

Obsah

Úvod	5
1. Diplomová práce z matematiky	6
1. Začínáme	6
2. Hlavičky a patičky	8
3. Tabulky	10
4. Věta, definice, důkaz,	11
5. Křížové odkazy	12
2. Sazba matematiky, balík amsmath	13
6. Základní pravidla matematické sazby	14
7. Fonty a symboly	15
7.1. Tučné matematické symboly	15
7.2. Tabulky matematických symbolů	16
8. Složené symboly, oddělovače a operátory	23
8.1. Násobné integrály	23
8.2. Šipky nad a pod výrazem	23
8.3. „Natahovací“ šipky	24
8.4. Elipsy (tečky)	24
8.5. Akcenty v matematickém režimu	24
8.6. Odmocniny	26
8.7. Rámování formulí	26
8.8. Složené symboly	26
8.9. Přesahy	27
8.10. Text uvnitř matematického prostředí	27
8.11. Jména operátorů	27
8.12. Binární operátory	28
8.13. Zlomky a související konstrukce	29
8.14. Řetězové zlomky	30
8.15. Oddělovače	31
8.16. Mezery v matematickém módu	33
9. Matice	34
9.1. Větvení	37
9.2. Víceřádkové indexy a exponenty	37
10. Rovnice a vzorce na více řádků	38
10.1. Prostedí <code>align</code>	38

10.2.	Prostředí <code>alignat</code>	41
10.3.	Prostředí <code>gather</code>	42
10.4.	Prostředí <code>multline</code>	42
10.5.	Prostředí <code>split</code>	43
10.6.	Prostředí <code>aligned</code> , <code>gathered</code> a <code>alignedat</code>	44
10.7.	Další úpravy formulí přes více řádků	45
10.8.	Příkaz <code>\intertext</code>	46
10.9.	Příklady použití prostředí pro rovnice na více řádků	46
10.10.	Číslování rovnic	52
3.	Grafika	53
11.	Obrázky kreslené L ^A T _E Xem	54
11.1.	Prostředí <code>picture</code>	54
11.2.	Mřížky — balík <code>graphpap</code>	60
11.3.	Kružnice velkých průměrů — balík <code>pspicture</code>	60
11.4.	Balík <code>Xy-pic</code>	62
11.5.	Balík <code>amscd</code>	66
11.6.	Barvy — balík <code>color</code>	67
12.	Obrázky vytvořené externě	69
12.1.	Příprava obrázku v grafickém formátu EPS (popř. PS)	69
12.2.	Vkládání obrázku ve formátu EPS (popř. PS) do L ^A T _E Xu	72
12.3.	Další příkazy balíku <code>graphicx</code>	79
13.	Začlenění obrázků do textu — prostředí <code>figure</code>	85
13.1.	Jak ovlivnit sazbu plovoucích obrázků	85
13.2.	Balík <code>float</code>	86
13.3.	Balík <code>subfigure</code>	87
14.	Popisy obrázků — balík <code>PSfrag</code>	88
15.	Obtékání grafických objektů textem	94
16.	Číslování obrázků	97
17.	Balík <code>mfpic</code>	98
18.	Balík <code>PStricks</code>	108
	Rejstřík	111
	Seznam balíků	114
	Seznam prostředí	114
	Seznam příkazů	115

Úvod

Skriptum je primárně určeno posluchačům matematických oborů na PřF MU v Brně. Tomu je podřízen obsah i koncepce celého textu. Užitečnou pomůckou se však může stát i řadě dalších uživatelů.

Autoři si kladli za cíl usnadnit studentům práci při sazbě diplomových prací a dalších matematických textů v systému $\text{\LaTeX} 2_{\epsilon}$.¹ Skriptum je koncipováno jako nadstavba nad publikací *Rybička J.: \LaTeX pro začátečníky* [2], na kterou se v textu často odkazujeme.

V první části jsou uvedeny pokyny k sazbě diplomové práce (hlavičky, nastavení velikosti stránek apod.) a rady k některým častým problémům vyskytujícím se v matematických textech, které nenajdete v publikaci [2]. Druhá část je věnována především popisu a příkladům použití balíku `amsmath` rozšiřujícímu možnosti matematické sazby. Třetí část popisuje některé možnosti tvorby „matematické“ grafiky v \LaTeX u a následně ukazuje způsoby jejího začlenění do textu. Popsána je i problematika začlenění grafiky vytvořené externě.

Posluchači matematických oborů na PřF MU v Brně mají možnost užívat síťovou instalaci \TeX u a další programové vybavení na počítači *bart* (na operačním systému Linux). Vzhledem k této skutečnosti jsou (až na malé výjimky) vynechány veškeré informace týkající se instalace samotného \TeX u, prohlížečů výstupních souborů, instalace potřebných fontů apod. a předpokládá se, že uživatelé budou pracovat na tomto serveru. Uživatelům, kteří „ \TeX ují“ jinde, lze doporučit stránky Československého sdružení uživatelů \TeX u ($\text{\textit{CS TUG}}$): <http://www.cstug.cz>. Analogicky je koncipován i výběr programů, u kterých je podrobněji popisována spolupráce s \LaTeX em.

Jak pracovat s textem? Text je koncipován tak, aby bylo možno číst samostatně jednotlivé partie. K rychlému vyhledávání slouží kromě obsahu a abecedního rejstříku také rejstříky tematicky zaměřené (Seznam příkazů, Seznam balíčků apod.).

Ve skriptech je uvedeno velké množství příkladů i s odpovídající částí zdrojového souboru. Příklady jsou označeny symbolem ■ a číslovány. Doporučujeme věnovat jim značnou pozornost, neboť často doplňují (nebo i zcela nahrazují) slovní komentář v textu.

★ Na konci některých odstavců jsou v rámečku (jako tento) uvedeny užitečné postřehy a upozornění.

V úvodu odstavců jsou uvedeny tzv. „předpoklady“, čímž se nejčastěji rozumí užití některého speciálního balíku `maker` (více o práci s balíky `maker` viz odstavec **1. Začínáme**).

Autoři děkují doc. RNDr. Jaromíru Kubenovi, CSc, RNDr. Petru Sojkovi a ing. Jiřímu Šremrovi, PhD. za pečlivé přečtení textu a cenné připomínky.

L. Čechová, R. Plch

¹Nadále budeme všude v textu pod označením \LaTeX mít na mysli $\text{\LaTeX} 2_{\epsilon}$.

Část 1

Diplomová práce z matematiky

1. Začínáme

Jak tedy začít? Pro sazbu diplomové práce z matematiky doporučujeme použít systém \TeX s jeho nadstavbou \LaTeX , třídou `article` nebo `report` a pro sazbu matematiky využít balíku `amsmath`. Národní prostředí definujeme pomocí balíku `czech`.

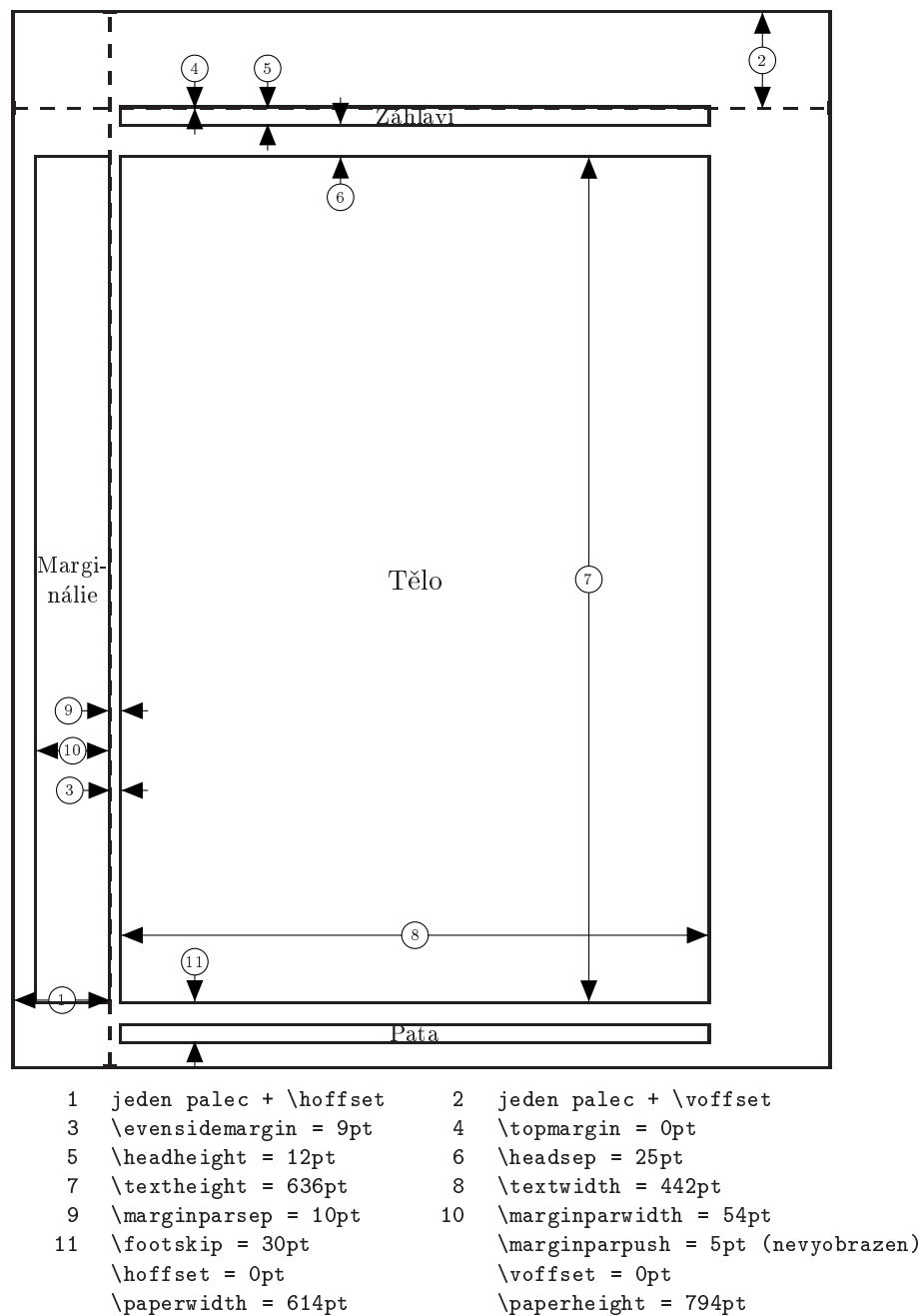
Předpokládáme-li užití některého balíku maker, např. balíku `amsmath`, pak je třeba zajistit v prvé řadě načtení maker tohoto balíku, tj. v preambuli uvést `\usepackage{amsmath}` (popř. `\usepackage[nepovinné parametry]{amsmath}`). Jestliže dostáváme při překladu zdrojového souboru chybové hlášení `! LaTeX Error: File 'amsmath.sty' not found`, pak to znamená, že vaše instalace neobsahuje příslušný balík, tj. v tomto případě soubor `amsmath.sty`. Tento soubor je třeba vyhledat (např. stáhnout z webovských stránek) a vložit do patřičného adresáře prohledávaného \LaTeX em — tím může být např. aktuální adresář, ve kterém máte váš zdrojový soubor, tj. `soubor.tex`. K vyhledávání balíků je možno využít např. archívy CTAN a $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ (<http://www.cstug.cz/ctan/index.html>). Některé balíky a uživatelské příručky lze najít také na <http://www.math.muni.cz/~mlc/latex/>.

Diplomové práce se tisknou obvykle jednostranně, proto není třeba použití volby `twoside` pro dvoustranné dokumenty (nařizuje vytváření levostranných a pravostranných stránek). Základní kostra dokumentu pak tedy vypadá takto:

```
\documentclass[12pt,a4paper]{article}
\usepackage{czech}
\usepackage{amsmath,amssymb}
\begin{document} Vlastní tělo dokumentu
\end{document}
```

Pokud nám vyhovují předdefinované rozměry stránky můžeme použít volitelný parametr `a4paper` nebo balík `a4wide`. Avšak někdy autor potřebuje tyto přednastavené hodnoty upravit podle svých představ. Na obrázku 1.1 jsou zobrazeny veškeré nastavitelné parametry, které ovlivňují vzhled stránky.

\LaTeX nabízí dva příkazy pomocí nichž lze tyto parametry měnit. Obvyčejně se objevují v preambuli dokumentu.



Obrázek 1.1: Parametry ovlivňující vzhled stránky

První příkaz umožňuje přiřadit libovolnému z těchto parametrů pevnou hodnotu:

$$\backslash\mathrm{setlength}\{parameter\}{délka}$$

Druhý umožňuje zvětšit libovolný z těchto parametrů o pevný rozměr *délka*

$$\backslash\mathrm{addtolength}\{parameter\}{délka}$$

Tento příkaz je možná užitečnější než `\setlength`, neboť umožňuje nastavovat rozměry vzhledem k již nastaveným hodnotám. Přidání jednoho centimetru k celkové šířce textu by se provedlo vložením těchto příkazů do preamble dokumentu:

$$\begin{aligned} &\backslash\mathrm{addtolength}\{\mathrm{hoffset}\}{-0.5\mathrm{cm}} \\ &\backslash\mathrm{addtolength}\{\mathrm{textwidth}\}{1\mathrm{cm}} \end{aligned}$$

2. Hlavičky a patičky

Nastavení hlaviček (záhlaví) a patiček je v L^AT_EXu definováno pomocí příkazů `\pagestyle` a `\pagenumbering`. Příkaz `\pagestyle` definuje obsah hlaviček a patiček (například kde se budou tisknout čísla stránek), zatímco `\pagenumbering` definuje formát čísla stránky. Předdefinované styly jsou tyto:

<code>empty</code>	hlavička i pata je prázdná, číslování není vypisováno
<code>plain</code>	prázdná hlavička, číslo stránky je uvedeno uprostřed paty
<code>headings</code>	prázdná pata, hlavička obsahuje název běžné kapitoly nebo sekce a číslo stránky
<code>myheadings</code>	prázdná pata, hlavička obsahuje číslo stránky a uživatelem specifikovanou informaci (pomocí <code>\markright</code> a <code>\markboth</code>)

Příkazem `\pagenumbering` lze ovlivnit způsob výpisu čísel stránek. Parametrem příkazu může být:

<code>arabic</code>	arabské číslice
<code>roman</code>	malé římské číslice
<code>Roman</code>	velké římské číslice
<code>alph</code>	písmena malé abecedy
<code>Alph</code>	písmena velké abecedy

Protože standardní možnosti modifikace vzhledu hlaviček a paty jsou značně omezené, doporučujeme při sazbě diplomové práce použít balík `fancyhdr`. Tento umožňuje:

- rozdělit hlavičku i patu do tří částí
- umístění vodorovné čáry do hlavičky i paty
- hlavičku i patu širší než je `textwidth`
- hlavičku i patu přes více řádků
- rozdílné nastavení pro sudé a liché stránky
- rozdílné nastavení pro první stránky kapitol

Pro použití tohoto stylu použijte v preambuli dokumentu příkazy `\usepackage{fancyhdr}` a `\pagestyle{fancy}` (musí být uveden až po nastavení `\textwidth`). Vzhled stránky definovaný pomocí `fancyhdr` můžeme zobrazit následujícím způsobem:

LeftHeader	CenteredHeader	RightHeader
tělo stránky		
LeftFooter	CenteredFooter	RightFooter

Informace v políčku `LeftHeader` a `LeftFooter` bude zarovnána nalevo, v políčku `CenteredHeader` a `CenteredFooter` centrována a v políčku `RightHeader` a `RightFooter` zarovnána vpravo. Tloušťku čáry pod hlavičkou a nad patou nastavujeme pomocí `\headrulewidth` (implicitní nastavení 0.4pt) a `\footrulewidth` (0pt). Při tloušťce 0pt je čára neviditelná. Na příkladu si nyní ukážeme nejčastěji používané nastavení pro diplomové práce (při použití třídy `article`):

1. DIPLOMOVÁ PRÁCE	3
tělo stránky	

```
\pagestyle{fancy}
\fancyhf{} %pro zrušení všech nastavení
\renewcommand{\sectionmark}[1]%
{\markboth{\thesection .\ #1 }{}}%
%generuje v hlavičce tečku za číslem kapitoly
\fancyhead[L]{\scshape\leftmark}
\fancyhead[R]{\thepage}
%příkaz \thepage vysadí číslo aktuální strany
\renewcommand{\headrulewidth}{0.4pt}
```

Na závěr si ještě ukážeme nastavení vhodné pro dvoustranné dokumenty:

```
\pagestyle{fancy}
\fancyhf{} % pro zrušení všech nastavení
\renewcommand{\sectionmark}[1]%
{\markboth{\thesection .\ #1 }{}}
\renewcommand{\subsectionmark}[1]%
{\markright{\thesubsection .\ #1 }{}}
\fancyhead[RE]{\scshape\leftmark}
\fancyhead[LO]{\scshape\rightmark}
\fancyhead[LE,R0]{\thepage}
%Na lichých stránkách vpravo (R0) a sudých stránkách vlevo (LE)
%bude vypisováno číslo stránky.
\renewcommand{\headrulewidth}{0.4pt}
```

Zobrazení kapitol

Můžeme měnit i tvar zobrazování nadpisů v záhlaví. Výpis je realizován pomocí příkazu `\markboth`, který měníme podle vlastních požadavků.

Příkaz `\thesection` zobrazuje číslo sekce, `\sectionmark` určuje, jak bude výpis vypadat a `\markboth` sází obsah svých parametrů do hlaviček. Tento příkaz má dva povinné parametry a jeho syntaxe je následující:

```
\markboth{levá strana}{pravá strana}
```

Parametr *levá strana* (`\leftmark`) se sází na levých (sudých) stranách a *pravá strana* (`\rightmark`) se sází na pravých (lichých) stranách. Příkaz `\markright{hlavička}` nastavuje pravou (lichou) hlavičku (`\rightmark`).

Změna předdefinovaných stylů

Pokud používáme třídu `report` a chceme jiný tvar záhlaví a paty na první stránce kapitoly, musíme předdefinovat styl `plain` (protože příkaz `chapter` má ve své definici uveden příkaz pro změnu stylu aktuální stránky: `\thispagestyle{plain}`).

Když např. budeme chtít na každou stranu, kde začíná kapitola, umístit text „STRANA X“, musíme provést následující úpravy:

```
\fancypagestyle{plain}{%
\fancyhf{}
\fancyfoot[C]{\scshape Strana \thepage}
\renewcommand{\headrulewidth}{0pt}}
```

3. Tabulky

Pro sazbu tabulek je v systému \LaTeX k dispozici prostředí `tabular`. Toto prostředí je podrobně popsáno v [2], zde proto uvádíme pouze jednu ukázkou použití:

■ Příklad 3.1.

A	B			C	D
	\emptyset	$M_1 \times M_2$	$=$		
–	–	+	+	+	+
–	+	+	+	+	–

```
\begin{tabular}{|c|c|c|c|c|c|}
\hline & \multicolumn{3}{c|}{B}& & \\
\cline{2-4}
\rule[-3mm]{0pt}{8mm}
\raisebox{8pt}[0pt]{A}
&\emptyset&M_1\times M_2&=& \\
\raisebox{8pt}[0pt]{C}& \\
\raisebox{8pt}[0pt]{D}& \\
\hline & & & & & \\
\hline & & & & & \\
\hline \end{tabular}
```

Rozšíření možností sazby tabulek poskytují balíky popsané v [7]. Balík `array` rozšiřuje možnosti svislého zarovnávání odstavcových sloupců a možnosti opakování materiálu v každém řádku, balík `tabularx` umožňuje sázet tabulky na určitou šířku s automatickým výpočtem šířky

jednotlivých sloupců. Pro tabulky přes více stran jsou určeny balíky supertab a longtable. Speciální zarovnání materiálu ve sloupcích definuje balík dcolumn a další možnosti ohraničujících čar tabulek pak balík hline.

4. Věta, definice, důkaz, ...

Hned na začátku práce je vhodné definovat prostředí pro sazbu vět, definic, příkladů, ... a jejich číslování. K tomuto účelu doporučujeme balík amsthm ([17]). Umožňuje volbu typu písma v hlavičce a těle příslušného objektu. Typický příklad použití:

```
\usepackage{amsthm}

\newtheorem{veta}{Věta}[section]
\newtheorem{lemma}[veta]{Lemma}

\theoremstyle{definition}
\newtheorem{definice}{Definice}

\theoremstyle{remark}
\newtheorem*{poznámka}{Poznámka}

\renewcommand{\proofname}{Důkaz}
```

Balík amsthm nabízí tři základní styly. Styl „plain“ produkuje hlavičku tučným písmem a tělo prostředí italikou, styl „definition“ hlavičku tučným písmem a tělo normálním písmem, styl „remark“ hlavičku italikou a tělo normálním písmem. Nepovinný parametr [section] zajišťuje číslování v rámci kapitol. Příkazem `\newtheorem{lemma}[veta]{Lemma}` definujeme prostředí lemma, které bude číslováno společně s větami (použili jsme nepovinný parametr [veta], kterým nastavujeme pro prostředí lemma stejný čítač jako pro prostředí veta). Pro prostředí pro důkazy proof je již předdefinováno i s čtverečkem na konec důkazu, důkazy se nečíslojí. Příkaz `\newtheorem*` definuje nečíslované prostředí. Standardně se píše číslo každého z prostředí za hlavičku, příkaz `\swapnumbers` umožňuje umístit číslo před hlavičku. K vytváření zcela nových stylů můžeme použít příkazu `\newtheoremstyle`.

Věta 4.1. *Ukázka prostředí pro věty.*

```
\begin{veta}
Ukázka prostředí pro věty.
\end{veta}
```

Lemma 4.2. *Ukázka prostředí pro lemma.*

```
\begin{lemma}
Ukázka prostředí pro lemma.
\end{lemma}
```

Věta 4.3 (Abelova). *Věta s označením.*

```
\begin{veta}[Abelova]
Věta s označením.
\end{veta}
```

Definice 1. *Ukázka definice.*

```
\begin{definice}
Ukázka definice.
\end{definice}
```

Poznámka. Ukázka poznámky.

```
\begin{poznámka}
Ukázka poznámky.
\end{poznámka}
```

Důkaz. Ukázka důkazu.

□

```
\begin{proof}
Ukázka důkazu.
\end{proof}
```

5. Křížové odkazy

V knihách, referátech a článcích jsou velice běžné křížové odkazy na obrázky, tabulky, rovnice či jiné oddíly textu.

L^AT_EX nabízí pro křížové odkazy pohodlný aparát, totiž příkazy:

```
\label{návěští}, \ref{návěští} a \pageref{návěští}
```

Zde *návěští* je jednoznačné označení vybrané uživatelem. L^AT_EX zamění `\ref` číslem takového oddílu, pododdílu, obrázku, tabulky či rovnice, jež byl označen odpovídajícím příkazem `\label`, tj. příkazem `\label`, který má jako argument stejné návěští jako příslušný odkaz `\ref`. V prostředí `figure` a `table` se `\label` musí vyskytovat až za příkazem `\caption` nebo musí být uvnitř argumentu příkazu `\caption`.

Namísto `\pageref` se vysází číslo stránky, na níž se nachází odpovídající příkaz `\label`. Číslo stránek L^AT_EX získává z pomocného souboru (`.aux`) vytvořeného v předchozím průběhu překladač programu T_EX, proto i při použití křížových odkazů je třeba překládat vstupní soubor programem T_EX několikrát.

■ Příklad 5.1.

Odkaz na tento pododdíl vypadá takto: „viz též oddíl 5.1 na straně 12.“

```
Odkaz na tento pododdíl
\label{sec:tato} vypadá takto:
\uv{viz též oddíl~\ref{sec:tato} na
straně~\pageref{sec:tato}.}
```

Část 2

Sazba matematiky, balík amsmath

Čtenáři doporučujeme před studiem této části prostudovat Kapitulu 7 Sazba matematického textu z [2]. V celém textu pracujeme s balíkem amsmath, který významně rozšiřuje možnosti matematické sazby oproti standardnímu L^AT_EXu. Balík amsmath (??) můžeme načítat s následujícími volbami:

centertags Označování vzorců používajících prostředí `split` bude uprostřed jejich výšky (implicitní nastavení).

tbtags V prostředí `split` umísťuje číslo rovnice na poslední (resp. první) řádek podle toho, jestli číslujeme rovnice na levé (resp. pravé) straně.

nosumlimits U velkých operátorů (suma, součin) budou ve vysazeném matematickém textu obory působnosti sázeny vpravo od operátoru.

sumlimits Nastavuje obvyklou konvenci psaní oborů působnosti u operátorů (implicitní volba).

intlimits Totéž co **sumlimits**, ale pro integrály.

nointlimits Opak k **intlimits** (implicitní volba).

namelimits Totéž co **sumlimits**, ale aplikuje se na operátory (`lim`, `max`, ...) (implicitní volba).

nonamelimits Opak k **namelimits**.

leqno Číslování formulí bude sázeno na levé straně vzorců.

reqno Číslování formulí bude sázeno na pravé straně vzorců (implicitní nastavení).

fleqn Rovnice nejsou centrovány, nýbrž zarovnávány vlevo.

Kteroukoli z těchto voleb můžeme zadat jako volitelný parametr příkazu `\usepackage`, např. `\usepackage[intlimits]{amsmath}`. Poslední tři volby je možno použít i přímo s příkazem `\documentclass`.

6. Základní pravidla matematické sazby

V matematickém módu se ignorují všechny mezery a nesmí se v něm objevit prázdný řádek. Implicitně je nastaven typ písma matematická kurzíva.

Sazba matematického textu je zabezpečována některým z matematických prostředí:

Prostředí `math`

Je určeno pro sazbu matematických vztahů uvnitř odstavce (běžného textu). Místo příkazů `\begin{math}` a `\end{math}` se častěji používá dvojice `\(` pro začátek a `\)` pro konec nebo pouhý T_EXovský znak `$` (pro začátek i konec).

Prostředí `displaymath`

Je určeno pro sazbu vysazených (displayed) matematických vztahů. Podobně jako u prostředí `math` lze místo příkazů `\begin{displaymath}` a `\end{displaymath}` použít ekvivalentní dvojice `\[` pro začátek a `\]` pro konce.

Prostředí `equation`

Prostředí je určeno pro sazbu vysazených matematických vztahů s automatickým číslováním. Číslo vzorce je umístěno na pravý okraj stránky v kulatých závorkách (pokud není nastaveno jinak).

■ Příklad 6.1.

$$\int_0^{\pi} \sin x \, dx = 2 \quad (2.1)$$

```
\begin{equation} \label{int1}
\int_0^{\pi} \sin x \, dx = 2
\end{equation}
```

Pokud použijeme volitelného označení pomocí `\label`, můžeme se na číslo přiřazené rovnici odkazovat pomocí `\ref`, případně `\eqref` (v druhém případě je číslo rovnice uvedeno automaticky v kulatých závorkách).

Rovnice jsou číslovány automaticky napříč celým dokumentem. Pokud chceme číslování v rámci každé kapitoly zvlášť, musíme v preambuli dokumentu uvést příkaz:

```
\numberwithin{equation}{section}
```

Použití `*` v označení prostředí (`equation*`) ruší číslování.

K označování rovnic v prostředí `equation` a `equation*` se dále používá příkaz `\tag` (viz také Příklad 10.4).

■ Příklad 6.2.

$$\int_0^{\pi} \sin x \, dx = 2 \quad (\text{Int})$$

```
\begin{equation}
\int_0^{\pi} \sin x \, dx = 2 \tag{Int}
\end{equation}
```

7. Fonty a symboly

Pouze v matematickém módu máme k dispozici následující speciální fonty (pro zpřístupnění těchto fontů je třeba použít `\usepackage{amsfonts}`, při použití balíku `amssymb` se načítá automaticky):

- „Blackboard Bold“ font — `\mathbb` je řídicí slovo, které vysází následující argument jako zdvojené písmo. V tomto fontu je pouze velká abeceda.

$\mathbb{A}, \dots, \mathbb{Z}$ `\mathbb{A}, \dots, \mathbb{Z}`

- „Gotický“ font — `\mathfrak` je řídicí slovo, které vysází následující argument gotickým písmem. Přístupná jsou malá i velká písmena.

$\mathfrak{n}, \dots, \mathfrak{P}$ `\mathfrak{n}, \dots, \mathfrak{P}`

- „Kaligrafický“ font — `\mathcal` je řídicí slovo, které vysází následující argument skriptovým fontem. Ve skriptovém fontu je pouze velká abeceda. Tento font je k dispozici přímo v L^AT_EXu.

$\mathcal{A}, \dots, \mathcal{Z}$ `\mathcal{A}, \dots, \mathcal{Z}`

Při použití balíku `eucal` s volbou `mathscr` je zároveň k dispozici i příkaz `\mathscr`, který sází jinou variantu kaligrafického fontu (tzv. Euler script).

$\mathscr{A}, \dots, \mathscr{Z}$ `\mathscr{A}, \dots, \mathscr{Z}`

7.1. Tučné matematické symboly

Příkaz `\mathbf` nemá vliv na většinu matematických symbolů. Je aplikovatelný pouze na písmena a čísla.

$\mathbf{A}, \mathbf{Pr1}$ `\mathbf{A}, \mathbf{Pr1}`
 $\Delta\Delta+\delta\delta$ `\Delta \mathbf{\Delta} \mathbf{+} \delta \mathbf{\delta}`
 δ `\delta \mathbf{\delta}`

(Všimněte si, že příkaz `\mathbf` neovlivnil znaménko plus a malé delta.)

Balík `amsbsy` (opět je načítán automaticky v rámci balíku `amsmath`) proto poskytuje dva doplňující příkazy, `\boldsymbol` a `\pmb`. Řídicí slovo `\boldsymbol` může být použito v matematickém režimu v následujících kombinacích (pokud v naší znakové sadě existuje tučná verze požadovaného symbolu):

- s malými a velkými písmeny řecké abecedy

$\boldsymbol{\alpha}, \dots, \boldsymbol{\omega}$ `\boldsymbol{\alpha}, \dots,`
 $\boldsymbol{\Gamma}, \dots, \boldsymbol{\Omega}$ `\boldsymbol{\omega}, \boldsymbol{\Gamma}, \dots,`
`\boldsymbol{\Omega}`

- s dalšími standardními znaky

A' `$A^{\boldsymbol{\prime}}$`

Pokud chceme vysadit celou formulku tučně, použijeme příkazu `\mathversion{bold}`:

$a \equiv c \pmod{\theta}$ `{\mathversion{bold}} $ a \equiv c \pmod{\theta}$`

Protože příkaz `\boldsymbol` je poměrně dlouhý, můžeme pro často používané tučné symboly nadefinovat nové příkazy:

$B_\infty + \pi B_1 \sim \mathbf{B}_\infty + \pi \mathbf{B}_1$ `\newcommand{\bpi}{\boldsymbol{\pi}}
\newcommand{\binfty}{\boldsymbol{\infty}}
$ B_{\infty} + \pi B_1 \sim
\mathbf{B}_{\binfty} \boldsymbol{+}
\bpi \mathbf{B}_{\b1} $`

Příkaz `\pmb` („poor man’s bold“) má jeden argument. Tento argument bude vysázen tučně, přičemž tento tučný tvar je vytvořen trojím přesazením téhož textu přes sebe s mírným přesahem. Je určen pro symboly, pro které neexistuje jejich tučná verze. Kvalita výstupu je v tomto případě většinou horší. Následující příklad ukazuje možné výsledky:

■ Příklad 7.1. $A_\infty + \pi A_0$

`\sim \mathbf{A}_{\boldsymbol{\infty}} \boldsymbol{+}
\boldsymbol{\pi} \mathbf{A}_{\boldsymbol{0}}
\sim \pmb{A}_{\pmb{\infty}} \pmb{+} \pmb{\pi} \pmb{A}_{\pmb{0}}`

$$A_\infty + \pi A_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0$$

Binární operátory a relace zůstanou i po aplikaci řídicího slova `\pmb` binárními operátory a relacemi, ale u velkých operátorů nebude fungovat automatické umísťování mezí. V tomto případě je třeba použít operátoru `\mathop`.

■ Příklad 7.2.

$\sum_{j < P} \prod_{\lambda} \lambda R(r_i) \quad \sum_{x_j} \prod_{\lambda} \lambda R(x_j)$ `\[\sum_{j < P}
\prod_{\lambda} \lambda R(r_i) \quad \mathop{\pmb{\sum}}_{x_j}
\mathop{\pmb{\prod}}_{\lambda} \lambda R(x_j) \]`

7.2. Tabulky matematických symbolů

V následujících tabulkách jsou uvedeny všechny symboly, které lze standardně použít v matematickém režimu.

