vás kresliť alebo predstavenie balíčka TiKZ*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 91–96, ISBN 978-80-970457-2-2.

[4] KOZUBÍK, A., KOZUBÍKOVÁ, Z.,: *Tabuľkové výpočty v systéme LaTeX*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 97–102, ISBN 978-80-970457-2-2.

[5] OLŠÁK, P.: *TeXbook naruby*, Brno, Konvoj 1997, 468 str., ISBN 80-85615-64-9.

Kontaktná adresa

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