K používání symbolů v tabulkách 2.13–2.17¹, je třeba v preambuli dokumentu vložit balík `amssymb` a samozřejmě je třeba mít nainstalovány matematické AMS fonty. Není-li tento balík

¹Tyto tabulky byly odvozeny od `symbols.tex` Davida Carlislea.

v systému nainstalován, lze ho získat na

CTAN:/tex-archive/macros/latex/packages/amslatex

Pokud např. z paměťových důvodů chcete používat jen některé z těchto symbolů, použijte příkaz `\DeclareMathSymbol`.

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>	\widetilde{A}	<code>\widetilde{A}</code>

Tabulka 2.1: Matematické akcenty

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

Tabulka 2.2: Malá řecká písmena

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Tabulka 2.3: Velká řecká písmena

\varGamma	<code>\varGamma</code>	\varDelta	<code>\varDelta</code>	\varTheta	<code>\varTheta</code>	\varLambda	<code>\varLambda</code>
\varXi	<code>\varXi</code>	\varPi	<code>\varPi</code>	\varSigma	<code>\varSigma</code>	\varUpsilon	<code>\varUpsilon</code>
\varPhi	<code>\varPhi</code>	\varPsi	<code>\varPsi</code>	\varOmega	<code>\varOmega</code>		

Tabulka 2.4: Skloněná velká řecká písmena

Následující relační operátory mají své negované protějšky. Negovaný operátor se vysází přidáním příkazu `\not` před příslušný symbol.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> či <code>\le</code>	\geq	<code>\geq</code> či <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	$\dot{=}$	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset ^a	<code>\sqsubset</code> ^a	\sqsupset ^a	<code>\sqsupset</code> ^a	\Join ^a	<code>\Join</code> ^a
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni , \owns	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq či <code>\ne</code>	<code>\neq</code> či <code>\ne</code>

Tabulka 2.5: Binární relace

^a K vysázení těchto symbolů je třeba vložit balík `latexsym`.

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleright	<code>\triangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\ast	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee , \lor	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\triangleup	<code>\bigtriangleup</code>	\triangledown	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\lhd ^a	<code>\lhd</code> ^a	\rhd ^a	<code>\rhd</code> ^a	\wr	<code>\wr</code>
\unlhd ^a	<code>\unlhd</code> ^a	\unrhd ^a	<code>\unrhd</code> ^a		

Tabulka 2.6: Binární operátory

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

Tabulka 2.7: Velké operátory

\leftarrow	<code>\leftarrow</code> či <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\rightarrow	<code>\rightarrow</code> či <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff (větší mezery)	<code>\iff</code> (větší mezery)	\leadsto	<code>\leadsto</code> ^a

Tabulka 2.8: Šipky

^a K vysázení těchto symbolů je třeba vložit balík latexsym.

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> či <code>\lbrack</code>	$]$	<code>]</code> či <code>\rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> či <code>\lbrace</code>	$\}$	<code>\}</code> či <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> či <code>\vert</code>	$\ $	<code>\ </code> či <code>\Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	. (párová neviditelná)			

Tabulka 2.9: Závorky

$($	<code>\lgroup</code>	$)$	<code>\rgroup</code>	\int	<code>\lmoustache</code>	$\}$	<code>\rmoustache</code>
\uparrow	<code>\arrowvert</code>	\uparrow	<code>\Arrowvert</code>	$\{$	<code>\bracevert</code>	$\}$	<code>\bracevert</code>

Tabulka 2.10: Velké závorky

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho ^a	<code>\mho</code> ^a	∂	<code>\partial</code>
$'$	<code>'</code>	\prime	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\Box ^a	<code>\Box</code> ^a	\diamond	<code>\Diamond</code> ^a
\bot	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

Tabulka 2.11: Různé symboly

^a K vysázení těchto symbolů je třeba vložit balík latexsym.

Tyto symboly lze užít i v textovém režimu.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>

Tabulka 2.12: Nematematické symboly

┌	<code>\ulcorner</code>	┐	<code>\urcorner</code>	└	<code>\llcorner</code>	┘	<code>\lrcorner</code>
---	------------------------	---	------------------------	---	------------------------	---	------------------------

Tabulka 2.13: AMS — závorky

\daleth	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>
-----------	-----------------------	-------------	------------------------	---------	--------------------	-----------	----------------------	----------	---------------------

Tabulka 2.14: AMS — řecké a hebrejské znaky

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot či \Doteq	<code>\doteqdot</code> či <code>\Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll či \llless	<code>\lll</code> či <code>\llless</code>	\ggg či \gggtr	<code>\ggg</code> či <code>\gggtr</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteqq	<code>\subseteqq</code>	\supseteqq	<code>\supseteqq</code>	\Vdash	<code>\Vdash</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\between	<code>\between</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleright	<code>\blacktriangleright</code>

Tabulka 2.15: AMS — binární relace

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\Leftrightarrow	<code>\leftleftarrows</code>	\Rrightarrow	<code>\rightrightarrows</code>	\Uparrow	<code>\upuparrows</code>
\Leftrightarrow	<code>\leftrightharrows</code>	\Rrightarrow	<code>\rightleftarrows</code>	\Downarrow	<code>\downdownarrows</code>
\Leftrightarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Uparrow	<code>\upharpoonleft</code>
\Leftrightarrow	<code>\twoheadleftarrow</code>	\Rrightarrow	<code>\twoheadrightarrow</code>	\Uparrow	<code>\upharpoonright</code>
\Leftrightarrow	<code>\leftarrowtail</code>	\Rrightarrow	<code>\rightarrowtail</code>	\Downarrow	<code>\downharpoonleft</code>
\Leftrightarrow	<code>\leftrightharpoons</code>	\Rrightarrow	<code>\rightleftharpoons</code>	\Downarrow	<code>\downharpoonright</code>
\Leftrightarrow	<code>\Lsh</code>	\Rrightarrow	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\Leftrightarrow	<code>\looparrowleft</code>	\Rrightarrow	<code>\looparrowright</code>	\rightsquigarrow	<code>\leftrightsquigarrow</code>
\Leftrightarrow	<code>\curvearrowleft</code>	\Rrightarrow	<code>\curvearrowright</code>		
\Leftrightarrow	<code>\circlearrowleft</code>	\Rrightarrow	<code>\circlearrowright</code>		

Tabulka 2.16: AMS — šipky

\nless	<code>\nless</code>	\ngtr	<code>\ngtr</code>	\varsubsetneq	<code>\varsubsetneq</code>
\lneq	<code>\lneq</code>	\gneq	<code>\gneq</code>	\varsupsetneq	<code>\varsupsetneq</code>
\nleq	<code>\nleq</code>	\ngeq	<code>\ngeq</code>	\subsetneq	<code>\subsetneq</code>
\nleqslant	<code>\nleqslant</code>	\ngeqslant	<code>\ngeqslant</code>	\supsetneq	<code>\supsetneq</code>
\lneqq	<code>\lneqq</code>	\gneqq	<code>\gneqq</code>	\nmid	<code>\nmid</code>
\lvertneqq	<code>\lvertneqq</code>	\gvertneqq	<code>\gvertneqq</code>	\nparallel	<code>\nparallel</code>
\nleqq	<code>\nleqq</code>	\ngeqq	<code>\ngeqq</code>	\nshortmid	<code>\nshortmid</code>
\lnsim	<code>\lnsim</code>	\gnsim	<code>\gnsim</code>	\nshortparallel	<code>\nshortparallel</code>
\lnapprox	<code>\lnapprox</code>	\gnapprox	<code>\gnapprox</code>	\nsim	<code>\nsim</code>
\nprec	<code>\nprec</code>	\nsucc	<code>\nsucc</code>	\ncong	<code>\ncong</code>
\npreceq	<code>\npreceq</code>	\nsucceq	<code>\nsucceq</code>	\nvdash	<code>\nvdash</code>
\precneqq	<code>\precneqq</code>	\succneqq	<code>\succneqq</code>	\nvDash	<code>\nvDash</code>
\precnsim	<code>\precnsim</code>	\succnsim	<code>\succnsim</code>	\nVdash	<code>\nVdash</code>
\precnapprox	<code>\precnapprox</code>	\succnapprox	<code>\succnapprox</code>	\nVDash	<code>\nVDash</code>
\subsetneq	<code>\subsetneq</code>	\supsetneq	<code>\supsetneq</code>	\ntriangleleft	<code>\ntriangleleft</code>
\varsubsetneq	<code>\varsubsetneq</code>	\varsupsetneq	<code>\varsupsetneq</code>	\ntriangleright	<code>\ntriangleright</code>
\subsetneq	<code>\subsetneq</code>	\supsetneq	<code>\supsetneq</code>	\ntrianglelefteq	<code>\ntrianglelefteq</code>
\subsetneqq	<code>\subsetneqq</code>	\supsetneqq	<code>\supsetneqq</code>	\ntrianglerighteq	<code>\ntrianglerighteq</code>
\nleftarrow	<code>\nleftarrow</code>	\nrightarrow	<code>\nrightarrow</code>	\nleftrightarrow	<code>\nleftrightarrow</code>
\nLeftarrow	<code>\nLeftarrow</code>	\nRightarrow	<code>\nRightarrow</code>	\nLeftrightarrow	<code>\nLeftrightarrow</code>

Tabulka 2.17: AMS — negované binární relace a šipky

$\dot{+}$	<code>\dotplus</code>	\cdot	<code>\centerdot</code>	\intercal	<code>\intercal</code>
\ltimes	<code>\ltimes</code>	\rtimes	<code>\rtimes</code>	\divideontimes	<code>\divideontimes</code>
\Cup či \doublecup	<code>\Cup</code> či <code>\doublecup</code>	\Cap or \doublecap	<code>\Cap</code> or <code>\doublecap</code>	\smallsetminus	<code>\smallsetminus</code>
\veebar	<code>\veebar</code>	\barwedge	<code>\barwedge</code>	\doublebarwedge	<code>\doublebarwedge</code>
\boxplus	<code>\boxplus</code>	\boxminus	<code>\boxminus</code>	\circleddash	<code>\circleddash</code>
\boxtimes	<code>\boxtimes</code>	\boxdot	<code>\boxdot</code>	\circledcirc	<code>\circledcirc</code>
\leftthreetimes	<code>\leftthreetimes</code>	\rightthreetimes	<code>\rightthreetimes</code>	\circledast	<code>\circledast</code>
\curlyvee	<code>\curlyvee</code>	\curlywedge	<code>\curlywedge</code>	\And	<code>\And</code>

Tabulka 2.18: AMS — binární operátory

\hbar	<code>\hbar</code>	\hslash	<code>\hslash</code>	\Bbbk	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\circledS	<code>\circledS</code>
\vartriangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
\triangledown	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\angle	<code>\angle</code>	\measuredangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

Tabulka 2.19: AMS — různé symboly

Příklad	Příkaz	Potřebný balík
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	eucal s volbou: <code>mathcal</code>
či \mathscr{ABC}	<code>\mathscr{ABC}</code>	eucal s volbou: <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	eufrak, <code>amsfonts</code> či <code>amssymb</code>
\mathbb{ABC}	<code>\mathbb{ABC}</code>	<code>amsfonts</code> či <code>amssymb</code>

Tabulka 2.20: Matematická abeceda

8. Složené symboly, oddělovače a operátory

8.1. Násobné integrály

`\iint`, `\iiint` a `\iiint` dávají znaky pro násobné integrály s upraveným mezerováním (poznamenejme, že `\int\int` má moc velké mezery mezi sebou). `\idotsint` dává dva integrální znaky s tečkami mezi nimi.

■ Příklad 8.1.

$$\iint_A f(x, y) \, dx \, dy \quad \iiint_A f(x, y, z) \, dx \, dy \, dz$$

$$\iiint_A f(w, x, y, z) \, dw \, dx \, dy \, dz \quad \int_A \cdots \int f(x_1, \dots, x_k) \, dx_1 \dots dx_k$$

```
\begin{gather*}
\iint\limits_A f(x,y)\,,\,dx\,dy\,\,\,\,\,\iiint\limits_A
f(x,y,z)\,,\,dx\,dy\,dz\,\,\,\,\,\iiint\limits_A
f(w,x,y,z)\,,\,dw\,dx\,dy\,dz\,\,\,\,\,\idotsint\limits_A f(x_1,\dots,x_k)\,,\,dx_1\dots dx_k
\end{gather*}
```

8.2. Šipky nad a pod výrazem

Vektory sázíme pomocí příkazu `\overrightarrow`.

■ Příklad 8.2.

$$\overrightarrow{AB} \qquad \qquad \qquad \$\overrightarrow{AB}$$$

Oproti standardnímu L^AT_EXu (příkazy `\overrightarrow` a `\overleftarrow`) máme k dispozici i následující „šipkové“ operátory:

■ Příklad 8.3.

$$\overrightarrow{\psi_\delta(t)E_t h} = \overrightarrow{\psi_\delta(t)E_t h}$$

$$\overleftarrow{\psi_\delta(t)E_t h} = \overleftarrow{\psi_\delta(t)E_t h}$$

$$\overleftrightarrow{\psi_\delta(t)E_t h} = \overleftrightarrow{\psi_\delta(t)E_t h}$$

```
\begin{align*}
\overrightarrow{\psi_\delta(t) E_t h}&=&
\overrightarrow{\psi_\delta(t) E_t h}&=&
\overleftarrow{\psi_\delta(t) E_t h}&=&
\overleftarrow{\psi_\delta(t) E_t h}&=&
\overleftrightarrow{\psi_\delta(t) E_t h}&=&
\overleftrightarrow{\psi_\delta(t) E_t h}
\end{align*}
```

Správná velikost šipek je volena i v indexech a exponentech, jak je vidět např. v integrálu $\int_{\overrightarrow{uv}} vt \, dt$, vysázeném pomocí `$\int_{\overrightarrow{uv}} vt \, dt$` .

8.3. „Natahovací“ šipky

Příkazy `\xleftarrow` a `\xrightarrow` produkují šipky, které se roztáhnou automaticky podle šířky odpovídajícího indexu (exponentu). Text indexu (exponentu) se zadává jako volitelný (povinný) argument příkazu:

■ Příklad 8.4.

$$A \xleftarrow{n+\mu-1} B \xrightarrow[T]{n\pm i-1} C$$

```
\begin{equation*}
A\xleftarrow{n+\mu-1}B
\xrightarrow[T]{n\pm i-1}C
\end{equation*}
```

■ Příklad 8.5.

$$0 \xleftarrow[\zeta]{\alpha} F \times \triangle[n-1] \xrightarrow{\partial_0 \alpha(b)} E^{\partial_0 b}$$

```
\[0 \xleftarrow[\zeta]{\alpha}
F\times\triangle[n-1]
\xrightarrow{\partial_0\alpha(b)}
E^{\partial_0b}
\]
```

8.4. Elipsy (tečky)

Sázíme pomocí `\dots`, píše tři tečky včetně správného umístění. Pozice (na základně řádku nebo centrované) se volí automaticky podle následujícího znaku. Vyskytují-li se tečky na konci formule, \TeX nemá potřebnou informaci pro umístění teček. V tomto případě použijeme `\dotsc` pro tečky před čárkou či středníkem, `\dotsb` pro tečky mezi binárními operátory či relacemi, `\dotsm` pro tečky v násobných symbolech a `\dotsi` pro tečky mezi integrály.

■ Příklad 8.6.

Řada A_1, A_2, \dots , součet $A_1 + A_2 + \dots$, součin $A_1 A_2 \dots$, a neurčitý integrál

$$\int_{A_1} \int_{A_2} \dots$$

Řada `A_1,A_2,\dotsc`,
součet `$A_1+A_2+\dotsb$`,
součin `$A_1A_2\dotsm$`,
a neurčitý integrál
`\[\int_{A_1}\int_{A_2}\dotsi\]`.

8.5. Akcenty v matematickém režimu

V matematickém režimu nelze užívat pro matematické symboly akcenty jako v odstavcovém režimu. Seznam příkazů pro akcenty v matematickém režimu viz [2, strana 75]. Následující příkazy pro akcenty umožňují psaní dvou akcentů nad jedním znakem i s jejich správným umístěním:

■ Příklad 8.7.

\hat{H}	\check{C}	\tilde{T}	\acute{A}	\grave{G}	\dot{D}	\ddot{D}	\breve{B}	\bar{B}	\vec{V}	<code>\[\hat{\hat{H}}\quad\check{\check{C}}\quad</code> <code>\quad</code> <code>\tilde{\tilde{T}}\quad\acute{\acute{A}}\quad</code> <code>\quad</code> <code>\grave{\grave{G}}\quad\dot{\dot{D}}\quad</code> <code>\quad</code> <code>\ddot{\ddot{D}}\quad\breve{\breve{B}}\quad</code> <code>\quad</code> <code>\bar{\bar{B}}\quad\vec{\vec{V}}\]</code>
-----------	-------------	-------------	-------------	-------------	-----------	------------	-------------	-----------	-----------	--

Používání dvojitých akcentů je komplikované a podstatně prodlužuje čas překladu, proto je vhodné často používané znaky definovat pomocí příkazu `\accentedsymbol` (nutné je předtím načíst balík `amsxtra`). Příkaz se používá podobně jako `\newcommand`:

■ Příklad 8.8.

$\hat{\hat{A}}$	$\dot{\bar{\delta}}$	<code>\accentedsymbol{\Ahathat}{\hat{\hat{A}}}</code> <code>\accentedsymbol{\dbardot}{\dot{\bar{\delta}}}</code> <code>\\$ \Ahathat \quad \dbardot \\$</code>
-----------------	----------------------	---

Poznamenejme, že takto definované znaky nemění automaticky velikost v indexech.

$\hat{\hat{A}}$	<code>\\$ \Ahathat ^{\Ahathat} \\$</code>
-----------------	---

K vytvoření menší velikosti můžeme použít zmenšení pomocí `\scriptstyle`.

$\hat{\hat{A}}$	<code>\accentedsymbol{\smallAhathat}{\scriptstyle{\hat{\hat{A}}}}</code> <code>\\$ \Ahathat ^{\smallAhathat} \\$</code>
-----------------	--

Existují i široké verze akcentů `\hat` a `\tilde` dosažitelné využitím příkazů `\widehat` a `\widetilde` s parametrem, nad nímž má být akcent umístěn.

$\widehat{xy}, \widetilde{xy}$	<code>\\$ \widehat{xy} \\$, \\$ \widetilde{xy} \\$</code>
--------------------------------	---

Protože šířka akcentů je omezená, poskytuje balík `amsxtra` jiný přístup pro extrémně dlouhé výrazy: $(AmBD)^{\wedge}$ místo \widehat{AmBD} . V balíku `amsxtra` jsou za tímto účelem definovány následující řídicí slova:

■ Příklad 8.9.

$(AmBD)^{\wedge}$	$(AmBD)^{\vee}$	<code>\begin{gather*}</code> <code>(AmBD)\sphat \quad (AmBD)\spcheck \quad</code> <code>(AmBD)\sptilde \quad (AmBD)\spdot \quad</code> <code>(AmBD)\spddot \quad (AmBD)\spdddot \quad</code> <code>(AmBD)\spbreve</code> <code>\end{gather*}</code>
$(AmBD)^{\sim}$	$(AmBD)^{\cdot}$	
$(AmBD)^{\cdots}$	$(AmBD)^{\cdots}$	
$(AmBD)^{\circ}$		

Kromě v \LaTeX u přístupných příkazů `\dot` a `\ddot` máme k dispozici i příkazy `\dddot` a `\ddddot`:

$$\dddot{Q} \quad \dddot{R} \quad \quad \$ \ \dddot{Q} \ \qquad \dddot{R} \ \$$$

8.6. Odmocniny

V standardním \LaTeX u není umístění indexů u odmocnin vždy nejvhodnější. Příkazy `\leftroot` a `\uproot` umožňují upravit pozici indexu odmocniny.

■ Příklad 8.10.

$$\sqrt[\beta]{k} \quad \sqrt[\beta]{k} \quad \quad \begin{array}{l} \backslash[\\ \sqrt[\beta]{k} \quad \backslashqquad \\ \sqrt[\leftroot{-2}\uproot{2}\beta]{k} \\ \backslash] \end{array}$$

Kladné argumenty posouvají index odmocniny doleva a nahoru, záporné doprava a dolů. Jednotky přírůstku jsou velmi malé, což je výhodné pro úpravu.

Pokud chceme mít všechny znaky odmocnin stejně veliké, použijeme `\mathstrut`. Je to neviditelný symbol, jehož výška a hloubka pod základnou je rovna maximální výšce a hloubce písmen abecedy.

■ Příklad 8.11.

$$\sqrt{a} + \sqrt{d} + \sqrt{y} + \sqrt{a} + \sqrt{d} + \sqrt{y} \quad \quad \begin{array}{l} \backslash[\ \sqrt{a} + \sqrt{d} + \sqrt{y} + \\ \sqrt{\mathstrut a} + \sqrt{\mathstrut d} + \\ \sqrt{\mathstrut y} \ \backslash] \end{array}$$

8.7. Rámování formulí

Příkaz `\boxed` umožňuje zarámovat formule (podobně jako `\fbox` s tím rozdílem, že obsah je v matematickém módu).

■ Příklad 8.12.

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))} \quad \quad \begin{array}{l} \backslash[\ \boxed{\eta \leq C(\delta(\eta) + \\ \Lambda_M(0, \delta))} \ \backslash] \end{array}$$

8.8. Složené symboly

Příkazy `\overset` a `\underset` umožňují umístit jeden symbol nad (pod) symbol jiný.

■ Příklad 8.13.

$$\overset{*}{X} \quad \underset{*}{X} \quad \overset{a}{X}_b \quad a \stackrel{\text{def}}{=} b \quad \quad \begin{array}{l} \backslash[\ \overset{*}{X} \quad \backslashqquad \\ \underset{*}{X} \quad \backslashqquad \\ \overset{a}{\underset{b}{X}} \quad \backslashqquad \\ a \ \overset{\text{def}}{=} b \\ \backslash] \end{array}$$

Pokud umístíme něco nad binární operátor nebo binární relaci, výsledek je opět binárním operátorem či binární relací (tj. zachovává si vlastnosti týkající se mezerování okolo).

Příkaz `\sideset` umožňuje umístit symbol do každého rohu velkého operátoru.

■ Příklad 8.14.

$$\prod_{k=1}^2 \prod_{l=3}^4 \sum_{0 \leq i \leq m} E_i \beta x$$

```
\[ \sideset{1^2}{3^4}\prod_k \qqquad
\sideset{}{} \sum_{0 \le i \le m}
E_i \backslash beta x \]
```

8.9. Přesahy

Pokud obsah argumentu příkazu `\smash` přesahuje nad či pod řádek, nemá tento přesah vliv na zvětšení mezery mezi řádky. Podobně můžeme použít volitelné argumenty `t` a `b`, pak se bude ignorovat přesah nad či pod řádek. Napíšeme-li `\smash[t]{\dfrac{1}{2}}`, dostaneme $\frac{1}{2}$, přičemž řádek není odsazen.

■ Příklad 8.15.

$$X_j = (1/\sqrt{\lambda_j})X'_j \quad X_j = (1/\sqrt{\lambda_j})X'_j$$

```
\[
X_j=(1/\sqrt{\smash[b]{\lambda_j}})X_j'
\qqquad X_j=(1/\sqrt{\lambda_j})X_j'
\]
```

8.10. Text uvnitř matematického prostředí

Pokud píšeme uvnitř matematického prostředí nějaký text, chápe se jako posloupnost jmen proměnných a sází se bez textového vyrovnání matematickou italicou. Abychom tomu předešli text uzavíráme do parametru příkazu `\text`.

■ Příklad 8.16.

$$f_{[x_{i-1}, x_i]} \text{ je monotonní, } i = 1, \dots, c+1$$

```
\[
f_{[x_{i-1}, x_i]} \text{\text{ je monotonní,}}
\quad i = 1, \dots, c+1
\]
```

Pokud se použije `\text` v exponentu nebo indexu, automaticky se mění jeho velikost.

8.11. Jména operátorů

Balík `amsmath` poskytuje příkazy `\DeclareMathOperator` a `\DeclareMathOperator*` pro definování nových funkcí (operátorů). Například příkaz `\DeclareMathOperator{tg}{tg}` znamená, že se `tg` bude v matematickém módu psát antikvou a okolo budou takové mezery, jaké jsou obvyklé u operátorů. Definice nových operátorů je třeba uvádět v preambuli dokumentu.

■ Příklad 8.17.

```

\DeclareMathOperator*\esssup{\ess\,sup}
\DeclareMathOperator{\meas}{meas}
\newcommand{\abs}[1]{\lvert#1\rvert}
\newcommand{\norm}[1]{\lVert#1\rVert}
\begin{align*}
\norm{f}_{\infty} &= \esssup_{x \in R^n} \abs{f(x)} \\
\meas_1\{u \in R_+^1 : f^*(u) > \alpha\} &= \\
\meas_n\{x \in R^n : \abs{f(x)} \geq \alpha\} &\quad \forall \alpha > 0.
\end{align*}

```

$$\|f\|_{\infty} = \operatorname{ess\,sup}_{x \in R^n} |f(x)|$$

$$\operatorname{meas}_1\{u \in R_+^1 : f^*(u) > \alpha\} = \operatorname{meas}_n\{x \in R^n : |f(x)| \geq \alpha\} \quad \forall \alpha > 0.$$

Rozdíl mezi verzí příkazu s hvězdičkou a bez hvězdičky je pouze v umístování mezí, jak je vidět v příkladu 8.17. Balík `amsmath` předdefinovává následující operátory: `\varlimsup`, `\varliminf`, `\varinjlim` a `\varprojlim`:

■ Příklad 8.18.

$$\begin{gather*}
\overline{\lim}_{n \rightarrow \infty} Q(u_n, u_n - u^\#) \leq 0 \\
\varliminf_{n \rightarrow \infty} |a_{n+1}| / |a_n| = 0 \\
\varinjlim (m_i^\lambda \cdot)^* \leq 0 \\
\varprojlim_{p \in S(A)} A_p \leq 0
\end{gather*}$$

```

\begin{gather*}
\varlimsup_{n \rightarrow \infty} \\
\mathcal{Q}(u_n, u_n - u^\#) \leq 0 \\
\varliminf_{n \rightarrow \infty} \\
|a_{n+1}| / |a_n| = 0 \\
\varinjlim (m_i^\lambda \cdot)^* \leq 0 \\
\varprojlim_{p \in S(A)} A_p \leq 0
\end{gather*}

```

8.12. Binární operátory

Binární operátory jsou uvedeny v tabulkách 2.6 a 2.18. V některých případech je možné považovat za binární operátor i operátor mod. Uvádíme jej níže v jeho čtyřech variantách.

- `\bmod` vysází slovo `mod` s mezerami po obou stranách.

$$f(m, n) = f(m, n \bmod m) \qquad \$f(m,n)=f(m,n\bmod m)\$$$

- `\pmod` má jeden argument a vysází nejdříve mezeru a pak v kulatých závorkách slovo `mod` spolu s argumentem (mezi slovem `mod` a argumentem je mezera).

$$n \equiv k + 1 \pmod{2m} \qquad \$n \equiv k+1 \pmod{2m}\$$$

- `\mod` je totéž co `\pmod`, ale nesází kulaté závorky.

$$n \equiv k + 1 \mod 2m \qquad \$n \equiv k+1 \mod{2m}\$$$

- `\pod` je totéž co `pmod`, ale nevysází slovo `mod`.

$$n \equiv k + 1 \pmod{2m} \qquad \$n \equiv k+1 \pod{2m}$$$

■ Příklad 8.19.

$$\begin{aligned} & \gcd(k, l \bmod k) \\ u & \equiv v + 1 \pmod{n^2} \\ u & \equiv v + 1 \bmod n^2 \\ u & \equiv v + 1 \pmod{n^2} \end{aligned} \qquad \begin{aligned} & \begin{aligned} & \backslash\begin{equation*} \\ & \backslash\gcd(k~, l \backslash\bmod k) \\ & \backslash\end{equation*} \\ & \backslash\begin{align*} \\ & u\sim& \backslash\equiv v\sim+ 1 \backslash\pmod{n^2} \\ & u\sim& \backslash\equiv v\sim+ 1 \backslash\bmod{n^2} \\ & u\sim& \backslash\equiv v\sim+ 1 \backslash\pod{n^2} \\ & \backslash\end{align*} \end{aligned} \end{aligned}$$

8.13. Zlomky a související konstrukce

Kromě standardního `\frac` poskytuje balík `amsmath` i příkazy `\dfrac` a `\tfrac` jako obvyklé zkratky pro `{\displaystyle\frac ... }` a `{\textstyle\frac ... }`.

■ Příklad 8.20.

$$\frac{1}{k} \log_2 c(f) - \frac{1}{k} \log_2 c(f) \qquad \begin{aligned} & \backslash[\backslash\frac{1}{k} \backslash\log_2 c(f) \backslash\quad \\ & \backslash\tfrac{1}{k} \backslash\log_2 c(f) \quad \backslash] \\ & a \\ & \$ \backslash\sqrt{\backslash\frac{1}{k} \backslash\log_2 c(f)} \backslash\quad \\ & \backslash\sqrt{\backslash\dfrac{1}{k} \backslash\log_2 c(f)} \$. \end{aligned}$$

Pokud chceme delší zlomkovou čáru, vsuneme například do čitatele i jmenovatele z obou stran úzkou mezeru o velikosti `\,`.

■ Příklad 8.21.

$$\frac{a}{b} = \frac{\frac{a}{c}}{\frac{b}{c}} \qquad \begin{aligned} & \backslash[\\ & \backslash\frac{ab= \\ & \backslash\dfrac{\backslash,\backslash\frac{ac\backslash,}{\backslash},\backslash\frac{bc\backslash,}{\backslash}, \\ & \backslash] \end{aligned}$$

Pro binomické výrazy typu $\binom{n}{k}$ jsou v balíku `amsmath` příkazy `\binom`, `\dbinom` a `\tbinom`. Co se velikosti týká, je to obdobné jako u zlomků.

■ Příklad 8.22.

$$2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2} \qquad \begin{aligned} & \backslash\begin{equation*} \\ & 2^k - \backslash\binom{k}{1} 2^{\backslash{k-1}} + \\ & \backslash\binom{k}{2} 2^{\backslash{k-2}} \\ & \backslash\end{equation*} \end{aligned}$$

■ Příklad 8.23.

$$\binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2}$$

a $\binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2}.$

```

\begin{equation*}
\binom{k}{1}2^{k-1}+\tbinom{k}{2}2^{k-2}
\end{equation*}
a
$\binom{k}{1}2^{k-1}+
\dbinom{k}{2}2^{k-2}$.
```

Možnosti příkazů `\frac`, `\binom` a jejich variant v sobě obsahuje obecnější příkaz `\genfrac` se šesti argumenty. Poslední dva odpovídají čitateli a jmenovateli příkazu `\frac`, první dva určují volitelné oddělovače, třetí určuje tloušťku zlomkové čáry a čtvrtý ovlivňuje velikost znaků: celočíselné hodnoty 0 – 3 odpovídají postupně nastavení `\displaystyle`, `\textstyle`, `\scriptstyle` a `\scriptscriptstyle`.

`\genfrac{left-delim}{right-delim}{thickness}{mathstyle}{numerator}{denominator}`

■ Příklad 8.24.

$$\left[\frac{M(y+u) - M(y) - CM(y)u}{\|u\|} \right]$$

```

\[
\genfrac{[]{}{1.3pt}{}
{M(y+u)-M(y)-CM(y)u}{\|u\|}
\]
```

Pro ukázkou uvádíme, jak je možné definovat příkazy `\frac`, `\tfrac` a `\binom`.

```

\newcommand{\frac}[2]{\genfrac{}{}{}{}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{(){}{0pt}{}{#1}{#2}}
```

8.14. Řetězové zlomky

Pro řetězové zlomky se používá konstrukce `\cfrac`.

■ Příklad 8.25.

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$$

```

\begin{equation*}
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+\cdots}
}}}
\end{equation*}
```

Posunu výrazu v čitatelích zlomků vlevo nebo vpravo dosáhneme příkazy `\cfrac[r]` a `\cfrac[l]`, výrazy ve jmenovateli jsou vždy centrovány.

$$\frac{1}{a_1 + \frac{2}{a_2 + \cdots + \frac{n}{a_n}}} \quad \frac{1}{a_1 + \frac{2}{a_2 + \cdots + \frac{n}{a_n}}}$$

```

\begin{equation*}
\cfrac[1]{1}{a_1+
\cfrac[1]{2}{a_2+\cdots +
\cfrac[1]{n}{a_n}}}}
\quad
\cfrac[r]{1}{a_1+
\cfrac[r]{2}{a_2+\cdots +
\cfrac[r]{n}{a_n}}}}
\end{equation*}

```

8.15. Oddělovače

Příkazy `\left` a `\right` dosazují správnou velikost oddělovačů:

■ Příklad 8.26.

$$\left(\frac{1}{1-x^2} \right)^2$$

```

\[
\left( \frac{1}{1-x^2} \right)^2
\]

```

`\left` je vždy otevírací a `\right` uzavírací, proto za každým `\left` musí existovat párové `\right`. Pokud nechceme některou ze závorek uvést, musíme použít tečku místo této závorky. Např.

■ Příklad 8.27.

$$\left\{ \frac{x}{y} \right.$$

```

\[\left\{ \dfrac{xy}{\right.\}
\]

```

Podle `\left` a `\right` se určuje i mezerování okolo. Protože znaky `[a]` T_EX považuje za znaky a nikoliv za závorky, je nutné tam, kde jsou ve významu závorek, je psát s `\left` a `\right` i v normální velikosti. Analogicky i při `|a|` pokud je používáme v smyslu závorek. Automatické určování velikosti oddělovače pomocí `\left` a `\right` má ale omezení, určování velikosti oddělovače je aplikováno mechanicky a rozsah velikostí není dostatečně „spojitý“. Může se tedy stát, že závorka (oddělovač), který T_EX sám vybere je větší (menší), než bychom chtěli. V tom případě máme možnost explicitně specifikovat velikost závorky (oddělovače) v předdefinovaných velikostech: `\big`, `\Big` (asi 1,5 krát větší jako `\big`), `\bigg` a `\Bigg` (asi 2,5 krát větší jako `\bigg`), viz. tabulka 2.21. Např.

Velikost oddělovače	text	<code>\left</code> <code>\right</code>	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
Výsledek	$(b)(\frac{c}{d})$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$

Tabulka 2.21: Velikost oddělovačů

8.16. Mezery v matematickém módu

Balík `amsmath` rozšiřuje množinu příkazů pro mezery v matematickém módu, viz. tabulka 2.22. Příkazy je možno použít i mimo matematický mód. Pro mezery v matematickém módu

Zkratka	Příklad	Zkratka	Příklad
	bez mezery $\Rightarrow\Leftarrow$		bez mezery $\Rightarrow\Leftarrow$
<code>\,</code>	<code>\thinspace</code> $\Rightarrow\Leftarrow$	<code>\!</code>	<code>\negthinspace</code> $\Rightarrow\Leftarrow$
<code>\:</code>	<code>\medspace</code> $\Rightarrow\Leftarrow$		<code>\negmedspace</code> $\Rightarrow\Leftarrow$
<code>\;</code>	<code>\thickspace</code> $\Rightarrow\Leftarrow$		<code>\negthickspace</code> $\Rightarrow\Leftarrow$
	<code>\quad</code> $\Rightarrow\Leftarrow$		
	<code>\qquad</code> $\Rightarrow\Leftarrow$		

Tabulka 2.22: Mezery v matematickém módu

můžeme použít i příkazu `\mspace` „matematické jednotky“ μ ($=1/18$ em, závisí na použitém fontu). Tj. negativní `\quad` vytvoříme příkazem `\mspace{-18.0mu}`.

Všimněte si rozdílů:

■ Příklad 8.33.

$f(1/\sqrt{n})$ srovnej s $f(1/\sqrt{n})$

```
$f(1 / \sqrt{n})$ srovnej
s $f(1 / \sqrt{n}\,)$
```

■ Příklad 8.34.

$f : A \rightarrow B$ srovnej s $f : A \rightarrow B$

```
$f: A \to B$ srovnej
s $f \colon A \to B$
```

Správné mezerování poskytuje příkaz `\colon`.

■ Příklad 8.35.

$|-f(x)|$ srovnej s $|-f(x)|$

```
$|-f(x)|$ srovnej
s $\left|-f(x)\right|$
```

Příkaz `\phantom`

Příkaz `` vynechá volné místo potřebné k vysazení jeho argumentu.

■ Příklad 8.36.

$$A = \begin{pmatrix} 1 & 3 & 1 \\ 2 & 1 & 1 \\ -2 & 2 & -1 \end{pmatrix}$$

```
\[
A = \begin{pmatrix}
\phantom{-}1 & \phantom{-}3 & \phantom{-}1\\
\phantom{-}2 & \phantom{-}1 & \phantom{-}1\\
-2 & \phantom{-}2 & -1\\
\end{pmatrix}
\]
```

■ Příklad 8.37.

$$\begin{array}{rcl} a + b + c + d & = & 0, \\ c + d + e & = & 5. \end{array}$$

```
\begin{align*}
a + b + c &+ d \phantom{+ e} = 0, \\
c &+ d + e = 5.
\end{align*}
```

9. Matice

Prostředí pro sazbu matic je podobné standardnímu `array` v \LaTeX u (viz. [2]), nemusí se ale specifikovat formát sloupců. Defaultní nastavení je do 10-ti centrovaných sloupců. (Pokud chcete jiné zarovnání v rámci sloupců, musíte použít prostředí `array`.) Jednotlivé řádky se oddělují pomocí `\\`, jednotlivé položky v řádku pomocí `&`. (Pokud by v některém řádku bylo méně položek než v ostatních, řádek se doplní zprava mezerami.)

■ Příklad 9.1.

$$\begin{array}{cccc} a + b + c & uv & x - y & 27 \\ a + b & u + v & z & 134 \end{array}$$

```
\begin{equation*}
\begin{matrix}
a + b + c & uv & x - y & 27 \\
a + b & u + v & z & 134
\end{matrix}
\end{equation*}
```

Maximální počet sloupců je určen čítačem `MaxMatrixCols`, jehož hodnotu měníme standardním způsobem:

■ Příklad 9.2.

$$\begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 2 & 3 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 11 & 12 \end{array}$$

```
\begin{equation*}
\setcounter{MaxMatrixCols}{12}
\begin{matrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 11 & 12
\end{matrix}
\end{equation*}
```

Příkazem `\hdotsfor{7}` jsme nastavili řádek teček v matici, argument příkazu určuje počet sloupců, ve kterých budou tečky. Pro změnu vzdálenosti mezi tečkami můžeme použít volitelný parametr v hranatých závorkách, např. `\hdotsfor[1.5]{3}`. Číslo v hranatých závorkách se používá jako násobitel, defaultní hodnota je 1.

Pro sazbu malých matic vhodných pro použití uprostřed textu používáme prostředí `smallmatrix`.

■ Příklad 9.3.

Abychom ukázali efekt použití tohoto prostředí, umístíme matici na toto místo: $\begin{pmatrix} a+b+c & uv \\ a+b & c+d \end{pmatrix}$ a obklopíme ji dalším textem tak, aby byl alespoň jeden řádek pod maticí.

Abychom ukázali efekt použití tohoto prostředí, umístíme matici na toto místo: `\begin{math}`
`\left(`
`\begin{smallmatrix}`
`a + b + c & uv \\`
`a + b & c + d`
`\end{smallmatrix}`
`\right)`
`\end{math}`
a obklopíme ji dalším textem tak, aby byl alespoň jeden řádek pod maticí.

Následující příklad ukazuje efekt použití prostředí `matrix`, `pmatrix`, `bmatrix`, `vmatrix`, `Vmatrix` a `Bmatrix`.

■ Příklad 9.4.

$$\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
\[
\begin{matrix} a & b \\ c & d \end{matrix} \quad
\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad
\begin{bmatrix} a & b \\ c & d \end{bmatrix}
```

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\| \quad \left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\}$$

```
\[
\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad
\left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\| \quad
\left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\}
```

Na závěr další příklady použití:

■ Příklad 9.5.

$$\begin{pmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \\ \vdots & \vdots & \ddots & \vdots \\ d_1 & d_2 & \dots & d_n \end{pmatrix}$$

```
\begin{equation*}
\begin{pmatrix}
a_1 & a_2 & \dots & a_n \\
b_1 & b_2 & \dots & b_n \\
\vdots & \vdots & \ddots & \vdots \\
d_1 & d_2 & \dots & d_n
\end{pmatrix}
```

■ Příklad 9.6.

$$\begin{pmatrix} D_1 t & -a_{12} t_2 & \dots & -a_{1n} t_n \\ -a_{21} t_1 & D_2 t & \dots & -a_{2n} t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1} t_1 & -a_{n2} t_2 & \dots & D_n t \end{pmatrix}$$

```
\begin{equation*}
\begin{pmatrix}
D_1 t & -a_{12} t_2 & \dots & -a_{1n} t_n \\
-a_{21} t_1 & D_2 t & \dots & -a_{2n} t_n \\
\dots & \dots & \dots & \dots \\
-a_{n1} t_1 & -a_{n2} t_2 & \dots & D_n t
\end{pmatrix}
\end{equation*}
```

■ Příklad 9.7.

$$\left\| \begin{array}{ccccc} \frac{\varphi}{(\varphi_1, \varepsilon_1)} & 0 & \dots & 0 & 0 \\ \frac{\varphi k_{n2}}{(\varphi_2, \varepsilon_1)} & \frac{\varphi}{(\varphi_2, \varepsilon_2)} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\varphi k_{n1}}{(\varphi_n, \varepsilon_1)} & \frac{\varphi k_{n2}}{(\varphi_n, \varepsilon_2)} & \dots & \frac{\varphi k_{nn-1}}{(\varphi_n, \varepsilon_{n-1})} & \frac{\varphi}{(\varphi_n, \varepsilon_n)} \end{array} \right\|$$

```
\[
\begin{Vmatrix}
\dfrac{\varphi}{(\varphi_1, \varepsilon_1)} & 0 & \dots & 0 & 0 \\
\dfrac{\varphi k_{n2}}{(\varphi_2, \varepsilon_1)} & \dfrac{\varphi}{(\varphi_2, \varepsilon_2)} & \dots & 0 & 0 \\
\dots & \dots & \dots & \dots & \dots \\
\dfrac{\varphi k_{n1}}{(\varphi_n, \varepsilon_1)} & \dfrac{\varphi k_{n2}}{(\varphi_n, \varepsilon_2)} & \dots & \dfrac{\varphi k_{nn-1}}{(\varphi_n, \varepsilon_{n-1})} & \dfrac{\varphi}{(\varphi_n, \varepsilon_n)}
\end{Vmatrix}
\]
```

■ Příklad 9.8.

$$\left(\begin{array}{ccc|ccc} 3 & -4 & 5 & 1 & 0 & 0 \\ 2 & -3 & 1 & 0 & 1 & 0 \\ 3 & -5 & -1 & 0 & 0 & 1 \end{array} \right)$$

```
\[
\left(
\begin{array}{ccc|ccc}
3 & -4 & 5 & 1 & 0 & 0 \\
2 & -3 & 1 & 0 & 1 & 0 \\
3 & -5 & -1 & 0 & 0 & 1
\end{array}
\right)
\]
```

9.1. Větvení

Prostředí `cases` dělá větvení:

■ Příklad 9.9.

$$P_{r-j} = \begin{cases} 0 & \text{pro } r-j \text{ liché,} \\ r! (-1)^{(r-j)/2} & \text{pro } r-j \text{ sudé.} \end{cases}$$

```
\begin{equation*}
P_{r-j}=
\begin{cases}
0& \text{\text{pro } $r-j$ liché},\\
r! \, , (-1)^{(r-j)/2}& \text{\text{pro } $r-j$ sudé}.
\end{cases}
\end{equation*}
```

Každý řádek v prostředí `cases` má dvě části, které jsou odděleny znakem `&`. První i druhé části každého řádku jsou srovnány pod sebe podle svých levých okrajů. Automaticky se sází levá složená závorka, jejíž výška je určena počtem řádků.

9.2. Víceřádkové indexy a exponenty

Příkaz `\substack` používáme pro sazbu několika řádků indexů nebo exponentů, `\!` používáme pro oddělování řádků. Tento příkaz můžeme použít všude tam, kde bychom použili obvyklý index nebo exponent.

■ Příklad 9.10.

$$\sum_{\substack{0 \leq i \leq m \\ 0 \leq j < n}} P(ij)$$

```
\begin{equation*}
\sum_{\substack{
0 \leq i \leq m \\
0 < j < n
}}
P(ij)
\end{equation*}
```

Obecnější je použití prostředí `subarray`, které umožní specifikaci zarovnávání jednotlivých řádků.

■ Příklad 9.11.

$$\sum_{\substack{i \in \Lambda \\ 0 \leq j < n}} P(ij)$$

```
\[
\sum_{\begin{subarray}{l}
i \in \Lambda \\
0 \leq j < n
\end{subarray}}
P(ij)
\]
```

10. Rovnice a vzorce na více řádků

Balík `amsmath` definuje několik nových prostředí pro sazbu vysazených (displayed) matematických vztahů přes několik řádků. Na rozdíl od prostředí `eqnarray` (viz. [2]) mají tato nová prostředí rozdílný přístup k označování míst zarovnávání: zatímco `eqnarray` je podobné prostředí `array` s parametry `{rcl}` a používá tedy dva znaky `&` (ampersand) k oddělení částí, které se mají zarovnávat, v strukturách definovaných v balíku `amsmath` označujeme pouze místo (místa), které se budou zarovnávat. Ampersand umísťujeme nalevo od znaku, který se má zarovnávat se znakem na předcházejících (následujících) řádcích. Struktury definované balíkem `amsmath` mají korektně řešeno mezerování kolem bodů zarovnání, zatímco prostředí `eqnarray` přidává kolem zarovnávacích bodů nadbytečné místo. Rozdíl ukazuje následující příklad.

■ Příklad 10.1.

$x^2 + y^2 = z^2$	(2.2)	<pre>\begin{align} x^2+y^2 &= z^2 \\ x^3+y^3 &< z^3 \end{align}</pre>
$x^3 + y^3 < z^3$	(2.3)	<pre>\begin{eqnarray} x^2+y^2 &= & z^2 \\ x^3+y^3 &< & z^3 \end{eqnarray}</pre>
$x^2 + y^2 = z^2$	(2.4)	
$x^3 + y^3 < z^3$	(2.5)	

10.1. Prostředí `align`

Prostředí `align` slouží k zarovnávání formulek pod sebe.

■ Příklad 10.2.

$x = y + z,$	(2.6)	<pre>\begin{align} \label{E:m11} x &= y + z, \\ u &= v + w. \end{align}</pre>
$u = v + w.$	(2.7)	<pre>\label{E:m11a}</pre>

Jednotlivé formule jsou odděleny `\\`. Místa, označená znakem `&` budou srovnávána pod sebou. Konstrukce využívá celé šířky stránky a po srovnání všech řádků pod sebou bude výsledek centrován vůči svislé ose stránky. Všimněte si také, že `align` je již samo o sobě matematickým prostředím.

Nečíslovaná verze:

■ Příklad 10.3.

$x^2 + y^2 = 1$		<pre>\begin{align*} x^2+y^2 &= 1 \\ x &= \sqrt{1-y^2} \end{align*}</pre>
$x = \sqrt{1 - y^2}$		

Každé prostředí, s výjimkou prostředí `split`, má jak číslovanou (bez hvězdičky), tak nečíslovanou (s hvězdičkou) formu. Jednotlivé číslo rovnice (řádku) můžeme zrušit příkazem `\notag`, případně můžeme označení nahradit vlastním užitím

`\tag{label}, \tag*{label}`

kde *label* může být libovolný text, který chceme použít pro označení rovnice. Verze příkazu s hvězdičkou `\tag*` způsobí, že *label* je vysázeno samostatně bez kulatých závorek okolo. Příkazy `\tag` i `\tag*` mohou být použity i v prostředích s hvězdičkou.

■ Příklad 10.4.

$x^2 + y^2 = z^2$	(2.8)	<code>\begin{align}</code>
$x^3 + y^3 = z^3$		<code>x^2+y^2 &= z^2 \label{eq:r2} \\</code>
$x^4 + y^4 = r^4$	(*)	<code>x^3+y^3 &= z^3 \notag \\</code>
$x^5 + y^5 = r^5$	*	<code>x^4+y^4 &= r^4 \tag{**} \\</code>
$x^6 + y^6 = r^6$	(2.8')	<code>x^5+y^5 &= r^5 \tag*{***} \\</code>
		<code>x^6+y^6 &= r^6 \tag{\ref{eq:r2}\$'} \\</code>
		<code>\end{align}</code>

Všimněte si efektu použití příkazů `\label` a `\ref` v předcházejícím příkladu.

■ Příklad 10.5.

$$h(x) = \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \quad (2.9)$$

$$= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2)$$

```
\begin{align} \label{E:ml2}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \right. \\
&\quad \left. \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\
&= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2) \notag
\end{align}
```

Srovnávat můžeme i na více místech pod sebou:

■ Příklad 10.6.

$x^2 + y^2 = 1$	$x^3 + y^3 = 1$	<code>\begin{align*}</code>
$x = \sqrt{1 - y^2}$	$x = \sqrt[3]{1 - y^3}$	<code>x^2 + y^2 &= 1 &</code>
		<code>x^3 + y^3 &= 1 \\</code>
		<code>x &= \sqrt{1 - y^2} &</code>
		<code>x &= \sqrt[3]{1 - y^3} &</code>
		<code>\end{align*}</code>

Prostředí `flalign` je variantou prostředí `align` s přidáním volného místa mezi zarovnávané sloupce.

■ Příklad 10.7.

$$\begin{array}{rcl}
 x^2 + y^2 = 1 & x^3 + y^3 = 1 & \\
 x = \sqrt{1 - y^2} & x = \sqrt[3]{1 - y^3} &
 \end{array}$$

```

\begin{flalign*}
x^2 + y^2 &= 1 & & \\
x^3 + y^3 &= 1 & & \\
x &= \sqrt{1 - y^2} & & \\
x &= \sqrt[3]{1 - y^3} & & 
\end{flalign*}

```

Zarovnejte vzorec

■ Příklad 10.8.

$$x_1 + y_1 + \left(\sum_{i < 5} \binom{5}{i} + a^2 \right)^2$$

se

$$\left(\sum_{i < 5} \binom{5}{i} + \alpha^2 \right)^2$$

tak, aby se $+a^2$ v prvním vzorci zarovnálo s $+\alpha^2$ ve vzorci druhém. Pokud použijeme konstrukce

```

\begin{align}
x_{1} + y_{1} + \left( \sum_{i < 5} \binom{5}{i} \right. \\
&\quad \left. + a^{2} \right)^2 \\
\left( \sum_{i < 5} \binom{5}{i} + \alpha^{2} \right)^2 \\
\end{align}

```

dostáváme chybové hlášení:

```

! Extra }, or forgotten \right.
<template> }
$}\ifmeasuring@ \savefieldlength@ \fi \set@field \endtemplate
1.140 \end{align}

```

protože \LaTeX není schopen vysadit „podvýraz“

```
x_{1} + y_{1} + \left( \sum_{i < 5} \binom{5}{i}
```

stojící před $\&$. Pro zarovnání formulí využijeme příkazu `\phantom`:

$$\begin{array}{c}
 x_1 + y_1 + \left(\sum_{i < 5} \binom{5}{i} + a^2 \right)^2 \\
 \left(\sum_{i < 5} \binom{5}{i} + \alpha^2 \right)^2
 \end{array}$$

10.2. Prostředí alignat

Tato konstrukce umožňuje srovnávat vzorce do více sloupců. Prostředí má jeden povinný parametr, který specifikuje počet sloupců. (Pro zadané n je $2n - 1$ počet ampersandů na řádek.) V každém řádku pak jsou jednak znakem & odděleny jednotlivé sloupce, jednak jsou v každém sloupci znakem & vyznačena místa, která mají být pod sebou srovnána. Konstrukce využívá celé šířky stránky a je centrována vůči svislé ose stránky.

Prostředí `xalignat` a `xxalignat` přidávají doplňující volné místo mezi zarovnávané struktury. `xalignat` rozprostře celou konstrukci přes celou šířku stránky, přičemž od okraje stránky je také vynechané místo. `xxalignat` je totéž co `xalignat`, ale na levém a pravém okraji celých řádků není vynechané místo. V tomto případě tedy čísla rovnic nedávají smysl a proto nejsou generovány.

■ Příklad 10.9.

$L_1 = R_1$	$L_2 = R_2$	(2.10)	$\begin{array}{l} \backslash\begin{array}{l} \text{alignat}\end{array}\{2\} \\ L_1 \& = R_1 \& \quad L_2 \& = R_2 \quad \backslash\backslash \\ L_3 \& = R_3 \& \quad L_4 \& = R_4 \\ \end{array}$
$L_3 = R_3$	$L_4 = R_4$	(2.11)	
$L_1 = R_1$	$L_2 = R_2$	(2.12)	$\begin{array}{l} \backslash\begin{array}{l} \text{xalignat}\end{array}\{2\} \\ L_1 \& = R_1 \& \quad L_2 \& = R_2 \quad \backslash\backslash \\ L_3 \& = R_3 \& \quad L_4 \& = R_4 \\ \end{array}$
$L_3 = R_3$	$L_4 = R_4$	(2.13)	
$L_1 = R_1$	$L_2 = R_2$		$\begin{array}{l} \backslash\begin{array}{l} \text{xxalignat}\end{array}\{2\} \\ L_1 \& = R_1 \& \quad L_2 \& = R_2 \quad \backslash\backslash \\ L_3 \& = R_3 \& \quad L_4 \& = R_4 \\ \end{array}$
$L_3 = R_3$	$L_4 = R_4$		

Nejčastěji se `alignat` používá pro konstrukce podobné této

■ Příklad 10.10.

$$\begin{aligned} x &= x \wedge (y \vee z) && \text{by distributivity,} \\ &= (x \wedge y) \vee (x \wedge z) && \text{by Condition (M),} \\ &= y \vee z \end{aligned} \tag{2.14}$$

$$\begin{aligned} x \wedge (y \vee z) &\quad \text{\textit{by distributivity,}} \\ &= (x \wedge y) \vee (x \wedge z) \\ &\quad \text{\textit{by Condition (M),}} \\ &\quad \text{\notag} \\ &= y \vee z^{\sim} \text{\notag} \end{aligned}$$

10.3. Prostředí `gather`

Umožňuje vysázet více vzorců pod sebe tak, že každý vzorec je na jednom řádku a je centrován vůči svislé ose stránky. Konstrukce využívá celé šířky stránky. Vertikální mezery mezi vzorci jsou menší, než by byly při uvozování a ukončování každého řádku dvojicí `\[a \]`. Řádky jsou oddělovány `\\`, v žádném řádku se nesmí objevit `&`.

■ Příklad 10.11.

$$\begin{array}{ll} (a+b)^2 = a^2 + 2ab + b^2 & (2.15) \\ (a+b) \cdot (a-b) = a^2 - b^2 & (2.16) \end{array}$$

Ještě jeden příklad (první a třetí řádek nebudou číslovány):

■ Příklad 10.12.

$$\begin{array}{l} x_1 x_2 + x_1^2 x_2^2 + x_3, \\ x_1 x_3 + x_1^2 x_3^2 + x_2, \\ x_1 x_2 x_3. \end{array} \quad (2.17)$$

```
\begin{gather}
x_{\{1\}} x_{\{2\}} + x_{\{1\}}^{\{2\}} x_{\{2\}}^{\{2\}} + x_{\{3\}}, \notag \\
x_{\{1\}} x_{\{3\}} + x_{\{1\}}^{\{2\}} x_{\{3\}}^{\{2\}} + x_{\{2\}}, \label{E:ml9} \\
x_{\{1\}} x_{\{2\}} x_{\{3\}}. \notag
\end{gather}
```

10.4. Prostředí `multline`

Tato konstrukce je užitečná, je-li potřeba rozdělit jeden vzorec na více řádků, přičemž není kladen důraz na srovnání těchto částí pod sebou, ale je snaha o co nejlepší čitelnost. Předpokládá se, že v této konstrukci budou alespoň dva řádky. První řádek bude vysázen co nejvíce vlevo, poslední co nejvíce vpravo a zbývající řádky budou centrovány vzhledem ke svislé ose stránky. Kromě toho je možné kterýkoliv řádek posunout doleva (zarovnat s prvním) nebo doprava (zarovnat s posledním) pomocí příkazů `\shoveleft` a `\shoveright`. Také je možné měnit vzdálenost odsazení první a poslední formule od okrajů textu pomocí hodnoty `\multlinegap`. Prostředí `multline` využívá celé šířky stránky.

■ Příklad 10.13.

$$\begin{array}{ll} \text{První řádek rovnice} & \\ \text{Druhý řádek rovnice} & \\ \text{Třetí řádek rovnice} & \\ \text{Čtvrtý řádek rovnice} & \\ \text{Pátý řádek rovnice} & \\ \text{Poslední řádek rovnice.} & (2.18) \end{array}$$

■ Příklad 10.14.

$$\begin{aligned}
 & (x_1x_2x_3x_4x_5x_6)^2 + \\
 & \quad (x_1x_2x_3x_4x_5 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6)^2 + \\
 & \quad (x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5)^2 \quad (2.19)
 \end{aligned}$$

```

\begin{multline}\label{E:ml13}
(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^{\{2\}} + \backslash\backslash
(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^{\{2\}} + \backslash\backslash
(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} +
x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} +
x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}})^{\{2\}}
\end{multline}

```

10.5. Prostředí split

Slouží podobně jako `multline` pro formule, které se nevleznou na jeden řádek. `split` ale umožňuje nastavit místo pro zarovnávání pomocí znaku `&`. Na rozdíl od ostatních prostředí není `split` samostatným matematickým prostředím a musí být použito v rámci jiného matematického prostředí, proto také nemá automatické číslování.

■ Příklad 10.15.

$$\begin{aligned}
 (a+b)^4 &= (a+b)^2(a+b)^2 \\
 &= (a^2+2ab+b^2)(a^2+2ab+b^2) \\
 &= a^4+4a^3b+6a^2b^2+4ab^3+b^4
 \end{aligned}$$

```

\begin{equation*}
\begin{split}
(a+b)^4 &= (a+b)^2 (a+b)^2 \backslash\backslash \\
&= (a^2+2ab+b^2)(a^2+2ab+b^2) \backslash\backslash \\
&= a^4+4a^3b+6a^2b^2+4ab^3+b^4 \backslash\backslash
\end{split}
\end{equation*}

```

Pokud při načítání balíku `amsmath` použijeme volbu `tbtags`, bude se číslo formule umísťovat na poslední (resp. první) řádek podle toho, jestli čísujeme rovnice na levé (resp. pravé) straně. Implicitní volba je `centertags`, která umísťuje číslo centrovane vzhledem k výšce konstrukce (za předpokladu, že je tam dostatek místa).

■ Příklad 10.16.

$$\begin{aligned}
 f &= (x_1x_2x_3x_4x_5x_6)^2 \\
 &= (x_1x_2x_3x_4x_5 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6)^2 \\
 &= (x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5)^2
 \end{aligned} \quad (2.20)$$

```

\begin{equation}
\begin{split}
f &= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^{\{2\}} \backslash\backslash \\
&= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + \\
&\quad x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} +
\end{split}

```

```

x_{1} x_{2} x_{4} x_{5} x_{6} +
x_{1} x_{2} x_{3} x_{5} x_{6})^2\\
&= (x_{1} x_{2} x_{3} x_{4} + x_{1} x_{2} x_{3} x_{5} +
x_{1} x_{2} x_{4} x_{5} + x_{1} x_{3} x_{4} x_{5}
+ x_{2} x_{3} x_{4} x_{5})^2
\end{split}
\end{equation}

```

10.6. Prostředí `aligned`, `gathered` a `alignedat`

Podobně jako `split` nejsou samostatnými matematickými prostředími, musí být použity v rámci jiného matematického prostředí. Pracují stejně jako `align`, `gather` a `aligned` s tím, že výsledná šířka konstrukce je určena nejširší formulí (u `gathered`) nebo nejširšími částmi srovnávacích formulí (u `aligned` a `alignedat`).

■ Příklad 10.17.

$\left. \begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned} \right\} \text{Maxwellovy rovnice}$	<pre> \begin{equation*} \left. \begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned} \right\} \text{Maxwellovy rovnice} \end{equation*} </pre>
--	---

Mají nepovinný parametr, který určuje jejich vertikální pozici vzhledem k materiálu vedle. Defaultní nastavení je centrování (`[c]`) a jeho efekt ukazuje následující příklad.

■ Příklad 10.18.

$\begin{aligned} x^2 + y^2 &= 1 & a \cdot (a + b) &= a^2 + ab \\ x &= \sqrt{1 - y^2} & b \cdot (a + b) &= ab + b^2 \end{aligned}$	<pre> \begin{equation*} \begin{aligned} x^2 + y^2 &= 1 \\ x &= \sqrt{1 - y^2} \end{aligned} \quad \quad \quad \begin{aligned} a \cdot (a + b) &= a^2 + ab \\ b \cdot (a + b) &= ab + b^2 \end{aligned} \end{equation*} </pre>
---	---

Ten samý příklad nyní s jiným vertikálním zarovnáním.

■ Příklad 10.19.

$x^2 + y^2 = 1$	$\begin{aligned} &\backslash\begin{equation*} \\ &\backslash\begin{aligned}[b] \\ x^2 + y^2 &= 1 \\ &x = \sqrt{1-y^2} \\ &\end{aligned} \\ &\end{aligned} \quad \backslashquad \\ &\begin{gathered}[t] \\ a \cdot (a+b) &= a^2 + ab \\ b \cdot (a+b) &= ab + b^2 \\ &\end{gathered} \\ &\end{equation*}$
-----------------	--

Následující konstrukce používáme tehdy, pokud chceme aby číslo formule bylo centrováno mezi dvěma řádky:

■ **Příklad 10.20.**

$$\begin{aligned} h(x) &= \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\ &= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2) \end{aligned} \quad (2.21)$$

```
\begin{equation}
\begin{aligned} \label{E:longInt2}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \right. \\
&\quad \left. \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\
&= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x-2)
\end{aligned}
\end{equation}
```

10.7. Další úpravy formulí přes více řádků

Pro zvětšení mezery mezi řádky můžeme použít v L^AT_EXu obvyklého `\[dimenze]`. Narozdíl od `eqnarray` prostředí balíku `amsmath` neumožňují stránkový zlom pokud nepoužijeme `\displaybreak` nebo `\allowdisplaybreaks`. `\displaybreak` se používá před `\`, způsobuje stránkový zlom v daném místě a podobně jako `pagebreak` v L^AT_EXu má volitelný parametr *stupeň* (0–4), podle kterého se řídí síla příkazu. Pokud chceme povolit stránkový zlom i uprostřed víceřádkových formulí, umístíme příkaz `\allowdisplaybreaks[1]` do preamble dokumentu. Volitelný argument 1–4 umožňuje jemnější nastavení. Příkaz `\allowdisplaybreaks` je možno používat i lokálně:

```
{\allowdisplaybreaks \begin{align} ... \end{align} }.
```

Při použití `\allowdisplaybreaks` máme k dispozici příkaz `*`, kterým můžeme zakázat stránkový zlom za zvoleným řádkem.

Příkazy `\displaybreak` a `\allowdisplaybreaks` neovlivňují chování prostředí `split`, `aligned`, `gathered` a `alignedat`.

10.8. Příkaz `\intertext`

Příkaz používáme pro vložení řádku (nebo dvou) textu do vysazených (displayed) matematických vztahů. Výhodou je to, že zachovává zarovnání, což by nebylo možné při zakončení displayed prostředí a jeho následném použití. Příkaz `\intertext` je možno použít pouze okamžitě za `\\` nebo `*`.

■ Příklad 10.21.

$$A_1 = N_0(\lambda; \Omega') - \phi(\lambda; \Omega'), \quad (2.22)$$

$$A_2 = \phi(\lambda; \Omega')\phi(\lambda; \Omega), \quad (2.23)$$

a konečně

$$A_3 = \mathcal{N}(\lambda; \omega). \quad (2.24)$$

```
\begin{align}
A_1&=N_0(\lambda;\Omega') - \\
&\quad \phi(\lambda;\Omega'), \\
A_2&=\phi(\lambda;\Omega') \\
&\quad \phi(\lambda;\Omega), \\
&\intertext{a konečně}
A_3&=\mathcal{N}(\lambda;\omega).
\end{align}
```

■ Příklad 10.22.

$$h(x) = \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx$$

Čtenáři se může zdát následující tvar jednodušeji čitelný:

$$= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2)$$

```
\begin{align}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\
&\intertext{Čtenáři se může zdát}
&\intertext{následující tvar jednodušeji čitelný:}
&= \int \frac{1 + f(x)}{1 + g(x)} dx - \\
&\quad 2 \tan^{-1}(x - 2)
\end{align}
```

10.9. Příklady použití prostředí pro rovnice na více řádků

Následují praktické příklady užití prostředí probraných výše.

■ Příklad 10.23.

$$\begin{aligned} f_{h,\varepsilon}(x, y) &= \varepsilon \mathbf{E}_{x,y} \int_0^{t_\varepsilon} L_{x,y_\varepsilon(\varepsilon u)} \varphi(x) du \\ &= h \int L_{x,z} \varphi(x) \rho_x(dz) \\ &\quad + h \left[\frac{1}{t_\varepsilon} \left(\mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) ds - t_\varepsilon \int L_{x,z} \varphi(x) \rho_x(dz) \right) \right. \\ &\quad \left. + \frac{1}{t_\varepsilon} \left(\mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) ds - \mathbf{E}_{x,y} \int_0^{t_\varepsilon} L_{x,y_\varepsilon(\varepsilon s)} \varphi(x) ds \right) \right] \end{aligned} \quad (2.25)$$

Všimněte si použití příkazu `\phantom` — vynechá volné místo rovné šířce jeho argumentu.

```
\begin{equation}
\begin{split}
```

```

f_{h,\varepsilon}(x,y)
&= \varepsilon \mathbf{E}_{x,y} \int_0^{t_\varepsilon}
  L_{x,y,\varepsilon}(\varepsilon u) \varphi(x) \, du \\
&= h \int L_{x,z} \varphi(x) \rho_x(z) \, dz \\
&\quad + h \biggl[ \frac{1}{t_\varepsilon} \mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) \, ds
  - \int L_{x,z} \varphi(x) \rho_x(z) \, dz \biggr] \\
&\quad + \phantom{=} h \biggl[ \frac{1}{t_\varepsilon} \mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) \, ds
  - \mathbf{E}_{x,y} \int_0^{t_\varepsilon} L_{x,y,\varepsilon}(\varepsilon s) \varphi(x) \, ds \biggr] \\
\end{split}
\end{equation}

```

Příklady 10.24 a 10.25 ukazují použití prostředí `split` a `align`.

■ Příklad 10.24.

$$\begin{aligned}
 |I_1| &= \left| \int_{\Omega} g R u \, d\Omega \right| \\
 &\leq C_3 \left[\int_{\Omega} \left(\int_a^x g(\xi, t) \, d\xi \right)^2 d\Omega \right]^{1/2} \\
 &\quad \times \left[\int_{\Omega} \left\{ u_x^2 + \frac{1}{k} \left(\int_a^x c u_t \, d\xi \right)^2 \right\} c \Omega \right]^{1/2} \\
 &\leq C_4 \left\| f \left| \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right| \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\|.
 \end{aligned} \tag{2.26}$$

$$\begin{aligned}
 |I_2| &= \left| \int_0^T \psi(t) \left\{ u(a, t) - \int_{\gamma(t)}^a \frac{d\theta}{k(\theta, t)} \int_a^\theta c(\xi) u_t(\xi, t) \, d\xi \right\} dt \right| \\
 &\leq C_6 \left\| f \int_{\Omega} \left| \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right| \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\|.
 \end{aligned} \tag{2.27}$$

```

\begin{align}
\begin{split}
|I_1| &= \left| \int_{\Omega} g R u \, d\Omega \right| \\
&\leq C_3 \left[ \int_{\Omega} \left( \int_a^x g(\xi, t) \, d\xi \right)^2 d\Omega \right]^{1/2} \\
&\quad \times \left[ \int_{\Omega} \left\{ u_x^2 + \frac{1}{k} \left( \int_a^x c u_t \, d\xi \right)^2 \right\} c \Omega \right]^{1/2} \\
&\leq C_4 \left\| f \left| \widetilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right| \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\widetilde{A}}(\Omega; \Gamma_r, T) \right\|.
\end{split}
\end{split} \tag{eq:A}
\begin{split}
\end{split}

```

```

|I_2| &= \left| \int_0^T \psi(t) \left\{ u(a,t) \right. \right. \\
&\quad \left. \left. - \int_{\gamma(t)}^a \frac{d\theta}{k(\theta,t)} \right. \right. \\
&\quad \left. \left. \int_a^{\theta} c(\xi) u_t(\xi,t) \, d\xi \right\} dt \right| \\
&\leq C_6 \left| \int_{\Omega} f \, d\Omega \right| \\
&\quad \left| \widetilde{S}^{-1,0}_{a,-} W_2(\Omega, \Gamma_1) \right| \\
&\quad \left| |u| \overset{\circ}{\rightarrow} W_2^{\widetilde{A}}(\Omega; \Gamma_r, T) \right|.
\end{split}
\end{align}

```

■ Příklad 10.25.

$$\begin{aligned}
 f &= (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 &= (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2
 \end{aligned} \tag{2.28}$$

$$\begin{aligned}
 &= (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4 x_5 + x_2 x_3 x_4 x_5)^2, \\
 g &= y_1 y_2 y_3.
 \end{aligned} \tag{2.29}$$

```

\begin{align} \label{E:ml6}
\begin{split}
f &= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^2 \\
&= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + \\
&\quad x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^2 \\
&= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} + \\
&\quad x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + \\
&\quad x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}})^2,
\end{split} \\
\label{E:ml7}
g &= y_{\{1\}} y_{\{2\}} y_{\{3\}}.
\end{align}

```

■ Příklad 10.26.

$$\begin{aligned}
 &\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2] - 2f(x)g(x)f(y)g(y) \, dx \right\} dy \\
 &= \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy
 \end{aligned} \tag{2.30}$$

```

\begin{multline} \label{eq:E1}
\int_a^b \biggl\{ \int_a^b [ f(x)^2 g(y)^2 + f(y)^2 g(x)^2 ] \\
- 2f(x) g(x) f(y) g(y) \, dx \biggr\} dy \\
= \int_a^b \biggl\{ g(y)^2 \int_a^b f^2 + f(y)^2 \\
\int_a^b g^2 - 2f(y) g(y) \int_a^b fg \biggr\} dy
\end{multline}

```

Následuje ten samý výstup, ale s `\multlinegap` nastaveným na nulu a bez číslování. Výstup byl generován s užitím:

```
{\setlength{\multlinegap}{0pt} \begin{multline*} ... \end{multline*}}
```

$$\begin{aligned} \int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2] - 2f(x)g(x)f(y)g(y) \, dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \end{aligned}$$

■ **Příklad 10.27.**

$$D(a, r) \equiv \{z \in \mathbf{C} : |z - a| < r\}, \quad (2.31)$$

$$\begin{aligned} \text{seg}(a, r) &\equiv \{z \in \mathbf{C} : \Im z = \Im a, |z - a| < r\}, \\ c(e, \theta, r) &\equiv \{(x, y) \in \mathbf{C} : |x - e| < y \tan \theta, 0 < y < r\}, \end{aligned} \quad (2.32)$$

$$C(E, \theta, r) \equiv \bigcup_{e \in E} c(e, \theta, r). \quad (2.33)$$

$$C(E, \theta, r) \equiv \bigcup_{e \in E} c(e, \theta, r). \quad (2.33)$$

$$\begin{gathered} D(a,r) \equiv \{ z \in \mathbf{C}: |z-a| < r \}, \\ \operatorname{seg}(a,r) \equiv \{ z \in \mathbf{C}: \\ \quad \operatorname{Im} z = \operatorname{Im} a, \quad |z-a| < r \}, \\ c(e,\theta,r) \equiv \{ (x,y) \in \mathbf{C}: \\ \quad |x-e| < y \tan \theta, \quad 0 < y < r \}, \\ C(E,\theta,r) \equiv \bigcup_{e \in E} c(e,\theta,r). \end{gathered}$$

■ **Příklad 10.28.**

$$\gamma_x(t) = (\cos tu + \sin tx, v), \quad (2.34)$$

$$\gamma_u(t) = (u, \cos tv + \sin ty), \quad (2.35)$$

$$\gamma_z(t) = \left(\cos tu + \frac{\alpha}{\beta} \sin tv, -\frac{\beta}{\alpha} \sin tu + \cos tv \right). \quad (2.36)$$

```
\begin{align}
\gamma_x(t) &= (\cos tu + \sin tv, v), && \\
\gamma_y(t) &= (u, \cos tv + \sin ty), && \\
\gamma_z(t) &= \left( \cos tu + \frac{\alpha}{\beta} \sin tv, \right. && \\
&\quad \left. - \frac{\beta}{\alpha} \sin tu + \cos tv \right). && \\
\end{align}
```

■ **Příklad 10.29.**

$$\begin{aligned}\varphi(x, z) &= z - \gamma_{10}x - \sum_{m+n \geq 2} \gamma_{mn}x^m z^n \\ &= z - Mr^{-1}x - \sum_{m+n \geq 2} Mr^{-(m+n)}x^m z^n \\ \zeta^0 &= (\xi^0)^2,\end{aligned}\tag{2.37}$$

$$\zeta^1 = \xi^0 \xi^1 \quad (2.38)$$

$$\zeta^1 = \xi^0 \xi^1 \quad (2.38)$$

```

\begin{gather*}
\begin{split}
\varphi(x,z)
&= z^{-\gamma_{10}} x - \sum_{m+n \geq 2} \gamma_{mn} x^m z^n \\
&= z^{-M_{-1}} x - \sum_{m+n \geq 2} M_{-1} x^m z^n
\end{split}
\end{gather*}

```

■ Příklad 10.30.

$$\begin{aligned}
 f &= (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 &= (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\
 &= (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4 x_5 + x_2 x_3 x_4 x_5)^2
 \end{aligned}$$

$$g = y_1 y_2 y_3 \quad (2.39)$$

$$h = z_1^2 z_2^2 z_3^2 \quad (2.40)$$

```

\begin{gather*} \label{E:ml11}
\begin{split}
f &= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^2 \\
&= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^2 \\
&= (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} + \\
&\quad x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + \\
&\quad x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}})^2
\end{split}
\end{gather*}

```

■ Příklad 10.31.

$$V_i = v_i - q_i v_j, \quad X_i = x_i - q_i x_j, \quad U_i = u_i, \quad \text{for } i \neq j; \quad (2.41)$$

$$V_j = v_j, \quad X_j = x_j, \quad U_j u_j + \sum_{i \neq j} q_i u_i. \quad (2.42)$$

```

\begin{alignat}{3}
V_i &= v_i - q_i v_j, & \& \quad X_i &= x_i - q_i x_j,
\end{alignat}

```



```

& \qquad U_i &= u_i, \qquad \text{for } i \neq j, \label{eq:B} \\
V_j &= v_j, \qquad \qquad \qquad & \qquad X_j &= x_j, \\
& \qquad U_j &= u_j + \sum_{i \neq j} q_i u_i. \\
\end{alignat}

```

■ **Příklad 10.32.**

$$x = y \qquad \text{podle (2.26)} \qquad (2.43)$$

$$x' = y' \qquad \text{podle (2.41)} \qquad (2.44)$$

$$x + x' = y + y' \qquad \text{podle Axiomu 1.} \qquad (2.45)$$

```

\begin{alignat}{2}
x &= y && \qquad \text{podle (\ref{eq:A})} \label{eq:C} \\
x' &= y' && \qquad \text{podle (\ref{eq:B})} \label{eq:D} \\
x + x' &= y + y' && \qquad \text{podle Axiomu 1.} \\
\end{alignat}

```

A ještě roztažená verze předcházejícího s použitím flalign:

$$x = y \qquad \text{podle (2.26)} \qquad (2.46)$$

$$x' = y' \qquad \text{podle (2.41)} \qquad (2.47)$$

$$x + x' = y + y' \qquad \text{podle Axiomu 1.} \qquad (2.48)$$

■ **Příklad 10.33.**

$$f(x) = x + yz \qquad g(x) = x + y + z$$

Důležité jsou i následující polynomy:

$$h(x) = xy + xz + yz \qquad k(x) = (x + y)(x + z)(y + z)$$

```

\begin{alignat*}{2}
f(x) &= x + yz && \qquad g(x) &= x + y + z \\
\intertext{Důležité jsou i následující polynomy:}
h(x) &= xy + xz + yz \\
&&& \qquad \qquad \qquad && \qquad k(x) &= (x + y)(x + z)(y + z) \\
\end{alignat*}

```

■ **Příklad 10.34.**

$x = 3,$ $y = 4,$ $z = 5;$	nebo	$x = 5,$ $y = 12,$ $z = 13.$	$\left[\begin{array}{l} \begin{array}{l} x = 3, \\ y = 4, \\ z = 5; \end{array} \\ \text{\quad\quad\quad nebo \quad} \\ \begin{array}{l} x = 5, \\ y = 12, \\ z = 13. \end{array} \end{array} \right]$
----------------------------------	------	------------------------------------	---

Část 3

Grafika

Jeden obrázek vydá za tisíc slov.
(příslloví)

V dnešní době se při psaní knihy, vědecké práce a často i článku nechceme zříkat obrázků, schémat či jiných grafických objektů. Grafika je běžně integrována přímo do textu a těžko věříme, že naši rodiče ještě psali diplomové práce na psacích strojích a obrázky vlepovali či rýsovali tuší.

Zbývající stránky tohoto textu budou věnovány práci s grafikou v systému \LaTeX . Pod tím budeme rozumět grafickou úpravu textu, tvorbu obrázků, schémat, diagramů ap. a integraci grafiky do textu.

Grafické objekty v \LaTeX ovském (i \TeX ovském obecně) dokumentu mohou vznikat několika způsoby:

- přímo v \LaTeX u pomocí balíků maker,
- pomocí programu METAFONT či METAPOST,
- integrací grafiky vytvořené externě speciálním grafickým systémem.

Výhodou grafiky vytvořené přímo v \LaTeX u je, že tento způsob nevyžaduje žádné další softwarové vybavení, nevýhodou určitá pracnost při vytváření obrázků. K jednoduchým obrázkům se hodí přímo okolí `picture` obsažené ve standartním \LaTeX u. Širší škálu možností poskytují balíky `pspicture`, `Xy-pic`, sada maker `PiCTEX` atd. Dále existují programy generující výstup ve formě \LaTeX ovského okolí `picture`, které se potom vloží do zdrojového dokumentu. Takovými programy jsou např. `TeXcad` nebo `gnuplot`.

Nabídka speciálních maker pro kreslení schémat obrázků ap. je široká a její zmapování přesahuje rámec tohoto textu. Výběr balíků popsanych v první části se řídil zaměřením tohoto textu na sazbu diplomové práce z matematiky. Nekladli jsme si za cíl podat u jednotlivých balíků kompletní uživatelskou příručku, ale spíše ukázat možnosti jednotlivých balíků na řadě příkladů. Tato forma zpracování by měla sloužit k tomu, aby čtenář rychle zjistil, který balík maker odpovídá jeho požadavkům, a snadno mohl použít základní příkazy. Pro náročnější je text doplněn odkazy na uživatelské příručky k příslušným balíkům v angličtině, popř. němčině.

V dnešní době je stále více upřednostňováno tvoření grafiky speciálními grafickými systémy (Publisher's Paintbrush, Corel Draw, Adobe Illustrator ap.) a to i grafiky „matematické“ či „geometrické“ (Maple, Matematika, CAD systémy ap.). Do \LaTeX u je možné integrovat

v podstatě libovolný grafický formát, ale nejjednodušší a uživatelsky nejpohodlnější je vkládat EPS (Encapsulated PostScript) formát. Tomuto tematu se budeme věnovat v druhé části (str. 69).

Pokud by se někdo přece jen rozhodl vytvářet vlastní grafiku, potom mu k tomu výborné prostředky poskytuje METAPOST (popř. vlastní METAFONT). Můžeme využívat METAFONTu či METAPOSTu, aniž bychom byli nuceni učit se nový programovací jazyk. K tomu slouží balíky `mfpic`. Zadáání obrázku má pak podobnou syntaxi jako u okolí `picture` s tím rozdílem, že je vygenerován zdrojový text pro METAFONT či METAPOST. Dalším přeložením je vysázen obrázek. Podrobněji je práce s makry `mfpic` popsána v kapitole 17.

11. Obrázky kreslené L^AT_EXem

Prostředí `picture` je součástí standardní instalace L^AT_EXu. Pokud je dobře ovládáte, zvládnete velmi rychle i nadstandardní L^AT_EXovská makra a kreslení METAPOSTem pomocí balíku `maker mfpic` (kapitola 17, str. 98). Z tohoto důvodu zařazujeme popis prostředí `picture` i do tohoto textu, ačkoliv je dosti podrobně uveden v [2]. Navíc v odstavci 11.1 uvádíme rozsáhlejší příklady, které ilustrují možnosti (i určitá omezení) tohoto prostředí.

Další odstavce obsahují popisy balíků, které v [2] čtenář nenajde. Tyto balíky jsou většinou speciálně zaměřeny (např. odstavec 11.4 popisuje balík `Xy-pic` pro kreslení algebraických diagramů).

11.1. Prostředí `picture`

Pracujeme v pravoúhlém souřadnicovém systému. Pro každý obrázek (tj. každé okolí `picture`) zavádíme vlastní souřadnice. Prostředí definujeme jako:

```
\begin{picture}(hx,hy)(dx,dy)

:

\end{picture},
```

kde `hx`, `hy` jsou souřadnice pravého horního rohu a `dx`, `dy` jsou souřadnice levého dolního rohu obdélníku, který bude systém rezervovat pro obrázek. Můžeme však umísťovat objekty i mimo tento obdélník, tj. např. přes okolní text. Vhodnými souřadnicemi lze tedy dosáhnout umístění požadovaného objektu do libovolného místa na stránce.

Všechny hodnoty souřadnic a jiných rozměrů uvnitř prostředí `picture` jsou chápány jako násobky jednotkové délky. Tato jednotková délka je uschována v proměnné dosažitelné pomocí délkové proměnné `\unitlength` a pokud ji nenastavíme, používá systém hodnotu `1pt`.¹ Chceme-li provést její nastavení např. na 1mm, zapíšeme příkaz: `\setlength{\unitlength}{1mm}`.

Např. zápisem `\begin{picture}(20,20)(-20,-20)` docílíme toho, že L^AT_EX bude rezervovat čtverec o velikosti 40×40 a současně zavedeme souřadný systém pro toto prostředí, který má počátek (0,0) ve středu námi definovaného čtverce. (Pokud nenařídíme obtékání okolního textu (viz kapitola 18), bude vynechán pás o výšce 40 přes celou šířku stránky.)

¹pt — tzv. „point“, $1\text{pt} \simeq 0,352\text{mm}$. Více o typografických jednotkách [2, str. 22–23].

Uvnitř prostředí `picture` se mohou objevit pouze příkazy pro vložení obrazového objektu — `\put` a `\multiput`. Základním příkazem je příkaz `\put`, který má syntaxi:

`\put (x_r, y_r){objekt}`

Souřadnice x_r a y_r jsou souřadnice referenčního bodu obrazového objektu. Objektem, který se má na pozici (x_r, y_r) umístit může být např.:

- **Text:** Text je nejjednodušším obrazovým objektem. Pomyslný obdélník obepínající text je umístěn levým dolním rohem v referenčním bodě. Lze použít všechny příkazy pro formátování textu v jednom řádku.
- **Matematický text:** Uzavřeme ho mezi znaky \$, jinak platí totéž, co pro obyčejný text.
- **Rámečky:** Lze použít tři druhy orámování:
 - plnou čarou `\framebox (šířka, výška) [pozice]{objekt}`
 - čárkovanou čarou `\dashbox{element} (šířka, výška) [pozice]{objekt}`
 - žádnou čarou `\makebox(šířka, výška) [pozice]{objekt}` (neorámovaný rámeček se velmi často užívá pro řízení pozice jiných objektů).

Parametry *šířka* a *výška* určují rozměry rámečku, parametr *pozice* určuje, v kterém místě rámečku bude umístěn referenční bod objektu vkládaného dovnitř. Parametr *pozice* je nepovinný, pokud je uveden, skládá se z jednoho až dvou písmen určujících horizontální a vertikální pozici referenčního bodu vkládaného objektu:

- t** – na horním okraji (top)
- c** – uprostřed (center)
- b** – na dolním okraji (bottom)
- l** – na levém okraji (left)
- r** – na pravém okraji (right)

Pokud parametr *pozice* vynecháme, je implicitně nastaveno **c**. Parametr *element* udává délku čárek a mezer (viz příklad 11.3).

- **Úsečky:** Počáteční bod kreslené úsečky je dán referenčním bodem. Koncový bod je dán sklonem a délkou úsečky.
 Příkaz pro vytvoření čáry má tvar `\line(x,y){délka}`. Parametry x a y udávají délky průmětů elementu do osy x a y . Obě čísla jsou celá z intervalu $\langle -6, 6 \rangle$ a nesmí mít společného dělitele > 1 . Výsledný element se pak v daném směru prodlouží podle parametru *délka*. Zadaná délka však neudává skutečnou délku úsečky, ale délku průmětu do osy x , u svislých úseček pak svislou délku (viz příklad 11.3).
- **Šipky:** Příkaz `\vector(x,y){délka}` se chová obdobně jako `\line` při kreslení úsečky. Rozdíl je v tom, že celá čísla x a y musí ležet v intervalu $\langle -4, 4 \rangle$. Záleží na směru kreslení, protože šipka je vždy v koncovém bodě. Šipka se nakreslí i v případě, že „vektor“ bude mít nulovou délku (viz příklad 11.2).

- **Kružnice:**

Kružnice: `\circle{p}`

Vyplněná kružnice: `\circle*{p}`

Kde **p** je průměr (**p** může být max. kolem 14 jednotek, pro kružnice větších průměrů je třeba použít balík `pspicture` – viz odstavec 11.3).

- **Ovály:** Ovál, přesněji obdélník o šířce x a výšce y , který má rohy nahrazeny čtvrtkružnicemi, lze nakreslit příkazem `\oval (x,y){část}`. Referenčním bodem je střed. Nepovinný parametr *část* umožňuje volbou jednoho ze čtyř písmen nakreslit levou (**l**), pravou (**r**), horní (**t**) či dolní (**b**) polovinu oválu. Kombinací dvou písmen, z nichž jedno udává horizontální a druhé vertikální polovinu, lze nakreslit čtvrtinu (viz příklad 11.1).

- **Bèziérový křivky:** Pro kreslení kvadratických Bèzierových křivek je k dispozici příkaz: `\qbezier [N](A_x, A_y)(B_x, B_y)(C_x, C_y)`

Tento příkaz vysází tečkovanou křivku z bodu A do bodu C s řídícím bodem B složenou z N teček. Pokud není nepovinný parametr N zadán, vykreslí se plná křivka (viz příklad 11.4). Pro kreslení křivek doporučujeme použít některý z nadstandartních balíků, protože zvláště v případě plné čáry nedává příkaz `\qbezier` příliš efektivní výsledek.

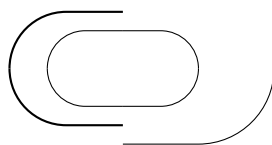
Tloušťka čar

U vodorovných a svislých čar lze nařídit libovolnou tloušťku čáry příkazem

`\linethickness{tloušťka},`

kde *tloušťka* čar je zadaná v obvyklých jednotkách. U šikmých čar, kroužků, oválů a křivek lze volit jen jednu ze dvou předdefinovaných tlouštěk. Implicitně je nastaveno 0.4pt, příkazem `\thicklines` se přepne na 0.8pt, příkazem `\thinlines` zpět na 0.4pt (viz příklad 11.4).

■ Příklad 11.1.



```
\setlength{\unitlength}{1mm}
\begin{picture}(60,30)(0,0)
\put(30,15){\oval(20,10)}
\thicklines
\put(30,15){\oval(30,15)[1]}
\thinlines
\put(30,15){\oval(40,20)[rb]}
\end{picture}
```

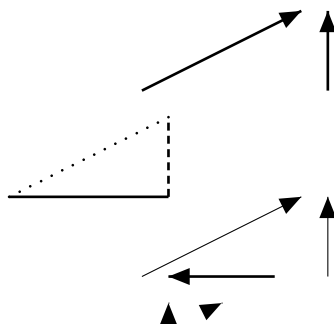
Opakování objektů

Dalším příkazem uvnitř prostředí `picture` je příkaz `\multiput`, který má tvar:

`\multiput(x_r, y_r)(x_p, y_p){počet}{objekt}{`

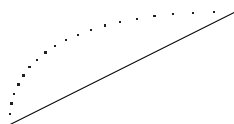
Souřadnice x_r a y_r jsou souřadnice referenčního bodu, x_p je posun v ose x , y_p je posun v ose y , *počet* je celé číslo udávající počet opakování, *objekt* je obrazový objekt, který se má opakovat.

■ Příklad 11.2.



```
\begin{picture}(120,120)(-20,-40)
\multiput(0,0)(4,2){16}{\circle*{1}}
\put(50,-30){\vector(2,1){60}}
\put(120,-30){\vector(0,1){30}}
\put(60,-40){\vector(0,1){0}}
\put(80,-40){\vector(2,1){0}}
\linethickness{1pt}
\put(50,40){\vector(2,1){60}}
\put(120,40){\vector(0,1){30}}
\put(0,0){\line(1,0){60}}
\multiput(60,0)(0,5){6}{\line(0,1){3}}
\put(100,-30){\vector(-1,0){40}}
\end{picture}
```

■ Příklad 11.3.



```
\setlength{\unitlength}{1mm}
\begin{picture}(80,40)(-10,-10)
\put(0,0){\line(2,1){30}}
\put(0,0){\line(1,0){30}}
\put(30,0){\line(0,1){15}}
\qbezier[20](0,0)(0,15)(30,15)
\put(40,10){\dashbox{2}(25,10)[b]{zde}}
\put(40,-5){\dashbox{4}(25,10)[lb]{zde}}
\put(0,-30){\framebox(20,10){}}
\end{picture}
```

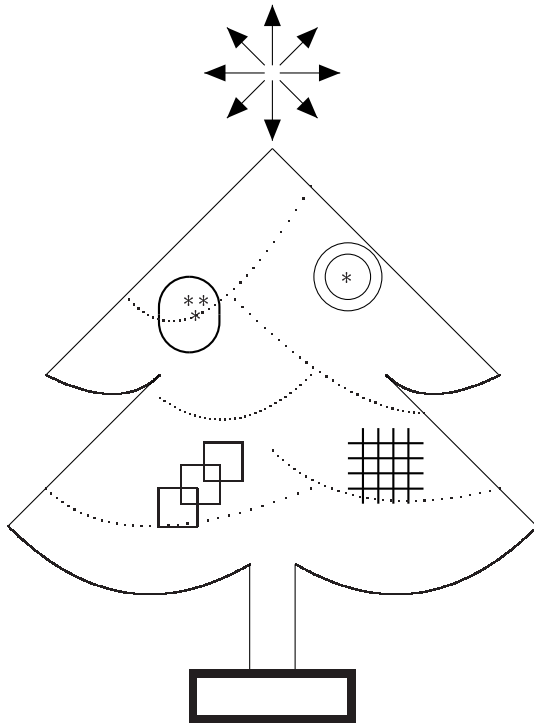
Poslední rámeček již překrývá další text, protože jeho referenční bod (levý dolní roh) je mimo plochu vymezenou pro obrázek.

Příklad 11.4 ukazuje složitější použití picture, takový obrázek již dá trochu práce.

Několik postřehů k balíku picture

- ★ Načrtněte si obrázek na čtverečkový papír, usnadní vám to vynášení souřadnic.
- ★ Pro kreslení složitějších křivek doporučujeme použít některý z nadstandardních balíčků, kreslení pomocí `qbezier` nedává zvláště v případě plné čáry příliš efektivní výsledek.
- ★ Rozmyslete si, zda není pro váš obrázek (obrázky) časově efektivnější seznámit se s některým nadstandardním balíkem, než ho pracně tvořit v prostředí picture a pak možná stejně zjistit, že nelze překonat určitá omezení.

■ Příklad 11.4.



```

\setlength{\unitlength}{1mm}
\begin{picture}(40,110)(-40,0)
%stromeček
\put(0,75){\line(1,-1){30}}
\put(0,75){\line(-1,-1){30}}
\qbezier(-30,45)(-20,40)(-15,45)
\qbezier(30,45)(20,40)(15,45)
\put(15,45){\line(1,-1){20}}
\put(-15,45){\line(-1,-1){20}}
\qbezier(-35,25)(-20,10)(-3,20)
\put(-3,20){\line(0,-1){15}}
\qbezier(35,25)(20,10)(3,20)
\put(3,20){\line(0,-1){15}}
%hvězda na špičce
\put(0,86){\vector(0,1){8}}
\put(0,84){\vector(0,-1){8}}
\put(1,85){\vector(1,0){8}}
\put(-1,85){\vector(-1,0){8}}
\put(1,86){\vector(1,1){5}}
\put(1,84){\vector(1,-1){5}}
\put(-1,86){\vector(-1,1){5}}
\put(-1,84){\vector(-1,-1){5}}
%ozdoby
\thicklines
\multiput(10,30)(0,2){4}{\line(1,0){10}}
\multiput(12,28)(2,0){4}{\line(0,1){10}}
\put(-11,53){\oval(8,10)}
\thinlines
\multiput(-15,25)(3,3){3}
{\framebox(5,5){}}
\put(10,58){\circle{6}}
\put(10,58){\circle{9}}
\put(9,57){\ast}
\put(-12,54){\ast}
\put(-10,54){\ast}
\put(-11,52){\ast}
%řetězy
\qbezier[30](-30,30)(-20,20)(5,30)
\qbezier[30](0,35)(10,25)(30,30)
\qbezier[30](-15,42)(-5,35)(5,45)
\qbezier[30](-5,55)(10,40)(20,40)
\qbezier[30](-19,55)(-10,45)(5,70)
%bedna
\linethickness{1mm}
\put(-10,0){\framebox(20,5){}}
\end{picture}

```


■ Příklad 11.5.

Zdrojový kód k obrázku 3.2 na straně 73.

```
\setlength{\unitlength}{5mm}
\begin{picture}(22,16)(-8,-6)
\put(0,-3){\line(1,0){5}}
\put(0,-3){\line(0,1){8}}
\put(0,5){\line(1,0){5}}
\put(5,-3){\line(0,1){8}}
\multiput(0,0)(0.5,0){10}{\line(1,0){0.3}}
%\put(0,0){\scalebox{9}{p}}
\put(1.5,6){šířka}
\put(4,6){\vector(1,0){1}}
\put(1,6){\vector(-1,0){1}}
\put(-2.8,2){výška}
\put(-1,1.5){\vector(0,-1){1.5}}
\put(-1,3){\vector(0,1){2}}
\put(-3.5,-2){hloubka}
\put(-1,-2.3){\vector(0,-1){0.7}}
\put(-1,-1){\vector(0,1){1}}
\put(-6,1){totální}
\put(-6,-0.3){výška}
\put(-4,-0.5){\vector(0,-1){2.5}}
\put(-4,2.5){\vector(0,1){2.5}}
\put(0,0){\circle*{0.4}}
\put(0.5,2.3){referenční bod}
\put(1,2){\vector(-1,-2){0.8}}
\put(5.4,-2.8){základní čára}
\put(5.2,-2){\vector(-1,2){0.8}}
\end{picture}
```

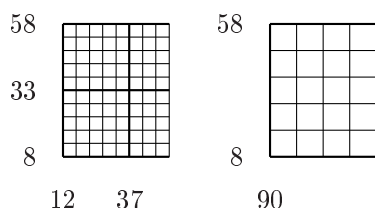
11.2. Mřížky — balík `graphpap`

Předpoklady: balík `graphpap`.

`\graphpaper[mezera](x_p, y_p)(šířka, výška)`

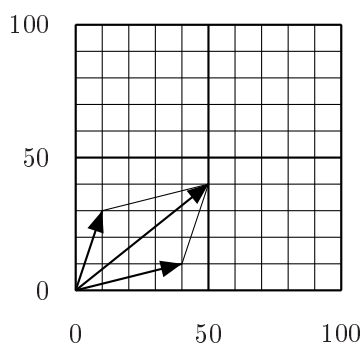
vykreslí mřížku začínající v bodě (x_p, y_p) o dané šířce a výšce. Rozteč mřížky je implicitně 10 jednotek nebo ji lze zadat pomocí nepovinného parametru *mezera*.

■ Příklad 11.6.



```
\graphpaper[5](12,8)(40,50)
\graphpaper(90,8)(40,50)
```

■ Příklad 11.7.



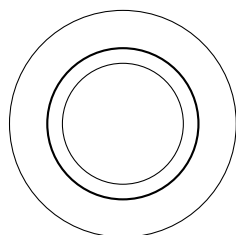
```
\begin{picture}(110,120)(-30,-10)
\graphpaper(0,0)(100,100)
\put(10,30){\line(4,1){40}}
\put(40,10){\line(1,3){10}}
\thicklines
\put(0,0){\vector(4,1){40}}
\put(0,0){\vector(5,4){50}}
\put(0,0){\vector(1,3){10}}
\end{picture}
```

11.3. Kružnice velkých průměrů — balík `pspicture`

Předpoklady: balík `pspicture`.

Při použití `pspicture` a příslušného ovladače je velikost průměru kružnic u příkazu `\circle` (popř. vyplněných kružnic u `\circle`) neomezená:²

■ Příklad 11.8.

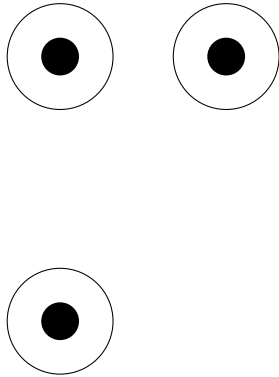


```
\setlength{\unitlength}{1mm}
\begin{picture}(60,40)(0,0)
\put(30,20){\circle{16}}
\thicklines
\put(30,20){\circle{20}}
\thinlines
\put(30,20){\circle{30}}
\end{picture}
```

²V [2, str. 92] je uvedeno, že použití neomezeného průměru kružnic umožňuje balík `pict2e`, ale tento balík (pokud je mi známo) nebyl vytvořen.

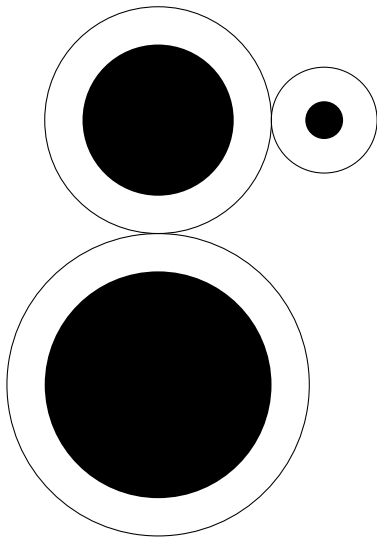
■ Příklad 11.9.

Pokud nepoužijeme `pspicture` a zadáme průměry větší než 14 jednotek, neobdržíme chybové hlášení, ale dostaneme špatný výsledek:



```
\setlength{\unitlength}{1mm}
\begin{picture}(50,70)
\put(25,20){\circle{40}}
\put(25,20){\circle*{30}}
\put(25,55){\circle{30}}
\put(25,55){\circle*{20}}
\put(47,55){\circle{14}}
\put(47,55){\circle*{5}}
\end{picture}
```

Nyní byl stejný zdrojový kód zpracován s použitím `pspicture`. Porovnejte s předchozím výsledkem.



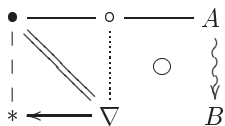
```
\setlength{\unitlength}{1mm}
\begin{pspicture}(50,70)
\put(25,20){\pscircle{40}}
\put(25,20){\pscircle*{30}}
\put(25,55){\pscircle{30}}
\put(25,55){\pscircle*{20}}
\put(47,55){\pscircle{14}}
\put(47,55){\pscircle*{5}}
\end{pspicture}
```

11.4. Balík Xy-pic

Předpoklady: balík xypic s parametrem `all`, tj. v preambuli `\usepackage[all]{xypic}`

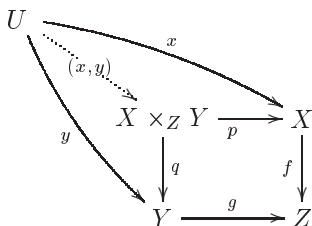
Balík Xy-pic nabízí velké možnosti pro sazbu grafů a diagramů.³ Velice snadno lze vysázet maticová diagramy a schémata užívaná např. v algebře nebo teorii kategorií.

■ Příklad 11.10.



```
\xymatrix{
{\bullet}\ar@{--}[d]\ar@{=}[dr]\ar@{-}[r]
& {\circ}\ar@{.}[d]\ar@{.}[dr]|{\bigcirc}
& A\ar@{-}[l]\ar@{>}[d]\\
{\ast}& {\nabla}\ar[l]
& B\ar@{.}[l] }
```

■ Příklad 11.11.



```
\xymatrix{
U\ar@/_[ddr]_y\ar@/^[drr]^x
\ar@{>}[dr]|-{(x,y)}\\
& X\times_Z Y\ar[d]^q\ar[r]_p
& X\ar[d]_f\\
& Y\ar[r]^g& Z }
```

■ Příklad 11.12.

$$A \times B \xrightarrow{f} C \cdots \cdots \cdots \overset{+}{D}$$

```
\xymatrix@1{A\times B\ar[r]^f & \\
C\ar@{.}[r]^{\scriptstyle +} & D}
```

Jak ukazuje příklad 11.10, sází se digram po řádcích a z leva do prava pomocí příkazu `\xymatrix{...}`. Pro jednořádkový diagram bychom použili příkazu `$_\xymatrix@1{...}$`, který musí být ale vždy v matematickém prostředí (viz příklad 11.12).

Symbody v rozích se uvádějí do složených závorek (jeden znak může být bez závorek), za symbolem vždy následuje definice všech „šipek“, které z něj vycházejí. Základní příkaz pro sazbu „šipky“ má syntaxi `\ar@{styl}[sm]`. Do složených závorek se uvede *styl*, do hranatých *směr*. Pokud je část `@{styl}` vynechána, vysází se plná „šipka“ (viz příklad 11.13). Pokud chceme něco vložit *nad* resp. *pod* resp. přímo *na* „šipku“, užijeme symbol `^` resp. `_` resp. `|` jak jsme ostatně ze syntaxe podobných příkazů zvyklí (viz příklad 11.11 a další). Možnosti definování *stylu* ukazuje tabulka 3.1.

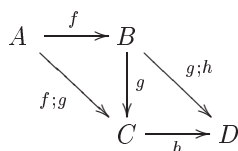
Obdobně jako u matic se k oddělování v řádku užívá `&` a k ukončení řádku `\\`. Diagramy se zakreslují po řádcích do pomyslné čtvercové sítě. Následující příklady ukazují některé možnosti, jak dosáhnout asymetrie v této síti:

³ Uživatelskou příručku k verzi 3.6 napsal Kristoffer H. Rose, Lyon, březen 1998. Interaktivní anglický manuál k verzi 3.2. (září 1995) je na <http://www.cl.cam.ac.uk/TeXdoc/xypic/xyguide-html/>.

@{=}	==	@{=>}	==>	@{-}	---
@{>}	→	@{.>}>	@{.}
@{-->}	-->	@{:>}>	@{<.}<
@{ }		@{~>}	~>	@{<->}	<-->

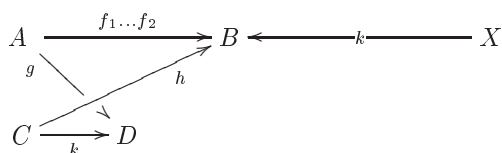
Tabulka 3.1: Některé *styly* „šipek“.

■ Příklad 11.13.



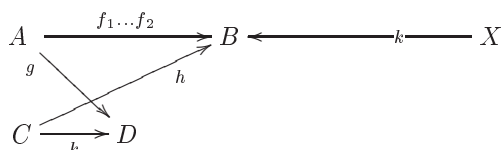
```
\xymatrix{
A~\ar[r]^f \ar[dr]_{f;g} & \\
B \ar[d]^g \ar[dr]^{g;h} & \\
& C \ar[r]_h & D }
```

■ Příklad 11.14.



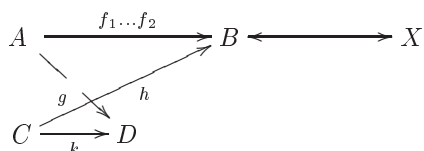
```
\xymatrix{
A~\ar[rr]^{f_1\dots f_2} \\
\ar[dr]_{(.2)\{g\}}|!{\{d\};[rr]}\hole \\
& B & X \ar@{->}[111]|{\{k\}} \\
C \ar[r]_{k} & \\
\ar[urr]_{>>>>>}\{h\} & D }
```

■ Příklad 11.15.



```
\xymatrix{
A~\ar[rr]^{f_1\dots f_2} \\
\ar[dr]_{(.2)\{g\}} \\
& B & X \ar@{->}[111]|!{\{11\};[11]}\{k\} \\
C \ar[r]_{k} & \\
\ar[urr]_{>>>>>}\{h\} & D }
```

■ Příklad 11.16.



```
\xymatrix{
A~\ar[rr]^{f_1\dots f_2} \\
\ar[dr]_{\{g\}}| \hole \\
& B & X \ar@{<->}[11] \\
C \ar[r]_{k} & \\
\ar[urr]_{>>>>>>>}\{h\} & D }
```

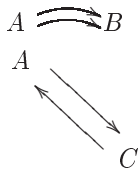
Prodloužení spojovací „šipky“ dosáhneme znásobením značky pro směr. Jak pohybovat s umístěním popisů vzhledem k „šipkám“ je patrné z př. 11.16. Příkaz `\hole`, zajistí přerušení „šipky“. Implicitně se „šipka“ přeruší uprostřed, obdobně se na střed bude vkládat popisek. Toto lze změnit příkazem `!\{t_1,t_2\}`, kde t_1 a t_2 jsou souřadnice vzhledem k počátečnímu a koncovému bodu „šipky“ vyjádřené násobností značek pro směr (obdobně jako u vyjadřování délky „šipek“).

Definici prohnuté a dvojité „šipky“ ukazuje příklad 11.17.

+	zvětšení	@{<->}	$A \longleftrightarrow B$
-	zmenšení	@_{{<->}}	$A \longleftrightarrow B$
+< velikost >	zvětšení o velikost	@^{{<->}}	$A \longleftrightarrow B$
-< velikost >	zmenšení o velikost	@2{{<->}}	$A \longleftrightarrow B$
[o]	kruhový tvar	@3{{<->}}	$A \longleftrightarrow B$
[F]	jednoduchý rámeček	@{{({}>}}	$A \longrightarrow B$
[F=]	dvojitý rámeček	@{{ >}}	$A \longrightarrow B$
[F.]	tečkovaný rámeček	@{{^{ }>}}	$A \longrightarrow B$
[F--]	čárkovaný rámeček	@3{{ >}}	$A \vDash B$
[F-,]	rámeček se stínem	@{{*}>}}	$A \bullet \longrightarrow B$

Tabulka 3.2: Některé *modifikace* a *varianty* „šipek“.

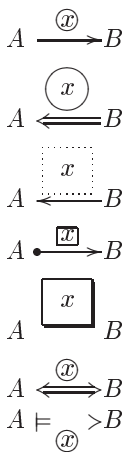
■ Příklad 11.17.



```
\xymatrix@1{
A~\ar@/^/[r] \ar@/^/@<-1ex>[r] & B \\
\yymatrix{ A~\backslash & C \ar@<1ex>[ul] \\
\ar@<1ex>[ul]; [] }
```

Popisy schémat mohou být modifikovány mnoha způsoby pomocí nepovinného parametru **modifikace{text}*. Také styly „šipek“ uvedené v tabulce 3.1 mohou mít další varianty (viz tabulka 3.2). Syntaxe je zřejmá z příkladu 11.18.

■ Příklad 11.18.



```
\xymatrix@1{
A~\ar@{->}[r]~*+<1ex>[o][F]{x} & B \\
\yymatrix@1{
A~\ar@2{<->}[r]~*+<3ex>[o][F]{x} & B \\
\yymatrix@1{
A~\ar@{<->}[r]~*+<3ex>[0][F.]{x} & B \\
\yymatrix@1{
A~\ar@{{*}>}[r]~*+[F]{x} & B \\
\yymatrix@1{
A~\ar@{}[r]~*+<3ex>[F-,]{x} & B \\
\yymatrix@1{
A~\ar@2{<->}[r]~*+<1ex>[o][F]{x} & B \\
\yymatrix@1{
A~\ar@2{{|>}}[r]~*+<1ex>[o][F]{x} & B }
```

Xy-pic umožňuje tvořit také prostorové matice (viz příklad 11.19, 11.20) a vytvořené objekty otáčet (viz příklad 11.21). Podrobný popis přesahuje rámec tohoto textu, zdroje dalších informací viz pozn. pod čarou 3.

11.5. Balík amscd

Předpoklady: balíky amsmath, amscd.

Jednoduché komutativní diagramy lze vytvářet s použitím amscd. Tento styl však zdaleka nedosahuje možností balíku $\text{\texttt{Xy-pic}}$ (viz odstavec 11.4), neboť umožňuje kreslit jen obdelnikové diagramy.⁴ Výhodou je jednoduchá syntaxe digramů pomocí prostředí CD. Toto prostředí je nutno používat v matematickém režimu, např. uzavřít do $\backslash[$, $\backslash]$.

K zadání šipek se užívají příkazy $\text{\texttt{@>>>}}$ pro šipku vpravo, $\text{\texttt{@<<<}}$ pro šipku vlevo, $\text{\texttt{@AAA}}$ pro šipku nahoru, $\text{\texttt{@VVV}}$ pro šipku dolů. Popisky pod šipku nebo vedle šipky (resp. nad nebo vpravo) lze vložit zapsáním mezi první a druhý (resp. druhý a třetí) znak určující šipku. Syntaxe je zřejmá z příkladů 11.22, 11.23.

■ Příklad 11.22.

$$\begin{array}{ccc}
 P & \xrightarrow{id \times \sigma} & P \times S \\
 p \downarrow & & \downarrow q \\
 M & \xrightarrow{\sigma} & P \times_H S
 \end{array} \quad (3.1)$$

```

\begin{equation}
\begin{CD}
P @>{id \times \sigma}>> P \times S \\
@VpVV @VVqV \\
M @>{\sigma}>> P \times_H S
\end{CD}
\end{equation}

```

■ Příklad 11.23.

$$\begin{array}{ccc}
 S^{\mathcal{W}_\lambda} \otimes T & \xrightarrow{j} & T \\
 \downarrow & & \downarrow P \\
 (S \otimes T)/I & \xlongequal{\quad} & (Z \otimes T)/J
 \end{array}$$

```

\begin{CD}
S^{\mathcal{W}_\lambda} \otimes T @>j>> T \\
@VVV @VV{P}V \\
(S \otimes T)/I @= (Z \otimes T)/J
\end{CD}

```

⁴Ke kreslení jednoduchých diagramů byl používán (zejména na sekci matematiky PřF MU v Brně) také soubor maker diag. Autoři ho však nedoporučují, neboť není rozšířený a zejména při zasílání článků do časopisů apod. činí potíže.

11.6. Barvy — balík `color`

Předpoklady: balík `color`.

Barevný text vytvoříte příkazem

`\textcolor{barva}{text}`, případně `\color{barva}{text}`.

■ Příklad 11.24.

Tohle jsou **moje** rukavice.

Tohle jsou **moje** rukavice.

Tohle jsou `\textcolor{red}{moje}` rukavice.

`\color{red}` Tohle jsou `\color{blue}{moje}` rukavice.

Příkazem `\colorbox{barva}{text}` vysázíte text na pozadí barevného obdélníka, příkazem `\fcolorbox{barva1}{barva2}{text}` vytvoříte kolem textu rámeček v barvě 1, který bude vyplněn barvou 2.

■ Příklad 11.25.

Tohle jsou `\colorbox{green}{moje}` rukavice.

Tohle jsou **moje** rukavice.

Tohle jsou `\fcolorbox{green}{yellow}{moje}` rukavice.

Tohle jsou **moje** rukavice.

Obarvit lze i např. matici `\fcolorbox{cyan}{yellow}{ $ \begin{vmatrix} 1 & 2 & 3 \\ 8 & 9 & 10 \\ 1 & 2 & 3 \end{vmatrix} $ }`.

Obarvit lze i např. matici $A = \begin{bmatrix} 1 & 2 & 3 \\ 8 & 9 & 10 \\ 1 & 2 & 3 \end{bmatrix}$.

Předdefinovány jsou barvy: `black`, `white`, `red`, `green`, `blue`, `yellow`, `cyan`, `magenta`.
Příkazem

`\definecolor{název barvy}{rgb}{k,l,m}` `\definecolor{názevbarvy}{rgb}{k,l,m}`

lze definovat vlastní barvu tak, že určíme jaké množství červené (*r*), zelené (*g*) a modré (*b*) se bude na výsledné barvě podílet. Parametry *k*, *l*, *m* jsou čísla od 0 do 1 odpovídající podílu uvedených tří barev.

Např. `magenta` je vytvořena jako `\definecolor{magenta}{rgb}{1,0,1}`. Tmavší odstín definujeme jako `\definecolor{magentatmava}{rgb}{.5,0,.5}`

V černobílém prostředí lze s výhodou použít různé odstíny šedé, které specifikujeme pouze jedním číslem od 0 (černá) do 1 (bílá) — viz příklad 11.26.

■ Příklad 11.26.

Něco může být tmavší a něco světlejší, barvy užívejte se vkusem a s mírou!!!

```
\definecolor{magentatmava}{rgb}{.5,0,.5}
```

```
\definecolor{mojebarva}{rgb}{.8,.5,.2}
```

```
\definecolor{tmava}{gray}{.5}
```

```
\definecolor{svetla}{gray}{.75}
```

Něco může být `\colorbox{tmava}{tmavší}` a něco `\colorbox{svetla}{světlejší}`,

barvy užívejte `\textcolor{mojebarva}{se vkusem}` a

```
\fcolorbox{magentatmava}{white}{s mírou}!!!
```

Příkazem `\pagecolor{barva}` vytvoříte barevné pozadí celé stránky. Pozadí změní barvu od aktuální stránky až do stránky, na která je další příkaz `\pagecolor{barva}`.

12. Obrázky vytvořené externě

V době, kdy byl napsán \TeX , neexistoval PostScript/EPS, JPEG, GIF, BMP ani jiné dnes běžné grafické formáty. Proto Knuthův⁵ DVI formát nepodporuje importovanou grafiku.

Vkládání grafiky probíhá až na úrovni zpracování DVI. Ten je nejčastěji konvertován do PostScriptu — například na *bartovi* pomocí *dvips*. PostScript (dále jen PS) je dnes jeden z nejpoužívanějších jazyků pro popis stránek a většina kvalitních tiskáren pracuje právě v tomto formátu, neboť umožňuje získat sazbu na profesionální úrovni. Nadále budeme předpokládat, že také váš text je připravován pro tisk z PS.⁶

Při konvertování DVI do PS je z importované grafiky nejlépe podporován EPS (Encapsulated PostScript), který je jakousi podmnožinou PostScriptového jazyka (podrobněji viz str. 70).

Ačkoliv lze dnes do \LaTeX u vkládat v podstatě libovolný grafický formát (EPS, JPEG, GIF, BMP apod.), my se zde zaměříme na způsob, který je dle zkušeností autorů pro začínající uživatele nejvhodnější. Budeme zpracovávat grafiku ve dvou krocích:

- **příprava obrázku v grafickém formátu EPS, popř. PS** (odstavec 12.1);
- **vložení takto připravovaného obrázku do \LaTeX u.**

Pro snadné vkládání grafiky vznikl k verzi $\text{\LaTeX}2_{\epsilon}$ „standardní“ balík *graphics* a jeho rozšířená verze – balík *graphicx*. Pro tento dokument byl použit *graphicx* (viz odstavec 12.2).

Tento způsob je jednoduchý a přitom dává rychle efektivní výsledky.

12.1. Příprava obrázku v grafickém formátu EPS (popř. PS)

Obrázek ve formátu EPS (popř. PS) lze získat buď tak, že ho sami vytvoříme nebo tak, že do EPS (PS) formátu konvertujeme obrázek ve formátu jiném (BMP, TIFF, JPEG apod.).

Vytváříme obrázky ve formátu EPS (popř. PS)

Obrázky lze vytvářet (modifikovat) ve formátu EPS (popř. PS) přímo užitím PostScriptových příkazů ve zdrojovém kódu, to je však pro většinu uživatelů obtížné. Snadnější je užití některého kreslicího programu (např. *xfig*).

xfig je volně dostupný (tzv. freeware) program pro UNIX/Xwindows.⁷ Lze v něm nakreslit obrázek, který bude importován ve formátu EPS. Bohužel v něm nelze libovolně modifikovat EPS obrázky vytvořené jiným kreslicím programem. (Spouštění na *bartovi* příkazem *xfig*). Manuál: <http://duke.usask.ca/~macphed/soft/fig/xfig-manual/>.

Podrobněji se budeme v celé kapitole věnovat matematickému software užívanému na KM PřF MU v Brně – Maple (pro UNIX i Windows). Export do formátu EPS umožňuje také AutoCAD a celá řada dalších aplikací. Některé programy vytvářejí bohužel nestandardní EPS

⁵Donald E. Knuth, tvůrce systému počítačové sazby \TeX , viz [2, str. 9].

⁶Podrobněji o vlastnostech PostScriptu a další zajímavé informace viz velmi srozumitelný článek P. Olšáka: *Jak \TeX pracuje s PostScriptem* [6].

⁷Objevila se i verze 3.2.3x pro MS Windows 9x/Me/NT/2000/XP, autoři s ní nemají zatím zkušenosti. Více informací na <http://www.cs.usask.ca/grads/wew036/latex/xfig.html>.

soubory, které je třeba před zpracováním jiným programem (např. před vložením do L^AT_EXu) nejdříve upravit. Příkladem jedné z takových rozšířených aplikací, která vytváří nestandardní výstup do EPS je Mathematica (starší verze 2.x).

Maple

Pomocí Maple lze snadno znázornit grafy funkcí, křivky, plochy ap. a po té exportovat do EPS. Velmi dobře lze také vyřešit záměnu popisů obrázků v Maple za popisky „L^AT_EXovské“ (viz kapitola 14).

Konvertujeme jiný grafický formát do EPS (popř. PS)

Existuje celá řada programů, které konvertují jiné grafické formáty do EPS, popř. PS. Jedním z nich, který lze vřele doporučit je **xv** (Interactive Image Display for the X Window System), který lze použít také na *bartovi* (spouští se příkazem **xv**).

xv lze použít nejen pro konverzi obrázků z jednoho grafického formátu do jiného, ale i k úpravě velikosti, barev apod.

Více informací včetně manuálu: <http://www.trilon.com/xv/downloads.html>

Jiný velmi dobrý grafický konvertor pro Unix a Linux i pro Windows NT, Macintosh je např.

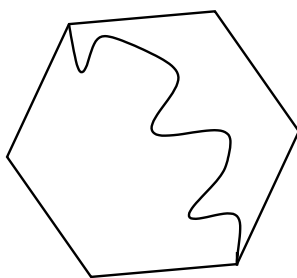
ImageMagick: <http://www.imagemagick.org/>.

Rozdíl mezi EPS a PS

EPS (Encapsulated PostScript) je jakousi podmnožinou PostScriptového jazyka, což znamená, že používá jen některé PostScriptové příkazy (např. nepoužívá příkazy pro velikosti stránek jako **letter**, **a4** apod). Druhý rozdíl je v tom, že EPS soubor obsahuje vždy tzv. BoundingBox, který udává rozměry obrázku v jednotkách **bp**.⁸ BoundingBox je uveden v záhlaví EPS souboru (viz příklad 12.1).

■ Příklad 12.1.

Následující obrázek byl vytvořen v kreslicím programu **xfig** (viz str. 69) a exportován jako EPS soubor. Vedle je ukázka několika prvních řádků zdrojového kódu⁹ souboru **sest.eps**, kde sedmý řádek obsahuje údaje o BoundingBox.



```
%!PS-Adobe-2.0 EPSF-2.0
%%Title: sest.eps
%%Creator: fig2dev Version 3.2 Patchlevel 1
%%CreationDate: Sat Aug 31 20:17:51 2002
%%For: mlc@celenka (Lenka Cechová)
%%Orientation: Portrait
%%BoundingBox: 0 0 114 103
%%Pages: 0
%%BeginSetup
%%EndSetup
%%Magnification: 1.0000
%%EndComments
```

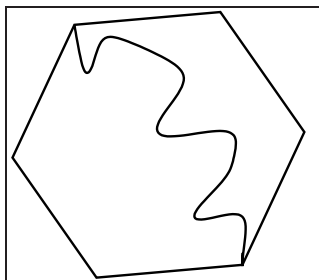
⁸**bp** — tzv. „big point“, $1\text{bp} = \frac{1}{72}\text{in}$, in — inch (palec) je 25,4 mm. Více o typografických jednotkách [2, str. 22–23].

⁹Zdrojový kód lze otevřít (a popř. modifikovat) v textovém editoru.

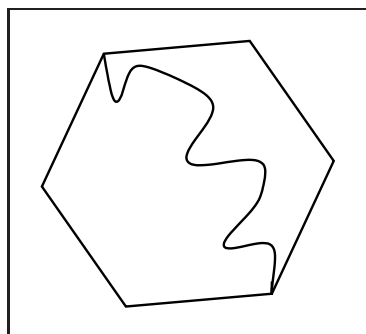
Čtyři čísla specifikující BoundingBox udávají souřadnice levého dolního rohu a pravého horního rohu. V našem případě má levý dolní roh souřadnice (0, 0) a pravý horní roh (114, 103), tj. obrázek je 114 bp široký a 103 bp vysoký.

Pokud vytvoříme rámeček pomocí `\frame`, bude mít právě rozměry BoundingBoxu, tj. změnou BoundingBox (přímo editací EPS souboru) lze změnit např. velikost tohoto rámečku:

`%BoundingBox: 0 0 114 103`



`%BoundingBox: -10 -10 124 113`



Je zřejmé, že BoundingBox je důležitý údaj pro začlenění grafického objektu do textu (L^AT_EX podle rozměrů BoundingBox vynechává místo v textu, transformuje obrázky apod.). Pokud u některých nestandardní EPS souborů chybí, je třeba ho doplnit. To lze udělat např. přímo vepsáním do EPS souboru. Jinou možnost, jak doplnit rozměry BoundingBox dává `psfixbb` (lze použít také na *bartovi*). Tento způsob je daleko snadnější, neboť `psfixbb` vypočítá BoundingBox tak, aby byl těsně kolem obrázku a tento BoundingBox vloží do EPS na konec souboru (viz příklad 12.2).

■ Příklad 12.2.

Ukázka přidání BoundingBox u obrázku vytvořeného v Maple pomocí `psfixbb` (obrázek na straně 88).

Část zdrojového kódu obrázku `sincos01.eps` po vygenerování v Maple:

```
%!PS-Adobe-3.0 EPSF
%%Title: Maple plot
%%Creator: MapleV
%%Pages: 1
%%BoundingBox: (atend)
%%DocumentNeededResources: font Helvetica
%%EndComments
```

Použití `psfixbb`:

```
$ psfixbb sincos01.eps
psfixbb: Processing sincos01.eps
DOPLNIT
```

Část zdrojového kódu obrázku `sincos01.eps` po vygenerování v Maple s vloženým BoundingBox:

```
DOPLNIT
```

Další ukázky práce s BoundingBoxem viz nepovinné parametry příkazu `\includegraphics` str. 75.

Řada PS souborů také obsahuje BoundingBox a lze je bez problému vkládat do \LaTeX u. Pokud tomu tak není, můžeme konvertovat PS soubor do EPS pomocí `ps2epsi` (toto lze použít také na *bartovi*, syntaxe: `ps2epsi soubor.ps`).

`ps2epsi` načte PS soubor, vypočítá rozměry BoundingBox a vytvoří EPS soubor s PostScriptovou grafikou. Výsledný soubor bude mít koncovku `.epsi`.

12.2. Vkládání obrázku ve formátu EPS (popř. PS) do \LaTeX u

Jak pracuje \LaTeX

Než přistoupíme k popisu vlastního vkládání obrázků do \LaTeX u, připomeňme stručně, jak \LaTeX pracuje. *Box* je v \LaTeX u libovolný objekt (písmeno, obrázek ap.), se kterým se zachází jako s jednotkou. Každý box má *referenční bod* na levé straně a *základní čáru*, která z tohoto referenčního bodu vychází (viz obr. 3.2). Pomocí referenčního bodu a základní čáry jsou sázena jednotlivá písmena na řádek (na tzv. *účaři* - viz [2, str. 25]). Stejným procesem je sázena také grafika. \LaTeX při vytváření DVI pouze „vypočítá“ kolik je třeba vynechat pro grafický objekt místa v textu, samotný objekt (např. obrázek) je dosazen až při překladu z DVI do PS. U starších DVI-prohlížečů se to projeví tím, že při prohlížení DVI souboru vidíte jen rámečky pro obrázky a nikoliv obrázky samotné (`xdvi` u novějších verzí operačního systému Linux – např. na *bartovi* – obrázky zobrazuje). Potom je třeba vytvořit příslušným programem (na *bartovi* je to `dvips`) postscriptový soubor a prohlédneme si jej například pomocí Ghostview. Tento postscriptový soubor (soubor s příponou `ps`) je již připraven k tisku.

Rozměry každého boxu jsou dány třemi údaji: *výškou*, *hloubkou* a *šířkou* (viz obr. 3.2). *Výška* je vzdálenost od referenčního bodu k hornímu okraji boxu, *hloubka* je vzdálenost od referenčního bodu k dolnímu okraji boxu a *šířka* je šířka boxu. *Totální výška* je součet výšky a hloubky (je zřejmé, že zadáním výšky a totální výšky je jednoznačně určena hloubka). Referenční bod u neotáčeného EPS-obrázku je jeho levý dolní roh, tj. hloubka je nulová a totální výška je ekvivalentní výšce. Při otáčení obrázku se tyto parametry mění (viz obr. 3.3).

Vlastní vkládání grafických objektů — balík `graphicx`

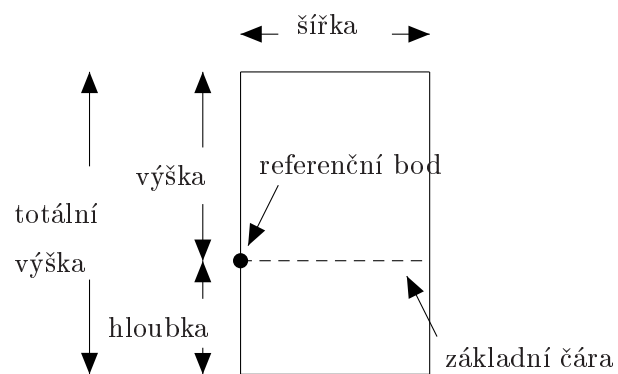
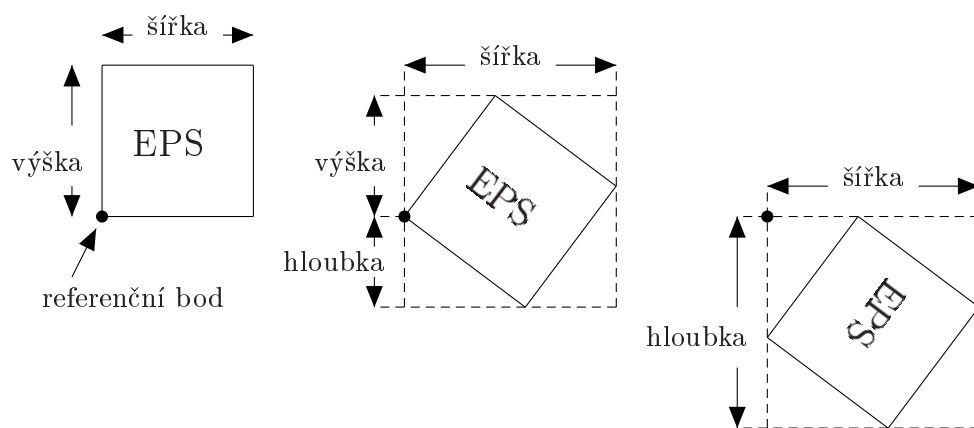
Pro $\text{\LaTeX}2_{\epsilon}$ byly vytvořeny dva grafické balíky `maker` — „standardní“ balík `graphics` a „rozšířený“ balík `graphicx`.¹¹ Oba balíky obsahují základní příkaz pro vložení obrázku

`\includegraphics {jméno souboru}`

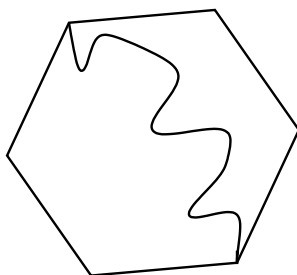
Tento příkaz je ale v každé verzi definován jinak. Příkaz `\includegraphics` ve verzi `graphicx` podporuje změny velikosti, rotaci ap. zadávané pomocí nepovinných parametrů, zatímco pro dosažení stejných změn pomocí příkazu `\includegraphics` ve verzi `graphics` je třeba použít dalších příkazů jako `\scalebox`, `\rotatebox` apod. (viz příklad 12.3). Práce s balíkem `graphicx` je pohodlnější, proto se v dalším textu budeme věnovat výhradně syntaxi příkazů v tomto balíku.

¹⁰Zdrojový kód k tomuto obrázku lze najít na straně 59 příklad 11.5.

¹¹Pro starší verzi \LaTeX u (pro $\text{\LaTeX}2.09$) byly vytvořeny balíky `maker epsf` a `psfig`. Jako hybrid těchto dvou grafických balíků pak vznikl `epsfig`.

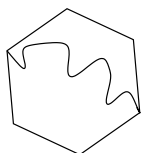
Obrázek 3.2: Jednoduchý *box* v \LaTeX u.¹⁰Obrázek 3.3: Otáčení *boxu* v \LaTeX u

■ Příklad 12.3.



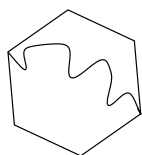
```
\includegraphics{sest.eps}
```

Původní obrázek `sest.eps` dvakrát zmenšíme a otočíme o třicet stupňů nejprve pomocí `\includegraphics` ve verzi `graphicx`:



```
\includegraphics[scale=0.5,angle=30]{sest.eps}
```

Pro srovnání ukážeme, jak lze téhož výsledku dosáhnout ve verzi `graphics`:



```
\scalebox{0.5}{
  \rotatebox{30}{
    \includegraphics{sest.eps}
  }}

```

Balík `graphicx`

Balík maker `graphicx` umožňuje vkládání obrázků vytvořených externě příkazem

```
\includegraphics[parametry]{jmeno souboru} \includegraphics[parametry]{jmeno souboru}
```

Nepovinné parametry slouží k další grafické úpravě vkládaných objektů (obrázků, textu apod.) Podrobněji popíšeme (vybrané) parametry: `height`, `totalheight`, `width`, `scale`, `angle`, `origin`, `bb`, `viewport`, `clip`, `noclip`, `draft`.

- **Rozměry:** `height=h` výška
`totalheight=t` totální výška
`width=s` šířka

Tyto parametry umožňují specifikovat rozměry vkládaného objektu zadáním jednoho popř. více údajů (viz obr. 3.2) v libovolných jednotkách, které \LaTeX akceptuje.¹²

Příkazem `\includegraphics[width=3cm]{soubor.eps}` dosáhneme toho, že šířka vkládaného obrázku bude 3cm (výška bude přepočítána v odpovídajícím poměru). Daleko vhodnější než zadávání fixních rozměrů je nastavení velikosti vkládaného objektu v závislosti na velikosti stránky, sazbě okolního textu ap., čímž se dokument stává nezávislejší.

¹²Více o typografických jednotkách [2, str. 22–23].

Např. příkazem `\includegraphics[width=0.5\texwidth]{soubor.eps}` vložíme obrázek o poloviční šířce, než je šířka sázeného textu. V případě, že změníme velikost stránek, tento poměr se zachovává a není třeba dokument znova upravovat.

Pokud používáme více parametrů, je třeba mít na paměti, že jsou „čteny“ zleva doprava a v tomto pořadí jsou prováděny také odpovídající transformace. Záměnou pořadí dostaneme různý výsledek (viz příklad 12.4).

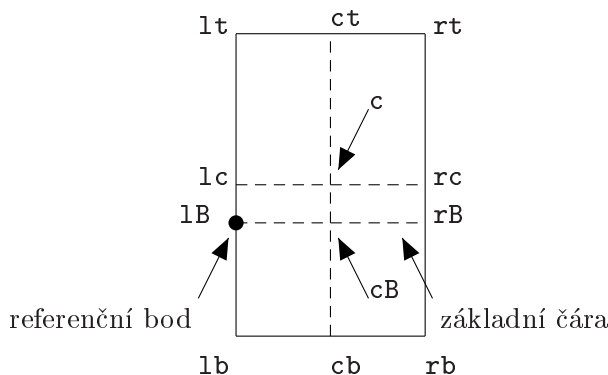
Současně s pořadím parametrů je třeba mít na zřeteli rozdíl mezi *výškou* a *totální výškou* vkládaného objektu (viz obrázky 3.2, 3.3 na straně 73). Tento rozdíl hraje důležitou roli zejména u objektů, které otáčíme nebo u nich měníme velikost - viz příklad 12.5).

- **Otáčení: `angle=n`**

Specifikuje velikost úhlu otáčení vkládaného objektu ve stupních. Kladné číslo zadává otáčení proti směru hodinových ručiček, záporné číslo ve směru hodinových ručiček. Primárně je objekt otáčen kolem referenčního bodu, pomocí parametru `origin` lze zvolit jiný bod.

- **Střed otáčení: `origin=p`**

Umožňuje zvolit jiný střed otáčení, než je primárně volený referenční bod (viz příklady 12.6, 12.7). Za `p` můžeme zvolit některou z dvanácti možností: `ct`, `rt`, `lc`, `c`, `rc`, `lB`, `cB`, `rB`, `lb`, `cb`, `rb`,¹³ kde např. `lt` značí top-left corner, tj. levý horní roh, apod. Polohu bodů vzhledem k BoundingBox vkládaného objektu ilustruje následující schéma:



- **BoundingBox: `bb = a b c d`**

Specifikuje rozměry BoundingBox zadáním čtyř čísel. Levý dolní roh BoundingBox má pak souřadnice (`a`, `b`), pravý horní roh (`c`, `d`). Jelikož příkaz `\includegraphics` načítá automaticky rozměry BoundingBox z EPS souboru (viz strana 88), nebývá parametr `bb` používán často. Pokud však rozměry BoundingBox v některém nestandardním EPS souboru chybí (popř. je chceme změnit jinak než přímou editací EPS – viz příklad 12.1), je tento parametr velmi užitečný.

¹³Na pořadí písmen ve dvojici nezáleží, tj. `lt` je totéž co `tl`.

- **Výřez obrázku:** `viewport = a b c d`

Specifikuje viditelnou část vkládaného objektu. Obdobně jako `BoundingBox` je zadána čtyřmi čísly, z nichž první dvě udávají souřadnice levého dolního rohu a další dvě souřadnice pravého horního rohu. Souřadnice jsou relativní vzhledem k levému dolnímu rohu `BoundingBoxu`. Pokud nechceme, aby byla zbývající část objektu zobrazena, je třeba použít parametr `clip`.

Primárně je nastaveno `noclip` a grafika se zobrazuje i vně výřezu (popř. `BoundingBoxu`). Při použití parametru `clip` je grafika vně oblasti určené k zobrazení „odstrážena“ a neobjeví se (viz příklad 12.8).

Použijeme-li `clip` jako nepovinný parametr balíku `graphicx` v záhlaví dokumentu, tj. `\usepackage[clip]{graphicx}`, bude mít vliv na všechny příkazy `\includegraphics` v dokumentu (u jednotlivých z nich lze tento vliv naopak potlačit použitím `noclip`).

- **Zobrazení `BoundingBox`:** `draft`

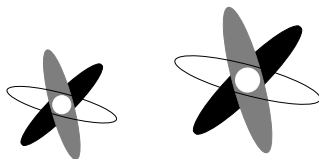
Použijeme-li nepovinný parametr `draf` u příkazu `\includegraphics`, zobrazí se na místo grafického objektu pouze `BoundingBox` a jméno vkládaného souboru.

Použijeme-li `draft` jako nepovinný parametr balíku `graphicx` v záhlaví dokumentu, tj. `\usepackage[draft]{graphicx}`, zobrazí se v celém dokumentu na místo grafických objektů pouze `BoundingBoxy` a jména vkládaných souborů. Toho lze s výhodou využít u dokumentu se složitější grafikou ke zrychlení prohlížení či tisku (viz příklad 12.5).

Pokud bychom chtěli rámeček kolem obrázku o velikosti `BoundingBoxu`, lze toho dosáhnout například příkazem `\frame` (v příkladě 12.1 byly rámečky vytvořeny následovně: `\frame{\includegraphics{soubor.eps}}`).

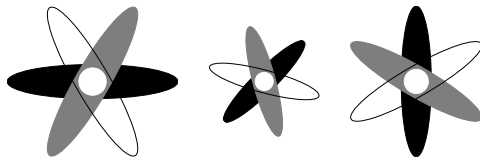
■ Příklad 12.4.

```
\includegraphics[angle=45,width=0.2\textwidth]{prik.eps}
\includegraphics[width=0.2\textwidth,angle=45]{prik.eps}
```



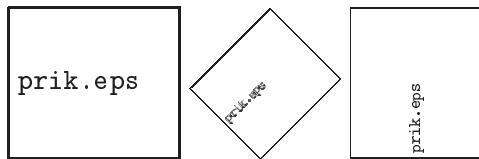
■ Příklad 12.5.

```
\includegraphics[totalheight=2cm]{prik.eps}
\includegraphics[angle=45,totalheight=2cm]{prik.eps}
\includegraphics[angle=90,totalheight=2cm]{prik.eps}
```



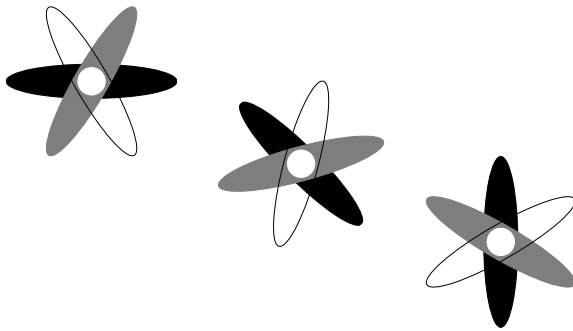
Na první pohled se může zdát divné, že druhý a třetí příkaz produkuje takto rozdílné výsledky. Situace je zřejmější, pokud necháme zobrazit `BoundingBox` vkládaných objektů. Vidíme, že

druhý i třetí objekt byl nejprve otočen o požadovaný úhel a potom upravena jeho výška na 2cm.

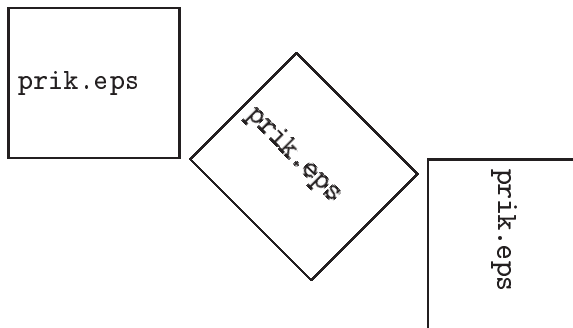


■ Příklad 12.6.

```
\includegraphics[totalheight=2cm]{prik.eps}
\includegraphics[totalheight=2cm,angle=45]{prik.eps}
\includegraphics[totalheight=2cm,angle=90]{prik.eps}
```



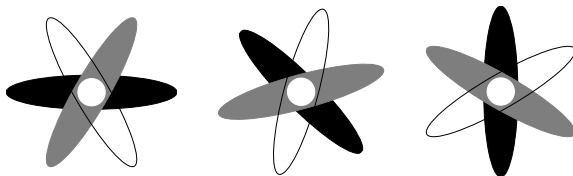
Pro názornost opět zobrazíme BoundingBox vkládaných objektů:



Situace v příkladu 12.6 je způsobena tím, že \LaTeX primárně otáčí grafické objekty kolem referenčního bodu, který není ve středu objektu (viz obr. 3.2 strana 73). Pomocí parametru `origin` můžeme zvolit otáčení kolem jiného bodu. Srovnajte příklad 12.6 s příkladem 12.7, kde je zvoleno otáčení kolem středu.

■ Příklad 12.7.

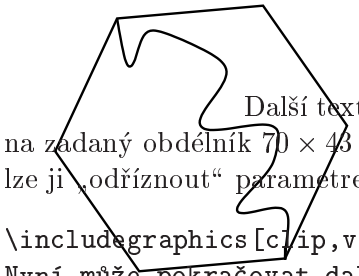
```
\includegraphics[totalheight=2cm]{prik.eps}
\includegraphics[totalheight=2cm,origin=c,angle=45]{prik.eps}
\includegraphics[totalheight=2cm,origin=c,angle=90]{prik.eps}
```



■ Příklad 12.8.

Použijme obrázek `sest.eps`, z příkladu 12.1, který má BoundingBox: 0 0 114 103. Pomocí parametru `viewport` z něj vyřízneme obdélník 70×43 bp v levém horním rohu, tj. levý dolní roh obdélníku má (relativně vzhledem k BoundingBox) souřadnice (0, 60) a pravý horní roh (70, 103).

```
\includegraphics[viewport = 0 60 70 103]{sest.eps}
```

Další text je sázen přes zbývající


Další text je sázen přes zbývající část obrázku. L^AT_EX vynechá místo pouze na zadaný obdélník 70×43 bp. Pokud nechceme, aby se zbývající část obrázku zobrazovala, lze ji „odříznout“ parametrem `clip`:

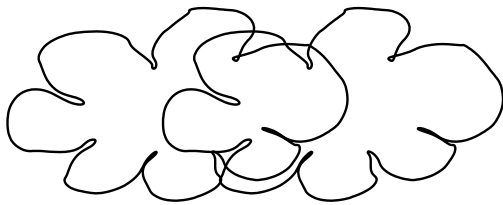
```
\includegraphics[clip,viewport = 0 60 70 103]{sest.eps}
```

Nyní může pokračovat další text

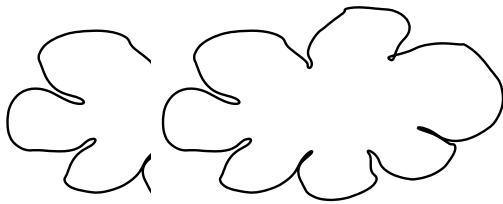


Nyní může pokračovat další text a nebude již překryt obrázkem.

■ Příklad 12.9.



```
\includegraphics
[viewport = 0 0 55 86]{mrak.eps}
\includegraphics{mrak.eps}
```



```
\includegraphics
[clip,viewport = 0 0 55 86]{mrak.eps}
\includegraphics{mrak.eps}
```

12.3. Další příkazy balíku **graphicx**

Kromě příkazu `\includegraphics` obsahuje balík **graphicx** další čtyři příkazy, které umožňují provádět některé transformace „libovolného“ L^AT_EXovského objektu.

Jelikož příkaz `\includegraphics` v balíku **graphicx** umožňuje použít nepovinné parametry `scale`, `angle`, `width` apod., nebývá užití následujících příkazů ve spojení s EPS souborem časté (viz příklad 12.3), ale s výhodou je použijeme při vkládání jiných objektů.

`\scalebox{h-násobek šířky}[v-násobek výšky]{objekt}`

Příkaz `\scalebox` umožňuje modifikovat objekt tak, že vynásobí *h*-krát původní šířku a *v*-krát původní výšku. Pokud nepovinný parametr pro násobek výšky chybí, zachová se původní poměr výšky k šířce. Jestliže je zadána záporná hodnota, vkládaný objekt se zrcadlí (zrcadlení lze dosáhnout i pomocí příkazu `\reflectbox{objekt}`) – viz příklad 12.11.

■ Příklad 12.10.

Krtek!

```
\scalebox{4}[0.8]{\large Krtek!}
```

Krtek!

```
\scalebox{4}{\large Krtek!}
```

■ Příklad 12.11.

Krtek!!kætɪX

```
\Large Krtek!%
\reflectbox{\Large Krtek!}
```

Krtek!
kætɪX

```
\Large Krtek!%
\scalebox{-1}{\Large Krtek!}
```

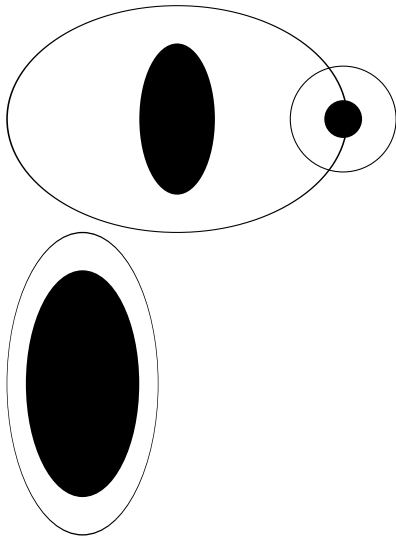
Krtek!
Kɪfɛɹɪ

```
\Large Krtek!%
\scalebox{1}[-1]{\Large Krtek!}
```

Krtek!
kɪfɛɹɪ

```
\Large Krtek!%
\scalebox{-1}[-1]{\Large Krtek!}
```

■ Příklad 12.12.



```
\setlength{\unitlength}{1mm}
\begin{picture}(50,70)
\scalebox{.5}[1]{\put(25,20){\circle{40}}
\put(25,20){\circle*{30}}}
\put(25,55){\scalebox{1.5}[1]{\circle{30}}}
\put(25,55){\scalebox{.5}[1]{\circle*{20}}}
\put(47,55){\circle{14}}
\put(47,55){\circle*{5}}
\end{picture}
```

V tomto příkladě jsme (pouze!) užitím `\scalebox` modifikovali příklad 11.9 ze strany 61. Všimněte si rozdílného působení příkazu `\scalebox` uvnitř a vně `\put`. V prvním případě ovlivňuje jen parametr `\circle`, zatímco ve druhém `\put` ovlivní i souřadnice umístění „oka“ (porovnejte vzhledem k nejmenšímu „oku“, které zůstalo nezměněno).

`\resizebox{šířka}{výška}{objekt}`

`\resizebox*{šířka}{totální výška}{objekt}`

Příkazy `\resizebox`, `\resizebox*` modifikují objekt podle požadovaných rozměrů (udáných v libovolných L^AT_EXem akceptovaných jednotkách). Pokud místo jednoho z rozměrů vložíme `!`, upraví se tento rozměr vzhledem k druhému tak, aby se zachoval původní poměr (viz příklady 12.13–12.14).

Rozdíl mezi `\resizebox` a `\resizebox*` ilustrují příklady 12.17, 12.22.

■ Příklad 12.13.

klubko
klubko

```
\resizebox{4cm}{3cm}{klubko}
\resizebox{4cm}{!}{klubko}
```

Místo fixních rozměrů lze použít nastavení velikosti vkládaného objektu v závislosti na velikosti stránky, sazbě okolního textu ap. (analogicky jako u `\includegraphics` – viz příklad 12.4). Také můžeme použít jako argument `\height`, `\totalheight`, `\width` nebo `\dept`, což umožňuje zachovat původní rozměr objektu (viz příklad 12.15).

■ Příklad 12.14.

▷◁

```
$_\triangleright\triangleleft$
```



```
\resizebox{3cm}{1cm}{$_\triangleright\triangleleft$}
```



```
\resizebox{3cm}{!}{$_\triangleright\triangleleft$}
```



```
\resizebox{!}{1cm}{$_\triangleright\triangleleft$}
```

■ Příklad 12.15.

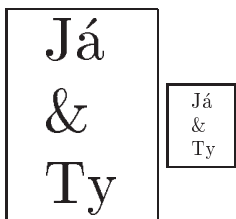


```
\resizebox{3cm}{\height}{$_\triangleright\triangleleft$}
```



```
\resizebox{\width}{1cm}{$_\triangleright\triangleleft$}
```

■ Příklad 12.16.



```
\framebox{
\resizebox{!}{15mm}{
\parbox{5mm}{Já \& Ty}
}}
\framebox{
\resizebox{!}{5mm}{
\parbox{5mm}{Já \& Ty}
}}
```

■ Příklad 12.17.



```
\underline{základní čára}
\resizebox{!}{1cm}{g}
\resizebox*{!}{1cm}{g}
```

`\rotatebox[střed otáčení]{úhel}{objekt}`

Příkaz `\rotatebox` otočí objekt o úhel zadaný ve stupních. Kladné číslo zadává otáčení proti směru hodinových ručiček, záporné číslo ve směru hodinových ručiček. Primárně je objekt otáčen kolem referenčního bodu, pomocí nepovinného parametru lze zvolit jiný bod:

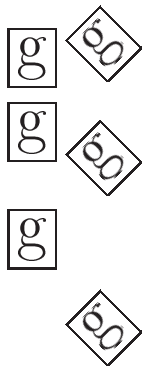
1. specifikací středu otáčení výběrem některé z dvanácti možností `ct`, `rt`, `lc`, `c`, `rc`, `lB`, `cB`, `rB`, `lB`, `cb`, `rb` (viz obr. na straně 75);
2. zadáním středu otáčení v souřadnicích `x=a`, `y=b`, (vzhledem k souřadnicím referenčního bodu — viz příklad 12.19).

■ Příklad 12.18.



```
\rotatebox{45}{\Huge zvíře}
\rotatebox{-45}{\Huge zvíře}
```

■ Příklad 12.19.



```
{\framebox{\Huge g}}
\rotatebox{45}
  {\framebox{\Huge g}}

{\framebox{\Huge g}}
\rotatebox[origin=rb]{45}
  {\framebox{\Huge g}}

{\framebox{\Huge g}}
\rotatebox[x=4cm,y=5cm]{45}
  {\framebox{\Huge g}}
```

■ Příklad 12.20.

Leden	Únor	Březen	Σ
12	35	17	64
24	27	13	64

```
\begin{tabular}{rrrr}
\rotatebox{45}{Leden}&
\rotatebox{45}{Únor}&
\rotatebox{45}{Březen}&
{\Large $\Sigma$}\\
\hline
12 & 35 & 17 & 64\\
24 & 27 & 13 & 64\\
\hline
\end{tabular}
```


Leden	Únor	Březen	Σ
12	35	17	64
24	27	13	64

```

\newcommand{\00}[1]{\rule{1em}{0pt}}
\makebox[0cm][c]
  {\rotatebox{45}{\ #1}}}
\begin{tabular}{rrrr}
\00{Leden}&\00{Únor}&&
\00{Březen}&{\Large $\Sigma$}
\\ \hline
12 & 35 & 17 & 64\\
24 & 27 & 13 & 64\\ \hline
\end{tabular}

```

■ Příklad 12.21.

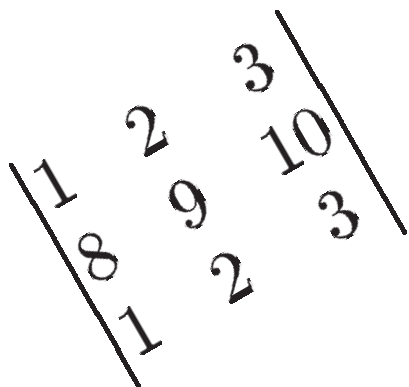


```

\fbbox{
  \resizebox{5cm}{20mm}{
    \rotatebox{45}{
      \parbox{2cm}{\raggedright
        Loupežník Loupežník
        Loupežník Loupežník}
    }}}

```

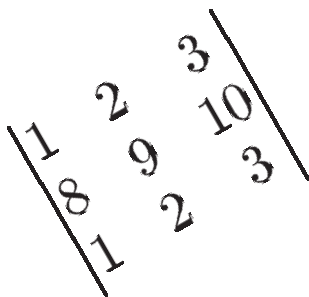
■ Příklad 12.22.



```

\resizebox{!}{40mm}{
  \rotatebox{30}
  {\$ \begin{vmatrix}
1 & 2 & 3 \\ 8 & 9 & 10 \\ 1 & 2 & 3
\end{vmatrix}$ }}

```



```

\resizebox*{!}{40mm}{
  \rotatebox{30}
  {\$ \begin{vmatrix}
1 & 2 & 3 \\ 8 & 9 & 10 \\ 1 & 2 & 3
\end{vmatrix}$ }}

```

Několik postřehů k balíku `graphicx`

- ★ Daleko vhodnější než zadávání fixních rozměrů je nastavení velikosti vkládaného objektu v závislosti na velikosti stránky, sazbě okolního textu ap., čímž se dokument stává nezávislejší (viz příklad 12.4).
- ★ Při otáčení objektů nezapomeňte, že primárně je střed otáčení referenční bod, který není totožný se středem objektu (viz příklady 12.6, 12.7, 12.19).
- ★ Pokud používáme více parametrů, je třeba mít na paměti, že jsou „čteny“ zleva doprava a v tomto pořadí jsou prováděny také odpovídající transformace. Záměnou pořadí dostaneme různý výsledek (viz příklad 12.4).
- ★ Jestliže vám dělá problém manipulovat s obrázky, čertejte si zpočátku `BoundingBox` kolem obrázku.

13. Začlenění obrázků do textu — prostředí `figure`

K velkým výhodám sazby v `LATEX`u patří to, že se nemusíme starat o formátování textu. `LATEX` rozděluje text „za nás“ co možná neoptimálněji do požadovaných oddílů (kapitol, sekcí, odstavců apod.) a sazí ho na stránky požadované velikosti podle typografických pravidel.

V některých případech ovšem požadujeme, aby určitá část (např. obrázek, tabulka) nebyla rozdělena na dvě stránky, popř. aby byla umístěna vždy na „vhodném“ místě (např. obrázek vždy nahoře na stránce) apod. K tomuto účelu slouží tzv. plovoucí prostředí. `LATEX` má dvě standardní plovoucí prostředí: `figure` (používá se obvykle pro obrázky) a `table` (používá se obvykle pro tabulky).

Tato prostředí jsou podrobně popsána v [2]. My se zde uvedeme pouze několik užitečných poznámek, které souvisejí s prostředím pro obrázky (`figure`) a které v [2] nenajdete.

13.1. Jak ovlivnit sazbu plovoucích obrázků

Systém `LATEX` obsahuje poměrně náročný algoritmus pro umísťování plovoucích objektů. Jedním z nich je, že zachovává jejich pořadí, tj. nikdy nedojde k přehození obrázků, popř. tabulek. Pokud některý z plovoucích objektů nemůže `LATEX` umístit (např. proto, že jsme nastavili nesplnitelné parametry) může to vést k situaci, že neumístěný objekt brání umístění všech následujících objektů a ty se pak soustředí na konec textu na samostatné stránky, popř. se tyto neumístěné objekty hromadí v paměti a dojde k překročení možností počítače. Překlad se ukončí s chybou `TeX capacity exceeded! ...You can ask wizard to enlarge me` (Kapacita `TEX`u překročena! ...Můžete se zeptat kouzelníka, zda mne nemůže zvětšit). Jednou z pomůcek usnadňujících činnost algoritmu umísťujícího plovoucí objekty je zvláštní specifikátor `!` popsáný v [2, str. 103].

Dále můžeme využít celou řadu parametrů, které ovlivňují chování algoritmu pro sazbu plovoucích objektů:

`topnumber` — čítač udávající maximální počet plovoucích objektů, které lze umístit do horní části stránky

`bottomnumber` — čítač udávající maximální počet plovoucích objektů, které lze umístit v dolní části stránky

`totalnumber` — čítač udávající maximální počet plovoucích objektů, které lze umístit na jednu stránku

`\topfraction` — číslo udávající část stránky, která může být maximálně věnována plovoucímu objektu umístěnému v horní části stránky

`\bottomfraction` — číslo udávající část stránky, která může být maximálně věnována plovoucímu objektu umístěnému v dolní části stránky

`\textfraction` — číslo udávající minimální část stránky, která musí být na stránce s plovoucími objekty věnována textu

`\floatpagefraction` — číslo udávající minimální část stránky, kterou musí na stránce s plovoucími objekty tyto objekty zabírat

`dbltopnumber` — čítač udávající maximální počet plovoucích objektů, které lze umístit do horní části stránky ve dvousloupcovém režimu

`\dbltopfraction` — číslo udávající maximální část stránky, která může být věnována plovoucím objektům v dvopustloupcovém režimu

`\dblfloatpagefraction` — číslo udávající minimální část stránky, kterou musí na stránce s plovoucími objekty tyto objekty zabírat v dvousloupcovém režimu

`\floatsep` — míra udávající vzdálenost mezi plovoucími objekty na samostatné stránce

`\textfloatsep` — míra udávající svislou vzdálenost mezi plovoucím objektem a textem

`\dblfloatsep` — vzdálenost mezi plovoucími objekty ve dvousloupcovém režimu

`\dbltextfloatsep` — vzdálenost mezi plovoucím objektem a textem ve dvousloupcovém režimu

`\intextsep` — vertikální mezera vložená nad a pod plovoucí objekt, který je umístěn v textu

Z uvedeného je zřejmé, že parametry jsou na sobě vzájemně závislé. Vhodným užitím těchto parametrů dostaneme požadovaný výsledek. Čítače lze přenastavit pomocí příkazu `\setcounter`, míry pomocí `\setlength`, ostatní parametry pomocí `\renewcommand`.

13.2. Balík float

Další plovoucí prostředí lze definovat pomocí balíku `maker float`. Podrobný popis balíku `float` přesahuje rámec tohoto textu. Poznamenejme pouze, že s užitím tohoto balíku se mimo jiné také rozšiřují možnosti, jak specifikovat požadované umístění plovoucího objektu pomocí nepovinných parametrů. V [2, str. 102] jsou uvedeny tyto standardní možnosti:

t – přednostně nahoře na stránce

b – přednostně dole na stránce

h – přednostně zde, kde je uveden zdrojový text

p – objekt se umístí na stránku, kde jsou jen plovoucí objekty a žádný text.

Kombinací těchto parametrů dosáhneme požadovaného umístění (viz příklad 14.2).

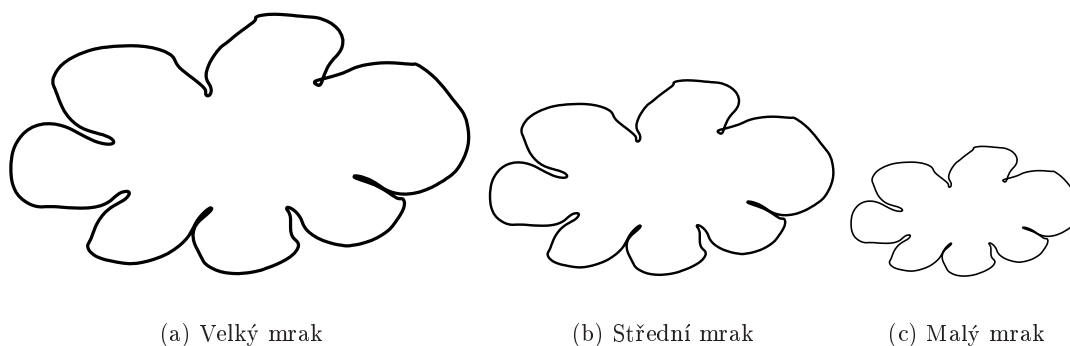
Parametr **h** znamená *zde, pokud je to možno* (nikoliv *vždy zde*). Použit striktní příkaz pro umístění obrázku v místě zdrojového textu nám dává právě balík `float`:

H – vždy zde, kde je uveden zdrojový text

(viz příklad 14.1).

13.3. Balík **subfigure**

Balík `subfigure` umožňuje vložit více obrázků (každý se svým vlastním popisem) do jednoho prostředí `figure` a následně pod tím umístit popisku obrázku jako celku (viz příklad 13.1).



Obrázek 3.4: Tři mraky

■ Příklad 13.1.

```
\begin{figure}
\centering
\mbox{\subfigure[Velký mrak]{\includegraphics[width=.4\textwidth]{mrak.eps}}
\subfigure[Střední mrak]{\includegraphics[width=.3\textwidth]{mrak.eps}}
\subfigure[Malý mrak]{\includegraphics[width=.2\textwidth]{mrak.eps}}
}
\caption{Tři mraky}
\end{figure}
```

Pokud se rozhodnete umístit dva obrázky (nebo i více) vedle sebe do jednoho prostředí `figure` se samostatnými číslovanými popisky, lze to udělat bez použití balíku `subfigure`, např. pomocí `minipage` (viz příklad 15.5).

14. Popisy obrázků – balík PSfrag

Předpoklady: balík PSfrag. Kromě umístění `psfrag.sty` do standardního adresáře pro L^AT_EXovská makra je třeba umístit `psfrag.pro` kamkoliv, kde má potřebné soubory váš PostScript driver. (Na *bartovi* je PSfrag instalován. Pokud používáte L^AT_EX pod windows, stačí si opatřit soubor `psfrag.ins` a přeložit ho L^AT_EXem. Tím se vygenerují soubory `psfrag.sty` a `psfrag.pro`, které pak pouze umístíte tak, jak bylo popsáno výše.)

Mnoho grafických systémů umožňuje exportovat výstup do EPS, ale málo z nich umožňuje do tohoto výstupu „rozumně“ vepisovat rovnice či jiné „vědecké texty“. PSfrag dává uživatelům možnost doplnit EPS soubor nejen o text, ale v podstatě o libovolný L^AT_EXovský výtvar.¹⁴ V grafickém souboru je třeba umístit tzv. *návěští* jako ukazatel pozice. Použitím L^AT_EXovského příkazu dojde k výměně návěští v obrázku a k jeho nahrazení rovnicí (popř. jiným objektem) v odpovídajících fontech, velikosti, popř. otočení apod. K této výměně dochází na úrovni překladu DVI souboru do PS souboru.

Syntaxe příkazu, který provede výměnu návěští za L^AT_EXovský objekt:

$$\backslash\text{psfrag}\{\textit{návěští}\}[\textit{posun}][\textit{psposun}][\textit{měřítko}][\textit{rotace}]\{\textit{objekt}\}$$

tj. např. jednoduchým příkazem `\psfrag{cos}{\$f(x)=\cos x + 1\$}` bude návěští `cos` nahrazeno odpovídající rovnicí. Příklad 14.1) ukazuje použití tohoto příkazu u obázku vytvořeného v Maple. Příkaz `\psfrag` musí být umístěn před použitím příkazu `\includegraphics`. Příkaz bude působit tak dlouho, dokud není okolní prostředí (např. `figure`) ukončeno.

■ Příklad 14.1.

Nejprve jsme v Maplu vytvořili obrázek pomocí následujících příkazů:

```
> with(plots):
> A:=textplot([3,-2,'sin'],align = LEFT):
> B:=textplot([-2.5,2.5,'cos'],align = LEFT):
> C:=plot([sin(x)-1,cos(x)+1], x=-2*Pi..2*Pi, xtickmarks=0, ytickmarks=0):
> display({A,B,C});
```

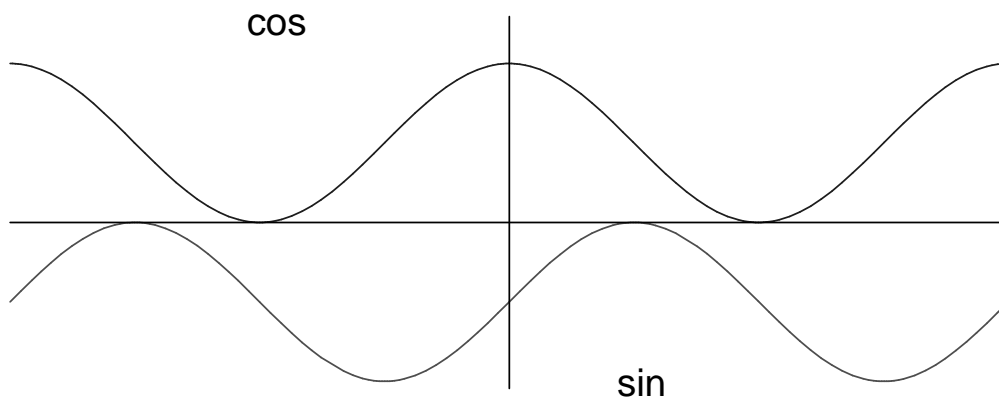
Potom vyexportujeme tento soubor s názvem `sincos.mws` do L^AT_EXu. Byl vytvořen soubor `sincos.tex` a kromě něj ještě soubor `sincos01.eps`, který obsahuje obrázek.¹⁵ Doplnili jsme BoundingBox pomocí `psfixbb` — viz strana 75).

Nyní tento obrázek můžeme vložit do L^AT_EXu:

```
\begin{figure}[H]
\includegraphics{sincos01.eps}
\caption{Obrázek vytvořený v~Maplu a exportovaný do EPS.}
\label{oo}
\end{figure}
```

¹⁴Uživatelskou příručku v angličtině k verzi 3.0 napsali Michael C. Grant a David Carlisle, viz např. <http://eewww.eng.ohio-state.edu/computing/latex2e/UserGuides.shtml>.

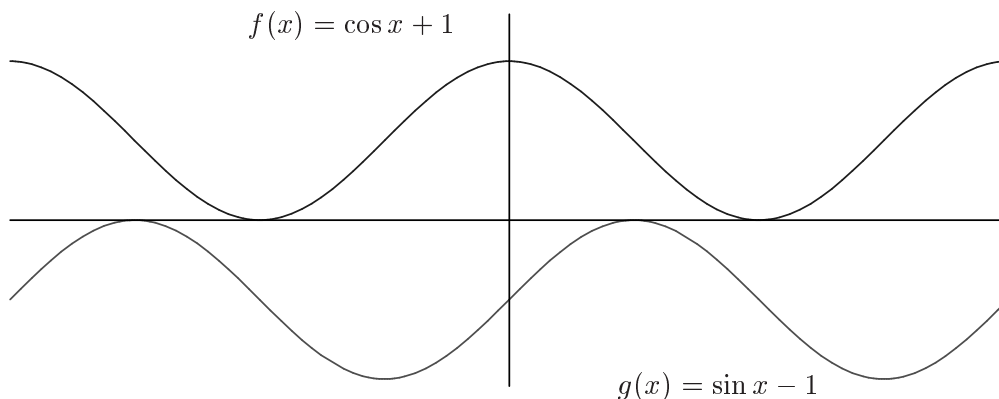
¹⁵Soubor `sincos.tex` obsahuje zápisník příkazů Maplu a lze ho přeložit do DVI s použitím balíku `maple2e.sty`.



Obrázek 3.5: Obrázek vytvořený v Maplu a exportovaný do EPS.

Vyměníme návěští za odpovídající rovnice:

```
\begin{figure}[hbt]
\psfrag{cos}{ $f(x)=\cos x + 1$ }
\psfrag{sin}{ $g(x)=\sin x - 1$ }
\includegraphics{sincos01.eps}
\caption{Obrázek \ref{oo} s~\LaTeX{}ovskými popiskami}
\end{figure}
```

Obrázek 3.6: Obrázek 3.5 s \LaTeX ovskými popiskami

Příkaz `\psfrag` umístí \LaTeX ovský *objekt* na stejné místo, kde je v EPS souboru *návěští*. Pokud se podíváte na výstup DVI prohlíčem, na levé straně daného obrázku bude umístěn seznam \LaTeX ovských objektů. Tento seznam vám umožňuje kontrolovat jejich vysazení. Ve finální PostScriptové verzi zmizí a \LaTeX ovské objekty budou umístěny na patřičné místo. DVI drivers nejsou bohužel schopny umístit \LaTeX ovské objekty do EPS (popř. PS) obrázku. Toto nahrazování se děje až na úrovni překladu DVI do PS, proto budete potřebovat výstup vytisknout nebo prohlédnout v PostScriptu. Vložení \LaTeX ovského objektu lze ovlivnit

nepovinnými parametry:

- **Posunutí**

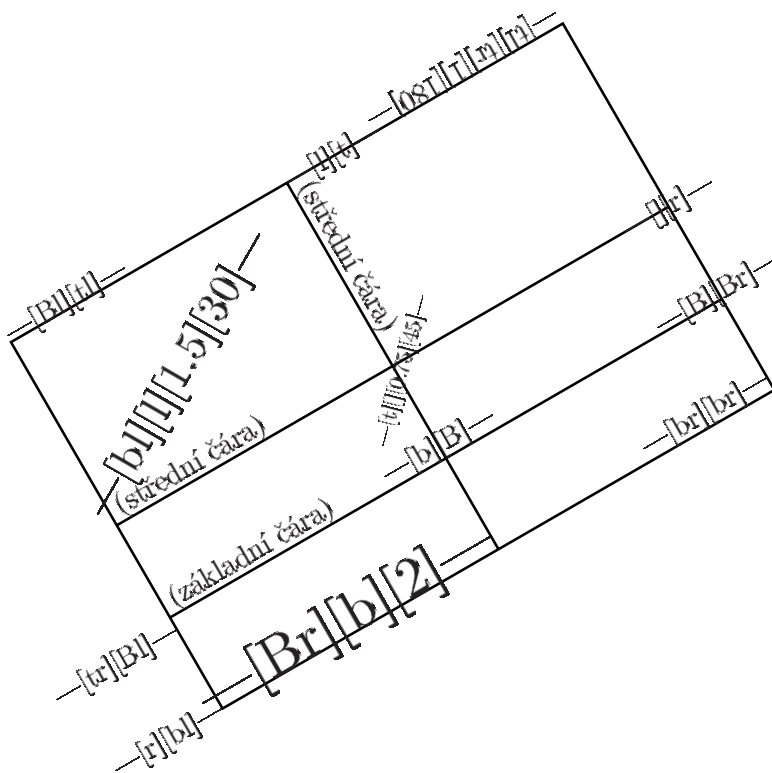
Parametry *posun* a *psposun* specifikují, jak jsou BoundingBox L^AT_EXovského objektu a BoundingBox návěstí v PostScriptovém souboru proti sobě umístěny. Parametr *posun* značí referenční bod L^AT_EXovského objektu, jeho syntaxe je analogická jako u parametru *origin* — viz strana 75. Primárně je nastaveno [B1].¹⁶ Parametr *psposun* specifikuje referenční bod PostScriptového objektu, nastavení je stejné jako u *posun*. Možnosti nastavení různých parametrů ukazuje obrázek 3.7.

- **Změna velikosti**

Parametr *měřítko* dává možnost měnit velikost L^AT_EXovského objektu. U textu doporučujeme raději měnit velikost fontu.

- **Otáčení**

Parametr *rotace* umožňuje otočení L^AT_EXovského objektu kolem referenčního bodu. Udává se ve stupních. Pokud otáčíme celý obrázek, pak otočení L^AT_EXovského objektu je rovno součtu otočení obrázku a parametru *rotace* (viz obrázek 3.7).



Obrázek 3.7: Ilustrace různých parametrů příkazu `\psfrag`.

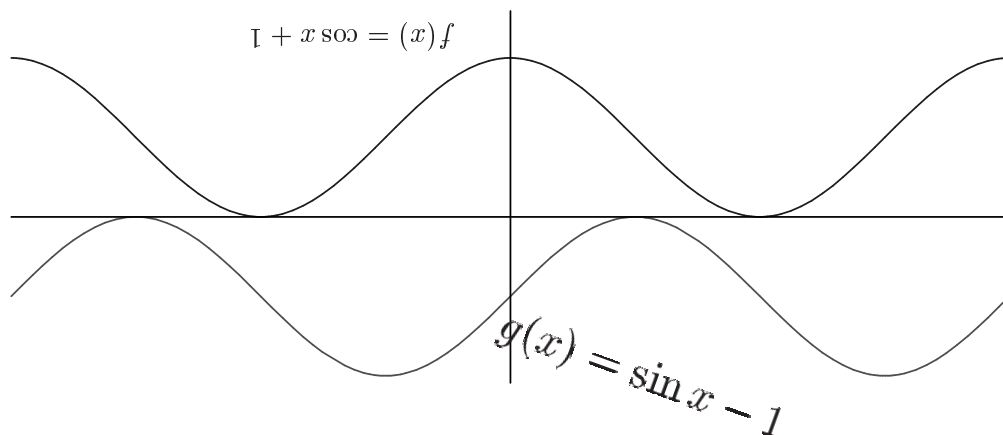
■ Příklad 14.2.

Návěstí z příkladu 14.1 modifikujeme pomocí nepovinných parametrů:

¹⁶Tato hodnota bude dosazena, pokud je argument úplně vynechán. Jestliže zapíšeme [], znamená to totéž jako [cc].


```
\psfrag{cos}{\scalebox{-1}{\mathrel{\cos x + 1}}}
\psfrag{sin}[cc][l][1.5][-20]{\mathrel{\sin x - 1}}

\includegraphics{sincos01.eps}
```



Jestliže pro dané *návěští* už existuje *objekt*, kterým má být toto návěští nahrazeno, příkaz `\psfrag` je bez varování přepíše. Varianta s hvězdičkou `\psfrag*` přidá tento nový *objekt* do seznamu, čímž k přepsání nedojde. Při použití hvězdičkové varianty může být jediný exemplář PostScriptového textu nahrazen několika různými objekty. Tato varianta byla použita u obrázku 3.7:

```
\psfrag{gA}[br][br]{|[br][br]|}
\psfrag*{gA}[Br][b][2]{|[Br][b][2]|}
\psfrag*{gA}[r][bl]{|[r][bl]|}
\psfrag*{gA}[tr][Bl]{|[tr][Bl]|}
\psfrag*{gA}[b][B]{|[b][B]|}
\psfrag*{gA}[B][Br]{|[B][Br]|}
\psfrag*{gA}[ ][r]{|[ ][r]|}
\psfrag*{gA}[t][ ][0.75][45]{|[t][ ][0.75][45]|}
\psfrag*{gA}[bl][l][1.5][30]{|[bl][l][1.5][30]|}
\psfrag*{gA}[Bl][tl]{|[Bl][tl]|}
\psfrag*{gA}[bl][Bl]{~~~~~(základní čára)}
\psfrag*{gA}[bl][l]{~~~~~(střední čára)}
\psfrag*{gA}[bl][t][1][-90]{~~~~~(střední čára)}
\psfrag*{gA}[l][t]{|[l][t]|}
\psfrag*{gA}[tl][tr][1][180]{|[tl][tr][1][180]|}
\resizebox{4in}{!}{\includegraphics[angle=30]{testfig.eps}}
```

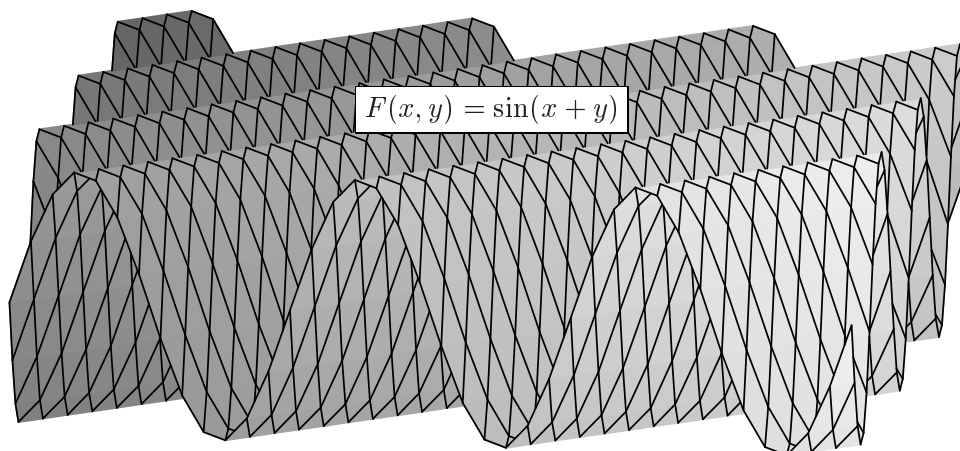
```
\begin{psfrags} \end{psfrags}
```

Prostředí `psfrags` může být použito, je-li to nezbytné, pro změnu hranice působnosti příkazu `\psfrag`. Jak již bylo řečeno, příkaz `\psfrag` účinkuje, dokud není ukončeno nejbližší okolní prostředí (`center`, `figure`, ...). Toto prostředí nemá v dokumentu žádný další efekt, proto není použití tohoto prostředí nezbytné.

■ Příklad 14.3.

U některých obrázků je třeba umístit popisku přes vybarvenou plochu:

```
\begin{figure}[hbt]
\psfrag{fce}{\fcolorbox{black}{white}{$F(x,y)=\sin(x+y)$}}
\includegraphics{funkce01.eps}
\caption{Popiska na grafu funkce je \uv{podložena} bílou barvou}
\end{figure}
```



Obrázek 3.8: Popiska na grafu funkce je „podložena“ bílou barvou

Vymezení vhodných návěstí

Vložíte-li do obrázku část textu (*návěstí*) pro jeho nahrazení pomocí balíku PSfrag, používejte *jednoduchá slova*, složená pouze z písmen a číslic. Vyvarujte se používání akcentů a speciálních znaků. Takto je téměř zaručeno, že bude balík pracovat tak, jak je zde popsáno. Přesto některé kreslicí balíky způsobí zaručeně problémy. Tyto problémy snadno odstraněny, jestliže pochopíte, jak PSfrag hledá návěstí.

PostScript používá pět příkazů pro zobrazování textu: **show**, **ashow**, **kshow**, **widthshow** a **awidthshow** (ale v mnoha případech bude EPS soubor obsahovat jen jejich zkratky.) PSfrag se pak snaží zachytit tyto příkazy. Jestliže je zachycený řetězec roven známému *návěstí*, PSfrag vloží *objekty*. Jestliže není, PSfrag zabrání pomocí příkazu ***show** postupovat normálně.

Řetězce, které jsou zobrazovány pomocí ***show** jsou uzavřeny do závorek. Například

```
(This is a test.) show zobrazí This is a test.
```

Nepárovým závorkám a jiným speciálním znakům musí v PostScriptovém řetězci předcházet zpětné lomítko - \. Například

```
(x = \ (0,1] ) show zobrazí x = (0,1]
```

Zde je obecné pravidlo pro PSfrag návěstí:

Návěstí nahrazované pomocí příkazu **\psfrag** musí vypadat přesně tak, jak to vyžaduje příkaz ***show** v EPS souboru, bez okolních závorek.

Jinými slovy, PSfrag bude fungovat, jestliže řetězec v příkazu `\psfrag` přesně odpovídá tomu, co je nalezeno v EPS souboru. Jestliže váš řetězec v EPS souboru obsahuje zpětná lomítka, budete je muset napsat také do argumentu příkazu `\psfrag`. PSfrag umí nahradit pouze celý řetězec, ne jeho část. Jestliže tedy váš EPS soubor obsahuje

```
(I want to replace the XXX here) show
```

pak příkaz `\psfrag` nebude fungovat, budete-li chtít nahradit pouze řetězec **XXX**.

Pro editaci EPS souborů lze používat váš oblíbený textový editor, neboť tyto soubory jsou právě jednoduché ASCII soubory.

Některé kreslicí knihovny bohužel zobrazují text tak, že zobrazují pomocí příkazu `show` *každý znak zvlášť*. Jinými slovy, jestliže použijete tento kreslicí nástroj k vložení řetězce „test“ do vašeho obrázku, bude to vypadat nějak takto:

```
(t) show (e) show (s) show (t) show
```

To má za následek, že užití balíku PSfrag je dosti nepohodlné, jste totiž limitováni *jednoznačnými* návěstími.

15. Obtékání grafických objektů textem

Předpoklady: balík `picinpar.sty`.

V tomto odstavci stručně popíšeme, jak obtékat grafické objekty textem pomocí balíku `picinpar.sty`.¹⁷

K vložení grafického objektu do textu slouží prostředí `figwindow`, popř. `tabwindow`. Prostředí `figwindow` se zpravidla používá pro obrázky, prostředí `tabwindow` pro tabulky. Tato prostředí jsou definována velmi podobně jako standardní L^AT_EXovská prostředí:

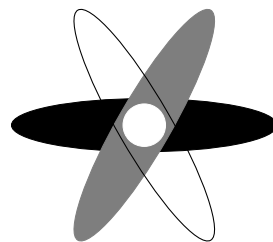
```
\begin{figwindow}[n,z,{objekt}]{popis}
...text...
\end{figwindow}
```

```
\begin{tabwindow}[n,z,{objekt}]{popis}
...text...
\end{tabwindow}
```

■ Příklad 15.1.

```
\begin{figwindow}[0,r,{\includegraphics{prik.eps}},{Hvězda}]
Uvnitř prostředí je třeba umístit text ....
\end{figwindow}
```

Uvnitř prostředí je třeba umístit text celého odstavce, do něhož má být grafický objekt začleněn. Všechny parametry jsou povinné. *Objektem* se rozumí vkládaný grafický objekt, kterým může být EPS soubor, prostředí `picture`, tabulka, `parbox` apod. Obrázky ve formátu EPS, popř. PS, je třeba vložit pomocí příkazu `\includegraphics` (viz str. 74). Parametr n značí počet řádků, po kolika bude grafický objekt začleněn do odstavce. Pokud má být hned na začátku, dosadíme číslo 0 (viz příklad 15.1). Parametr z určí, jak má být objekt začleněn do odstavce. Můžeme použít jednu ze tří možností: `r` – vpravo, `l` – vlevo, `c` – ve středu. Do posledních složených závorek umístíme *popis*. Jestliže nechceme žádný popis, necháme závorky prázdné. Prostředí `figwindow` čísluje obrázky v kombinaci s plovoucím prostředím pro obrázky `figure` (viz odstavec 13).



Obrázek 3.9: Hvězda

■ Příklad 15.2.

```
\begin{tabwindow}[2,c,{
\begin{tabular}[t]{|l|l|r|}
\hline
Šremr & Jiří & 1200 \\
Hakl & Robert & 450 \\
Maršálek & Zdeněk & 9850 \\
\hline\hline
}
```

¹⁷K obtékání grafických objektů textem bylo vytvořeno více balíků maker (např. `picins.sty`), ale nejlépe se nám osvědčil právě balík `picinpar.sty`, ačkoliv i ten má svá úskalí — viz strana 96.

```
& $\Sigma$ & 11500\\
\hline
\end{tabular}
},{ }
Analogicky prostředí ...
... nebo ji úplně odstranit.
\end{tabwindow}
```

Analogicky prostředí `tabwindow` čísluje tabulky v kombinaci s plovoucím prostředím pro tabulky `table`. Prostředí `tabular` (obdobně jako prostředí `picture` apod.) můžeme umístit

Šremr	Jiří	1200
Hakl	Robert	450
Maršálek	Zdeněk	9850
	Σ	11500

přímo jako *objekt* — viz příklad 15.2. Jestliže nechceme k tabulkám, popř. obrázkům, dávat žádný popis, není žádoucí, aby za číslem tabulky, popř. obrázku, byla dvojtečka. Odstavec 16 popisuje, jak nahradit dvojtečku tečkou nebo ji úplně odstranit.

Tabulka 3.3:

■ Příklad 15.3.

```
\begin{figwindow}[0,c,{\scalebox{7}{$\heartsuit$}},{}]
Tento příklad ukazuje ...
... jak to udělat..
\end{figwindow}
```

Tento příklad ukazuje, že vložené příkazů balíku `graphicx` (viz odstavec odkazem na experimentování čtenáře. byla dvojtečka za číslem obrázku zamě- jedna z možností, jak to udělat.



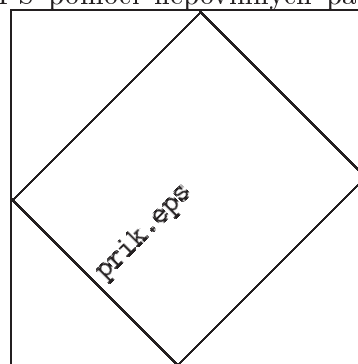
grafické objekty lze upravovat pomocí (viz odstavec 12.2). Podrobný popis vynecháme Všimněte si, že u tohoto obrázku již něna za tečku. V odstavci 16 je popsána

Obrázek 3.10.

■ Příklad 15.4.

```
\begin{figwindow}[1,r,{\includegraphics[draft,angle=45]{prik.eps}},{Otáčení}]
Analogicky lze upravovat ...
... vyznačen.
\end{figwindow}
```

Analogicky lze upravovat vkládanou grafiku ve formátu EPS pomocí nepovinných parametrů příkazu `\includegraphics`. Příklad 15.3 ukazuje, jak bude obtékán otočený obrázek, tj. obtékán bude nikoliv skutečný BoundingBox, ale obdélník, který je pro názornost na obrázku 3.11 vyznačen.



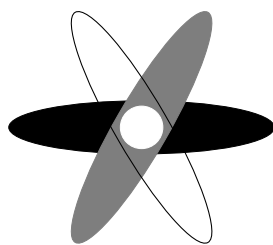
Obrázek 3.11: Otáčení

Několik postřehů k balíku picinpar

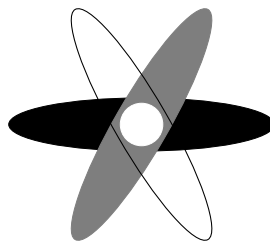
- ★ Pokud do odstavce, ve kterém je obtékáný obrázek vychází zlom stránky, chová se prostředí `figwindow` i `tabwindow` problematicky. Většinou jeden řádek textu nevysází.
- ★ Problémy nastanou také v případě, kdy je text, kterým má být obrázek obtékán nedostatečně dlouhý.
- ★ Než se rozhodnete pro obtékání obrázků textem, zvažte, jestli by nebylo vhodné umístit raději dva obrázky vedle sebe do jednoho prostředí `figure`, např. pomocí `minipage`. Takový způsob dělá daleko méně potíží zejména při přestránkování (viz příklad 15.5).

■ Příklad 15.5.

```
\begin{figure}[hb]
\renewcommand{\figurename}{Obr.}
\begin{minipage}{0.4\textwidth}
\centering
\includegraphics{prik.eps}
\caption{První obrázek}
\end{minipage}
\hfil
\begin{minipage}{0.4\textwidth}
\centering
\includegraphics{prik.eps}
\caption{Druhý obrázek}
\end{minipage}
\end{figure}
```



Obr. 3.12: První obrázek



Obr. 3.13: Druhý obrázek

16. Číslování obrázků

Odstranění teček za číslem obrázku lze provést s použitím balíku `caption2` jednoduchým příkazem:

```
\usepackage{caption2}
\renewcommand{\captionlabeldelim}{~}
```

Analogicky můžeme místo dvojtečky dosadit tečku příkazem

```
\renewcommand{\captionlabeldelim}{.}
```

Jak změnit popis obrázku (např. zkrátit slovo „Obrázek“ na „Obr.“) ukazuje příklad 15.5. Pokud používáme balík pro obtékání obrázků `picinpar` (viz odstavec 15), je třeba změnit adekvátně také definici `wincaption`, což je jednoduchá možnost je vyjmout z balíku `\makeatletter` a `\makeatother` uděláte vlastní úpravy analogicky jako změněno na kapitálky a odstraněna s popisem obrázku s obrázkem v pří-



OBRÁZEK 3.14

příkaz pro popisy obtékaných obrázků. líku `picinpar` odpovídající část a vložit ji do vašeho zdrojového textu. Snadno v příkladě 16.1, kde bylo tučné písmo tečka (srovnejte tento popis obrázku kladě 15.3).

■ Příklad 16.1.

```
%%% odstraneni tecek za Obrazek (ve figwindow)+zmena pisma
\makeatletter
\def\wincaption{\winrefstepcounter{\capytype
\w@dblarg{\@wincaption\@capytype}}

\long\def\@wincaption#1[#2]#3\par{\par\addcontentsline{\csname
ext@#1\endcsname}{#1}{\protect\numberline{\csname
the#1\endcsname}{\ignorespaces #2}}\begingroup
\@parboxrestore
\normalsize
\@makewincaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
\endgroup}

\long\def\@makewincaption#1#2{%
% \setbox\@tempboxa\hbox{\footnotesize{\bf #1:\enspace}{#2}}%
% zaprocentovaný řádek byl nahrazen řádkem následujícím:
\setbox\@tempboxa\hbox{\sc #1 \enspace}{#2}%
\ifdim\wd\@tempboxa>\picwd
% {\footnotesize{\bf #1:\enspace}{#2}\par}}
% zaprocentovaný řádek byl nahrazen řádkem následujícím:
{\footnotesize{\sc #1 \enspace}{#2}\par}}
\else
\hbox to\picwd{\hfil\box\@tempboxa\hfil}
\fi}
\makeatother
```

17. Balík mfpic

Prostředí `picture` má jen velmi omezené možnosti kreslení. Uvedené balíky sice dávají možnosti daleko větší, ale jsou vždy speciálně zaměřeny (např. `Xy-pic` – algebraické diagramy). Velmi široké možnosti ke kreslení obrázku nejen metematically zaměřených dává balík `mfpic`. Jak bylo uvedeno v úvodu, tento balík maker využívá pro kreslení `METAFont`, případně `METAPOST`, ale zdrojový kód se zapisuje do vstupního souboru `LaTeXu` a užití příkazu samotného `METAFontu` je pro uživatele skryto. Není třeba se učit programovací jazyk.

Podrobně je problematika tvorby obrázků v `METAPOSTu` rozebrána v diplomové práci M. Krátká *Tvorba obrázků pro matematické texty pomocí METAPOSTu* [5]. Práce je snadno dostupná v elektronické podobě na <http://mirka.janik.cz/dp/>. Zde uvádíme jen základní informace a několik příkladů ilustrujících rozsáhlé možnosti tohoto balíku.

Předpoklady:

- v preambuli `\usepackage[metapost,mplabels,truebbox]{mfpic}`,
- v adresáři prohledávaném `LaTeXem` (např. aktuální adresář) balíky `mfpic.sty`, `mfpic.tex`, `grafbase.mp`, `dvipsnam.mp`¹⁸

`Mfpic` sám (pro `LaTeX`) načítá balík `graphics` pro vkládání námi vytvořených obrázků, jež je prováděno takto:

```
\includegraphics{jmeno souboru}
```

Každý soubor využívající `mfpic` s `METAPOSTem` má v `LaTeXu` následující strukturu:

```
\documentclass{article}
\usepackage[metapost]{mfpic}%%% pokud neuvedeme nepovinný para-
                             %%% metr, bude použit pro tvorbu
                             %%% obrázků Metafont
%\usepackage{mfpic}  %%% tyto dva řádky jsou ekvivalentní s řád-
%\usemetapost        %%% kem předcházejícím

\begin{document}

\opengraphsfile{obrazek}

%libovolný zdrojový kód

\begin{mfpic}[a][b]{c}{d}{e}{f}
%příkazy popisující obrázek - výsledkem je obrazek.1
\end{mfpic}

%okolí mfpic lze zapisovat také takto:

\mfpic[g][h]{i}{j}{k}{l}
```

¹⁸<http://comp.uark.edu/~luecking/tex/mfpic.html> nebo [1].


```
%příkazy popisující obrázek - výsledkem je obrazek.2
\endmfpic

\closegraphsfile

\opengraphsfile{picture}

\mfpic[m][n]{o}{p}{q}{r}
%příkazy popisující obrázek - výsledkem je picture.1
\endmfpic

\closegraphsfile

\end{document}
```

Příkazem

```
\opengraphsfile{název_souboru}
```

otevíráme soubor, do kterého se zapisují příkazy METAPOSTu, příkaz

```
\closegraphsfile
```

tento soubor zavírá (tyto příkazy by měly být používány pouze v udaném tvaru, to jest ne jako okolí `\begin{okolí} ... \end{okolí}` v L^AT_EXu). Příkazy

```
\mfpic [měřítkona_ose_x] [měřítkona_ose_y] {xmin} {xmax} {ymin} {ymax}
```

```
...
```

```
\endmfpic
```

```
\begin{mfpic}[měřítkona_ose_x][měřítkona_ose_y]{xmin}{xmax}{ymin}{ymax}
```

```
...
```

```
\end{mfpic} (pouze v LATEXu)
```

uzavírají popis jednoho obrázku. Měřítko musí být zadáno alespoň na jedné ose; je-li zadáno pouze jedno, je měřítko shodné pro osu x i y . Čísla x_{\min} , x_{\max} určují rozsah osy x , čísla y_{\min} , y_{\max} určují rozsah osy y .

Nyní jsou dvě možnosti způsobu zpracování vstupního souboru:

Po překladu vstupního souboru L^AT_EXem vzniknou soubory `obrazek.mp` a `picture.mp`. Tyto soubory přeložíme programem `mpost` a poté znovu spustíme L^AT_EX. Výsledkem je soubor s příponou `.dvi`. Nejsou-li při jeho prohlížení viditelné obrázky (například `xdvi` na novějších verzích operačního systému Linux obrázky zobrazuje), vytvoříme příslušným programem (`dvips`) postscriptový soubor a prohlédneme si jej například pomocí Ghostview.

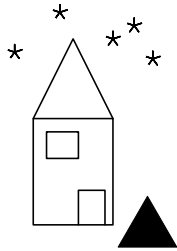
Volby mfpic

Pro diplomovou práci doporučujeme jednu z následujících kombinací:

```
\usepackage [metapost, mplabels, truebbox] {mfpic} ,
\usepackage [metapost, mplabels, truebbox, clip] {mfpic} .
```

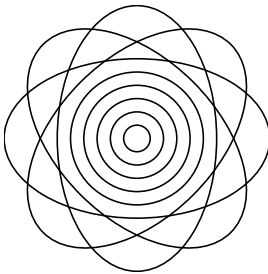
Volba `mplabels` zajistí, aby popisky zpracovával METAPOST; volba `truebbox` zajistí, aby METAPOST určil přesný bounding box obrázku a tento předal T_EXu (a tedy nevzniknou problémy například při obtékání obrázku); volba `clip` způsobí ořezání výsledného obrázku na rozměr námi zadaný (tudíž zamezíme zasahování okolního textu do obrázku).

■ Příklad 17.1.



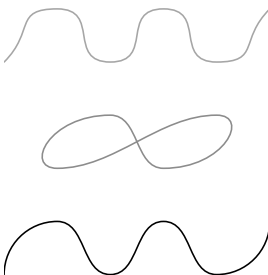
```
\begin{mpic}[10]{0}{10}{0}{10}
\rect{(2,1.5),(5,5.5)}
\polyline{(2,5.5),(3.5,8.5),(5,5.5)}
\lines{(3.7,1.5),(3.7,2.8),(4.7,2.8),(4.7,1.5)}
\polygon{(2.5,4),(2.5,5),(3.7,5),(3.7,4)}
\plotsymbol[17pt]{SolidTriangle}{(6.3,1.3)}
\plotsymbol[5pt]{Star}{(1.3,8),(3,9.5),
(5,8.5),(6.5,7.75),(5.8,9)}
\end{mpic}
```

■ Příklad 17.2.



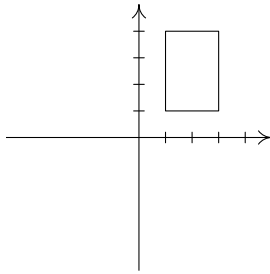
```
\begin{mpic}[10]{-5}{5}{-5}{5}
\circle{(0,0),2.5}
\circle{(0,0),2}
\circle{(0,0),1.5}
\circle{(0,0),1}
\circle{(0,0),0.5}
\ellipse{(0,0),5,3}
\ellipse[45]{(0,0),5,3}
\ellipse[90]{(0,0),5,3}
\ellipse[135]{(0,0),5,3}
\end{mpic}
```

■ Příklad 17.3.



```
\begin{mpic}[20]{0}{5}{0}{5}
\curve{(0,0),(1,1),(2,0),
(3,1),(4,0),(5,1)}
\draw[.55white]\cyclic{
(1,2),(2,3),(3,2),
(4,3)}
\draw[.65white]\fcncurve{
(0,4),(1,5),(2,4),
(3,5),(4,4),(5,5)}
\end{mpic}
```

■ Příklad 17.4.



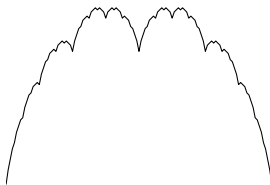
```
\mfpic[10]{-5}{5}{-5}{5}
\axes
\xmarks{1,2,3,4}
\ymarks{1,2,3,4}
\rect{(1,1),(3,4)}
\endmfpic
```

■ Příklad 17.5.



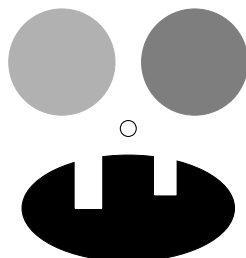
```
\mfpic[10]{0}{5}{0}{5}
\turtle{(1,4),(2,1),(2,-1),(0,-3),
(-3,0),(1,3),(1,-1),(0,-1),(-1,0.5)}
\endmfpic
```

■ Příklad 17.6.



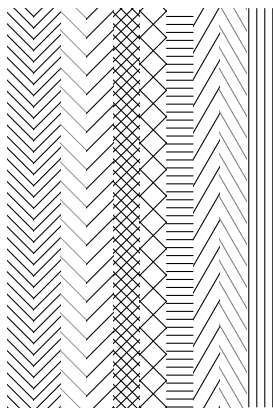
```
\mfpic[10]{0}{5}{0}{5}
\sector{(2.5,2.5),2,30,-30}
\sector{(2.5,2.5),2.5,45,135}
\sector{(2.5,2.5),1.5,150,190}
\sector{(2.5,2.5),1,220,290}
\curve[2]{(2.5,2.5),(3.5,1.5),(3,0)}
\endmfpic
```

■ Příklad 17.7.



```
\mfpic[10]{-5}{5}{-5}{5}
\gfill\ellipse{(0,-3),4,2}
\gclear\rect{(-2,-3),(-1,-1)}
\gfill[white]
\rect{(1,-2.5),(1.8,-1)}
\shade[0.9]\circle{(-2.5,2.5),2}
\shade[0.7]\circle{(2.5,2.5),2}
\circle{(0,0),0.3}
\endmfpic
```

■ Příklad 17.8.

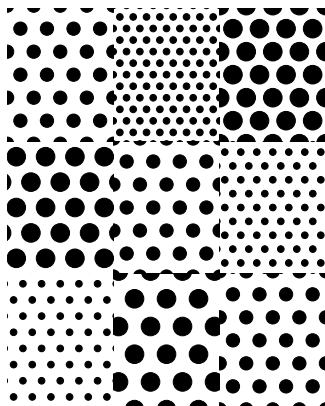


```

\mfpic[10]{0}{10}{0}{15}
\lhatch\rect{(0,0),(1,15)}
\rhatch\rect{(1,0),(2,15)}
\lhatch[5][.6white]
    \rect{(2,0),(3,15)}
\rhatch[5]\rect{(3,0),(4,15)}
\lhatch\rect{(4,0),(5,15)}
\lhatch[7]\rect{(5,0),(6,15)}
\lhatch\rect{(6,0),(7,15)}
\lhatch[5,60 deg]
    \rect{(7,0),(8,15)}
\lhatch[4,120 deg][.5white]
    \rect{(8,0),(9,15)}
\lhatch[3,90 deg]
    \rect{(9,0),(10,15)}
\endmfpic

```

■ Příklad 17.9.

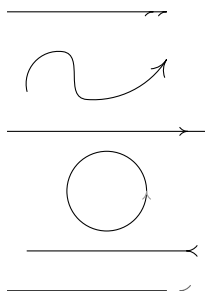


```

\mfpic[10]{0}{12}{0}{15}
\polkadot\rect{(0,10),(4,15)}
\polkadot\rect{(4,5),(8,10)}
\polkadot\rect{(8,0),(12,5)}
\polkadotwd{2.5pt}
\polkadot[5]
    \rect{(4,10),(8,15)}
\polkadot[6]
    \rect{(8,5),(12,10)}
\polkadot[7]\rect{(0,0),(4,5)}
\polkadotwd{7pt}
\polkadot\rect{(8,10),(12,15)}
\polkadot[11]
    \rect{(0,5),(4,10)}
\polkadot[12]
    \rect{(4,0),(8,5)}
\endmfpic

```

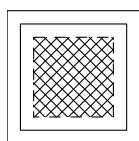
■ Příklad 17.10.



```
\mfpic[15]{0}{5}{0}{7}
\arrow\arrow[b 5pt]\lines{
    (0,7),(4,7)}
\arrow[l 6pt]\curve{(0.5,5),(1.4,6),
    (2,4.8),(4,5.8)}
\arrow[b 7pt]\lines{(0,4),(5,4)}
\arrow[c .6white]\circle{
    (2.5,2.5),1}
\arrow[r180][14pt]
    \lines{(0.5,1),(4.5,1)}
\arrow[r180][14pt][b5pt][c.4white]
    \lines{(0,0),(4,0)}

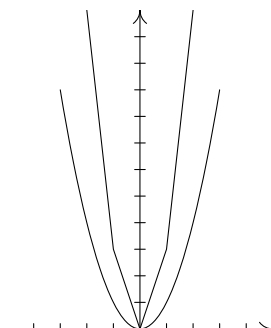
\endmfpic
```

■ Příklad 17.11.



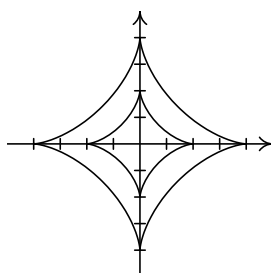
```
\mfpic[10]{0}{5}{0}{5}
\rect{(0,0),(5,5)} % \draw není nutné
\draw\rect{(.5,.5),(4.5,4.5)}
\setrender{\dashed\XHatch}
\rect{(1,1),(4,4)}
\endmfpic
```

■ Příklad 17.12.



```
\mfpic[10]{5}{5}{0}{12}
\axes
\Xmarks{-4,-3,-2,-1,1,2,3,4}
\Ymarks{1,2,3,4,5,6,7,8,9,10,11}
\function{-3,3,.5}{x*x}
\function[p]{-2,2,1}{3*x*x}
\endmfpic
```

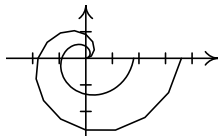
■ Příklad 17.13.



```
\mfpic[10]{-5}{5}{-5}{5}
\axes
\Xmarks{-4,-3,-2,-1,1,2,3,4}
\Ymarks{-4,-3,-2,-1,1,2,3,4}
\parafcn{0,1440,22.5}{
    ((3*cosd (t/4))+cosd ((3*t)/4),
    (3*sind (t/4))-sind ((3*t)/4))}
\parafcn[p]{0,1440,22.5}{
    (.5*((3*cosd (t/4))+
        cosd ((3*t)/4)),
    .5*((3*sind(t/4))-
        sind ((3*t)/4)))}

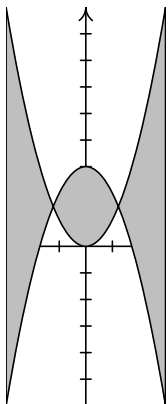
\endmfpic
```

■ Příklad 17.14.



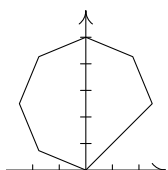
```
\mfpic[10]{-3}{5}{-3}{2}
\axes
\xmarks{-2,-1,1,2,3,4}
\ymarks{-2,-1,1}
\plrfcn{0,360,22.5}{.005*t}
\plrfcn[p]{0,360,22.5}{.01*t}
\endmfpic
```

■ Příklad 17.15.



```
\mfpic[10]{-3}{3}{-6}{9}
\axes
\xmarks{-2,-1,1,2}
\ymarks{-5,-4,-3,-2,-1,1,2,3,4,5,6,7,8}
\shade\btwnfcn[s]{-3,3,.5}{x*x}{-x*x+3}
\btwnfcn[s]{-3,3,.5}{x*x}{-x*x+3}
\endmfpic
```

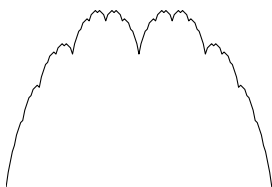
■ Příklad 17.16.



```
\mfpic[10]{-3}{3}{0}{6}
\axes
\xmarks{-2,-1,1,2}
\ymarks{1,2,3,4,5}
\plrregion{45,180,22.5}{5*sind(t)}
\endmfpic
```

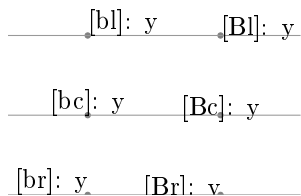
■ Příklad 17.17.

Funkce na následujícím obrázku je sedmým částečným součtem řady, která konverguje ke spojitě funkci nemající nikde derivaci. V `mfpic` je znázornění takové funkce velmi jednoduché:



```
\mfpic[100]{0}{1}{0}{1}
\fdef{f}{x}{min(x-(floor x),
(ceiling x)-x)}
\function[p]{0,1,1/128}{f(x)+
(.5*f(2*x))+
(.25*f(4*x))+
(.125*f(8*x))+
(.0625*f(16*x))+
(.03125*f(32*x))+
(.015625*f(64*x))}
\endmfpic
```

■ Příklad 17.18.

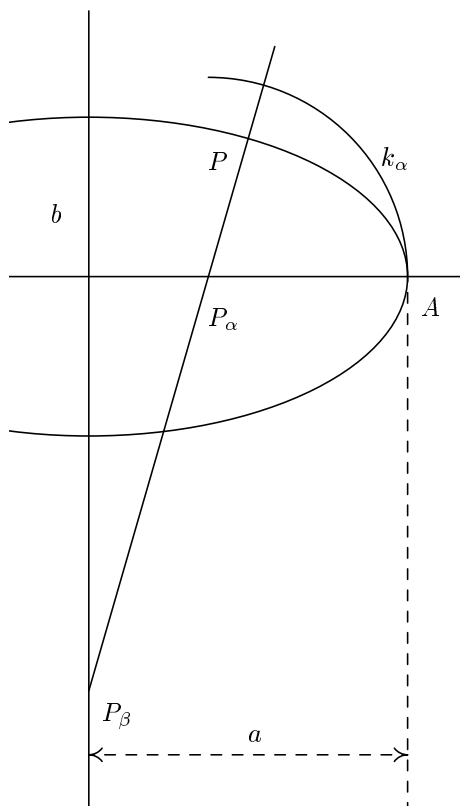


```

\mfpic[10][-3]{8}{-6}{1}
\fillcolor{.5white}
\point{(0,0),(5,0),(0,-3),
      (5,-3),(0,-6),(5,-6)}
\draw[.6white]
      \lines{(-3,0),(8,0)}
\draw[.6white]
      \lines{(-3,-3),(8,-3)}
\draw[.6white]
      \lines{(-3,-6),(8,-6)}
\tlabel[bl](0,0){[bl]: y}
\tlabel[Bl](5,0){[Bl]: y}
\tlabel[bc](0,-3){[bc]: y}
\tlabel[Bc](5,-3){[Bc]: y}
\tlabel[br](0,-6){[br]: y}
\tlabel[Br](5,-6){[Br]: y}
\end{mfpic}

```

■ Příklad 17.19.



```

\begin{mfpic}[10][-3]{14}{-20}{10}
\pointdef{A}(12,0)
\pointdef{B}(0,6)
\pointdef{P}(6,sqrt{27})
\pointdef{O}(0,0)
\lines{(0,-sqrt{243}),\P}
\turtle{\P,(1,sqrt{12})}
\lines{(-5,0),(14,0)}
\lines{(0,-20),(0,10)}
\ellipse{O,12,6}
\arc[c]{(4.5,0),(12,0),90}
\dashed\lines{A,(12,-20)}
\dashed\arrow\reverse\arrow
\lines{(0,-18),(12,-18)}
\tlabel[br](-1,2){$b$}
\tlabel[bl](12.5,-1.5){$A$}
\tlabel[bl](4.5,-2){$P_{\alpha}$}
\tlabel[bl](6,-17.5){$a$}
\tlabel[bl](0.5,-17){$P_{\beta}$}
\tlabel[bl](11,4){$k_{\alpha}$}
\tlabel[bl](4.5,4){$P$}
\end{mfpic}

```



```

\arc[p]{A,-alpha,0,.7}
\point{(.35*dir 225)}
\arc[p]{.5[A,B],180,270,.7}
\arc[p]{S,270-alpha,270,.7}
\draw[.65white]\arc[p]{S,alpha-90,270-alpha,abs(S-A)}
\coords
\mirror{A}{B}
\draw[.65white]\arc[p]{S,alpha-90,270-alpha,abs(S-A)}
\endcoords
\mfsrc{X=S+abs(S-A)*dir(77);}
\pen{1.3pt}      \polygon{A,B,X}
\pen{.5pt}
%\mfsrc{path a;    %% vyznačení úhlu lze nakreslit také takto
%a=fullcircle scaled 42pt shifted zconv (X);
%draw subpath (ypart (zconv ((X--A)) intersectiontimes a),
%ypart(zconv((X--B)) intersectiontimes a)) of a;}%
\mfsrc{x=angle(X-A); y=angle(X-B);}
\arc[p]{X,x+180,y+180,.7}
\tlabel[cr](xpart A -.3,ypart A){$A$}
\tlabel[cl](xpart B +.3,ypart B){$B$}
\tlabel[bl](xpart C +.1,ypart C){$C$}
\tlabel[bl](xpart S +.1,ypart S){$S_1$}
\tlabel[cl](xpart 0 +.1,ypart 0){$S_2$}
\tlabel[bl](0.1,0.1){$M$}
\tlabel[bc](xpart X,ypart X +.1){$X$}
\tlabel[br](xpart .5[A,S] -.1,ypart .5[A,S]){$r$}
\tlabel[bl](xpart .5[B,S] +.1,ypart .5[A,S]){$r$}
\mfsrc{Y=S+abs(S-A)*dir(60);}
\tlabel[bl](xpart Y +.1,ypart Y){$k_1$}
\tlabel[tl](xpart Y +.1,-ypart Y){$k_2$}
\mfsrc{pair f,g,h;f=.9*dir((x+y)/2+180)+X;
      g=.9*dir(-alpha/2)+A;
      h=.9*dir(270-(alpha/2))+S;}
\tlabel[cc](xpart f,ypart f){$\alpha$}
\tlabel[cc](xpart g,ypart g){$\alpha$}
\tlabel[cc](xpart h,ypart h){$\alpha$}
\endmfpic

```

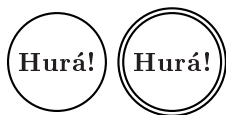
18. Balík PStricks

Na závěr příklad balíku, který v sobě spojuje výhody \LaTeX ých maker a PostScriptu.

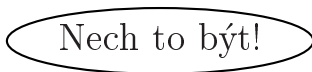
■ Příklad 18.1.



```
\psdiabox{Pozor zvíře!}
```



```
\pscirclebox{\bfseries Hurá!}  
\pscirclebox[doubleline=true]  
{\bfseries Hurá!}
```

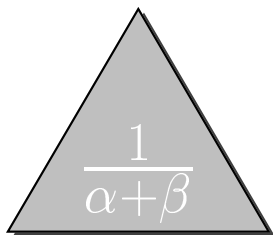


```
\psovalbox{\Large Nech to být!}
```

Grafická efekty, které umožňuje PStricks, lze umocnit ještě užitím barev. Kromě balíku PStricks je třeba načíst ještě balík pscolor (definice barev analogicky jako u color - viz strana 67).

Uvedme několik variant `\psframebox`:

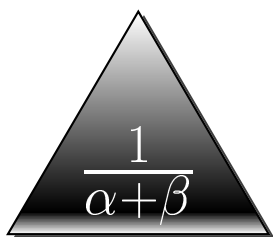
■ Příklad 18.2.



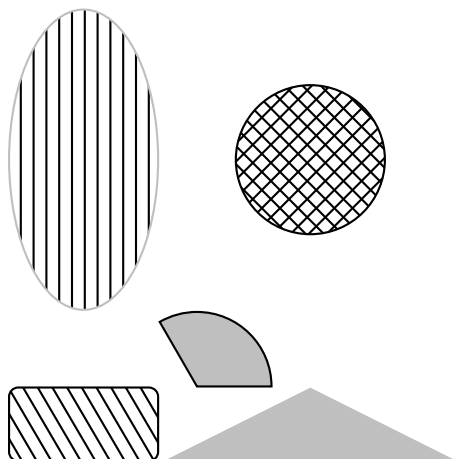
```
\pstribox[shadow=true,fillstyle=solid,%  
fillcolor=lightgray]  
{\color{white}\Huge $\frac{1}{\alpha + \beta}$}
```

Balík grad umožňuje vytvářet barevné přechody:

■ Příklad 18.3.



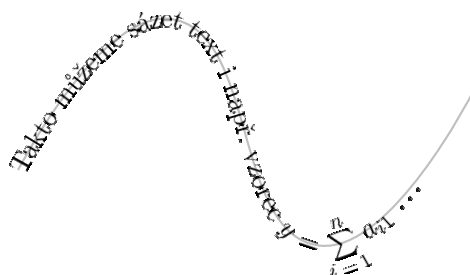
```
\pstribox[shadow=true,fillstyle=gradient,%  
gradbegin=white,gradend=black]  
{\color{white}\Huge $\frac{1}{\alpha + \beta}$}
```



```
\begin{pspicture}(0,0)(6,7)
\psframe[framearc=0.25,
         fillstyle=vlines,
         hatchangle=30]
(0,0)(2,1)
\pswedge[fillstyle=solid,
         fillcolor=lightgray]
(2.5,1){1}{0}{120}
\psellipse[linecolor=lightgray,
          fillstyle=vlines,
          hatchangle=0]
(1,4)(1,2)
\pscicle[fillstyle=crosshatch]
(4,4){1}
\pstriangle[linestyle=none,
           fillstyle=solid,
           fillcolor=lightgray]
(4,0)(4,1)
\end{pspicture}
```

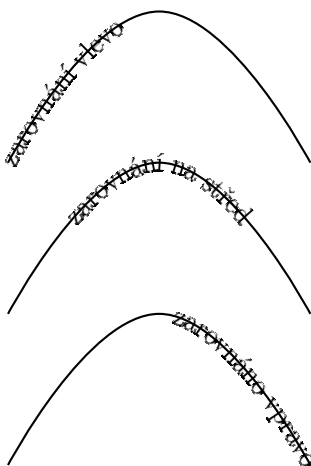
Pomocí balíku maker `pst-text` lze vysázet text na křivku:

■ Příklad 18.4.



```
\begin{pspicture}(-4,-3.2)(3,0.2)
\psset{linecolor=lightgray}
\pstextpath
{\pscurve(-4,-2)(-2,0)(0,-3)(2,-1)}
{Takto můžeme sázet text i např. vzorec}
$y=\sum\limits_{i=1}^n a_i$ \ldots
\end{pspicture}
```

■ Příklad 18.5.



```
\begin{pspicture}(0,0)(4.2,6.2)
\pstextpath[r]
{\pscurve(0,0)(2,2)(4,0)}
{zarovnáno vpravo}
\pstextpath[c]
{\pscurve(0,2)(2,4)(4,2)}
{zarovnáno na střed}
\pstextpath[l]
{\pscurve(0,4)(2,6)(4,4)}
{zarovnáno vlevo}
\end{pspicture}
```

Literatura

- [1] <http://www.cstug.cz>
- [2] Rybička J.: *L^AT_EX pro začátečníky*, 2. vydání, KONVOJ, Brno 1999.
- [3] Goossens M., Rahtz S., Mittelbach F.: *The L^AT_EX Graphics Companion*, 1997
- [4] Kristoffer H. Rose, X_Y-pic User's Guide, 1996, (<http://www.cstug.cz>)
- [5] M. Krátká *Tvorba obrázků pro matematické texty pomocí METAPOSTu*, diplomová práce, PřF MU Brno, 2001.
(v elektronické podobě na <http://mirka.janik.cz/dp/>.)
- [6] Petr Olšák: Jak T_EXpracuje s PostScriptem, Zpravodaj CStug u 3/1993
- [7] Goossens M., Rahtz S., Mittelbach F.: *The L^AT_EX Companion*, 1994.
- [8] Goossens M., Rahtz S., Mittelbach F.: *The L^AT_EX Graphics Companion*, 1997.
- [9] Grätzer G.: *Math into T_EX*, Birkhäuser Boston 1993.
- [10] Chlebíková J.: *Manuál k A_MS-T_EXu verze 2.1*.
<http://www.edi.fmph.uniba.sk/chlebikova/amsmanual/index.html>
- [11] Lichá M., Ulrych O.: *A_MS-T_EX verze 2.1*, Praha 1992.
- [12] Moravec D.: *Balíček fancyhdr.sty*, Zpravodaj Československého sdružení uživatelů T_EXu, **11** (4), 186–195 (2001).
- [13] Oostrum P.: *Page layout in L^AT_EX*.
CTAN/macros/latex/contrib/supported/fancyhdr/fancyhdr.tex
- [14] Reckdahl K.: *Using imported graphics in L^AT_EX 2_ε*.
CTAN/info/epslatex.ps nebo CTAN/info/epslatex.pdf
- [15] *User's Guide to AMSFonts Version 2.2d*, AMS 2002.
<ftp://ftp.ams.org/pub/tex/doc/amsfonts/amsfndoc.pdf>
- [16] *User's Guide for the amsmath Package*, AMS 1999.
<ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf>
- [17] *Using the amsthm Package*, AMS 2000.
<ftp://ftp.ams.org/pub/tex/doc/amscs/amsthdoc.pdf>

Rejstřík

a4wide, 6
\accentedsymbol, 25
\addtolength, 8
algebraických diagramů, 46
amsbsy, 15
amsmath, 5, 6, 11, 13, 15, 27, 28, 32, 44, 101
amssymb, 15, 16
amstex, 29
amsthm, 11, 101
amsxtra, 25
angle, 65
\ar, 54
array, 10
array, 33

barvy, 58
battomnumber, 76
bb, 65
\Big, 31
\big, 31, 32
\Bigg, 31
\bigg, 31
\binom, 29, 30
Bmatrix, 34
bmatrix, 34
\boldsymbol, 15, 16
\bottomfraction, 76
BoundingBox, 61–63, 66
\boxed, 26

caption2, 88
cases, 36
center, 82
\cfrac, 30
\circle, 48, 52, 71
\circle*, 48
clip, 65, 67
\closegraphsfile, 90

color, 58
\color, 58
\colorbox, 58
czech, 6

\dashbox, 47
\dbinom, 29
\dblfloatpagefraction, 77
\dblfloatsep, 77
\dbltextfloatsep, 77
\dbltopfraction, 77
dbltopnumber, 76
dcolumn, 11
\ddddot, 26
\ddot, 26
\DeclareMathOperator, 27
\DeclareMathOperator*, 27
\DeclareMathSymbol, 16
\definecolor, 58
\dfrac, 29
displaymath, 14
\documentclass, 13
\dot, 26
draft, 65, 67
DVI, 63
dvips, 60

\endmfpic, 90
epsf, 63
epsfig, 63
\eqref, 44
equation, 14
equation*, 14
eucal, 15

fancyhdr, 8, 9
\fcolorbox, 58
figure, 78

figure, 76, 78, 79, 82, 85
 figwindow, 85
 float, 77
 \floatpagefraction, 76
 \floatsep, 77
 \frac, 29, 30
 \frame, 67
 \framebox, 47

 \genfrac, 30
 grad, 99
 graphics, 60, 63
 graphicx, 60, 63, 65, 67, 70, 75
 graphpap, 52
 \graphpaper, 52

 \hat, 25
 \hdotsfor, 34
 height, 65
 hhline, 11
 \hole, 55

 \idotsint, 23
 \iint, 23
 \iint, 23
 ImageMagick, 61
 \includegraphics, 63, 65–67, 70, 72
 \intextsep, 77

 Kružnice, 48, 52
 křížové odkazy, 12

 \label, 12
 \left, 31, 32
 \leftroot, 26
 \line, 47
 \linethickness, 48
 longtable, 11

 \makeatletter, 88
 \makeatother, 88
 \makebox, 47
 Maple, 60
 \markboth, 10
 math, 14
 \mathbb, 15
 \mathbf, 15
 \mathcal, 15
 \mathfrak, 15
 \mathop, 16
 \mathscr, 15
 \mathstrut, 26
 matrix, 34
 \medspace, 33
 mfpic, 46, 89
 \mfpic, 90
 minipage, 78
 \mspace, 32
 \multiput, 47, 48

 \negmedspace, 33
 \negthickspace, 33
 \negthinspace, 33
 \newcommand, 25
 \newtheorem, 11
 \newtheorem*, 11
 \newtheoremstyle, 11
 noclip, 65, 67
 \numberwithin, 14, 44

 \opengraphsfile, 90
 origin, 65, 66
 \oval, 48
 \overleftarrow, 23
 \overrightarrow, 23
 \overset, 26

 \pagenumbering, 8
 \pageref, 12
 \pagestyle, 8, 9
 picinpar, 88
 pict2e, 52
 picture, 49
 picture, 46, 86
 pmatrix, 34
 \pmb, 15, 16
 ps2epsi, 63
 pscol, 99
 psfig, 63
 PSfrag, 79, 83, 84
 \psfrag, 79–84
 psfrags, 82
 pspicture, 48, 52
 pst-text, 100
 \put, 47, 71

`\qbezier`, 48
`\quad`, 33
`\quad`, 33
`\ref`, 12
 referenční bod, 63, 75
`\reflectbox`, 70
`\renewcommand`, 77
`\resizebox`, 71
`\resizebox*`, 71
`\right`, 31, 32
`\rotatebox`, 63, 73
 scale, 65
`\scalebox`, 63, 70, 71
`\sectionmark`, 10
`\setcounter`, 44, 77
`\setlength`, 8, 46, 77
`\sideset`, 27
 matrix, 34
`\smash`, 27
 split, 13
 subarray, 36
 subequations, 44
 subfigure, 78
`\substack`, 36
 supertab, 11
`\swapnumbers`, 11
 table, 76, 86
 tabular, 10, 86
 tabularx, 10
 tabwindow, 85, 86
`\tag`, 14, 37
`\tag*`, 37
`\tbinom`, 29
`\text`, 27
`\textcolor`, 58
`\textfloatsep`, 77
`\textfraction`, 76
`\tfrac`, 29
`\thesection`, 10
`\thicklines`, 48
`\thickspace`, 33
`\thinlines`, 48
`\thinspace`, 33
`\thispagestyle`, 10
`\tilde`, 25
`\topfraction`, 76
 topnumber, 76
 totalheight, 65
 totalnumber, 76
`\underset`, 26
`\unitlength`, 46
`\uproot`, 26
`\usepackage`, 6, 9, 13
`\varinjlim`, 28
`\varliminf`, 28
`\varlimsup`, 28
`\varprojlim`, 28
`\vector`, 47
 viewport, 65, 67
 Vmatrix, 34
 vmatrix, 34
 výřez obrázku, 67
`\widehat`, 25
`\widetilde`, 25
 width, 65
 wincaption, 88
 xfig, 60
`\xleftarrow`, 24
`\xrightarrow`, 24
 xv, 61
`\xymatrix`, 54
 xypic, 54
 základní čára, 63

Seznam balíků

a4wide, 6
 amsbsy, 15
 amsmath, 5, 6, 11, 13, 15, 27, 28, 32, 44, 101
 amssymb, 15, 16
 amstex, 29
 amsthm, 11, 101
 amsxtra, 25
 array, 10
 caption2, 88
 color, 58
 czech, 6
 dcolumn, 11
 epsf, 63
 epsfig, 63
 eucal, 15
 fancyhdr, 8, 9
 figure, 78
 float, 77
 grad, 99
 graphics, 60, 63
 graphicx, 60, 63, 65, 67, 70, 75
 graphpap, 52
 hhline, 11
 longtable, 11
 mfpic, 46, 89
 picinpar, 88
 pict2e, 52
 picture, 49
 pscol, 99
 psfig, 63
 PSfrag, 79, 83, 84
 pspicture, 48, 52
 pst-text, 100
 subfigure, 78
 supertab, 11
 tabularx, 10
 xypic, 54

Seznam prostředí

Bmatrix, 34
 Vmatrix, 34
 array, 33
 bmatrix, 34
 cases, 36
 center, 82
 displaymath, 14
 equation*, 14
 equation, 14
 figure, 76, 78, 79, 82, 85
 figwindow, 85
 math, 14
 matrix, 34
 minipage, 78
 picture, 46, 86
 pmatrix, 34
 psfrags, 82
 smallmatrix, 34
 split, 13
 subarray, 36
 subequations, 44
 table, 76, 86
 tabular, 10, 86
 tabwindow, 85, 86
 vmatrix, 34
 wincaption, 88
 \put, 47, 71

Seznam příkazů

<code>\accentedsymbol</code> , 25	<code>\idotsint</code> , 23
<code>\addtolength</code> , 8	<code>\iiint</code> , 23
<code>\ar</code> , 54	<code>\iint</code> , 23
<code>\Big</code> , 31	<code>\includegraphics</code> , 63, 65–67, 70, 72, 79
<code>\big</code> , 31, 32	<code>\intertextsep</code> , 77
<code>\Bigg</code> , 31	<code>\label</code> , 12
<code>\bigg</code> , 31	<code>\left</code> , 31, 32
<code>\binom</code> , 29, 30	<code>\leftroot</code> , 26
<code>\boldsymbol</code> , 15, 16	<code>\line</code> , 47
<code>\bottomfraction</code> , 76	<code>\linethickness</code> , 48
<code>\boxed</code> , 26	<code>\makeatletter</code> , 88
<code>\cfrac</code> , 30	<code>\makeatother</code> , 88
<code>\circle</code> , 48, 52, 71	<code>\makebox</code> , 47
<code>\circle*</code> , 48	<code>\markboth</code> , 10
<code>\closegraphsfile</code> , 90	<code>\mathbb</code> , 15
<code>\color</code> , 58	<code>\mathbf</code> , 15
<code>\colorbox</code> , 58	<code>\mathcal</code> , 15
<code>\dashbox</code> , 47	<code>\mathfrak</code> , 15
<code>\dbinom</code> , 29	<code>\mathop</code> , 16
<code>\dblfloatpagefraction</code> , 77	<code>\mathscr</code> , 15
<code>\dblfloatsep</code> , 77	<code>\mathstrut</code> , 26
<code>\dbltextfloatsep</code> , 77	<code>\medspace</code> , 33
<code>\dbltopfraction</code> , 77	<code>\mfpic</code> , 90
<code>\ddddot</code> , 26	<code>\mspace</code> , 32
<code>\ddot</code> , 26	<code>\multput</code> , 47, 48
<code>\DeclareMathOperator</code> , 27	<code>\negmedspace</code> , 33
<code>\DeclareMathOperator*</code> , 27	<code>\negthickspace</code> , 33
<code>\DeclareMathSymbol</code> , 16	<code>\negthinspace</code> , 33
<code>\definecolor</code> , 58	<code>\newcommand</code> , 25
<code>\dfrac</code> , 29	<code>\newtheorem</code> , 11
<code>\documentclass</code> , 13	<code>\newtheorem*</code> , 11
<code>\dot</code> , 26	<code>\newtheoremstyle</code> , 11
<code>\endmfpic</code> , 90	<code>\numberwithin</code> , 14, 44
<code>\eqref</code> , 44	<code>\opengraphsfile</code> , 90
<code>\fcolorbox</code> , 58	<code>\oval</code> , 48
<code>\floatpagefraction</code> , 76	<code>\overleftarrow</code> , 23
<code>\floatsep</code> , 77	<code>\overrightarrow</code> , 23
<code>\frac</code> , 29, 30	<code>\overset</code> , 26
<code>\frame</code> , 67	<code>\pagenumbering</code> , 8
<code>\framebox</code> , 47	<code>\pageref</code> , 12
<code>\genfrac</code> , 30	<code>\pagestyle</code> , 8, 9
<code>\graphpaper</code> , 52	<code>\pmb</code> , 15, 16
<code>\hat</code> , 25	<code>\psfrag</code> , 79–84
<code>\hdotsfor</code> , 34	<code>\put</code> , 47, 71
<code>\hole</code> , 55	<code>\qbezier</code> , 48
	<code>\qquad</code> , 33

`\quad`, 33
`\ref`, 12
`\reflectbox`, 70
`\renewcommand`, 77
`\resizebox`, 71
`\resizebox*`, 71
`\right`, 31, 32
`\rotatebox`, 63, 73
`\scalebox`, 63, 70, 71
`\sectionmark`, 10
`\setcounter`, 44, 77
`\setlength`, 8, 46, 77
`\sideset`, 27
`\smash`, 27
`\substack`, 36
`\swapnumbers`, 11
`\tag`, 14, 37
`\tag*`, 37
`\tbinom`, 29
`\text`, 27
`\textcolor`, 58
`\textfloatsep`, 77
`\textfraction`, 76
`\tfrac`, 29
`\thesection`, 10
`\thicklines`, 48
`\thickspace`, 33
`\thinlines`, 48
`\thinspace`, 33
`\thispagestyle`, 10
`\tilde`, 25
`\topfraction`, 76
`\underset`, 26
`\unitlength`, 46
`\uproot`, 26
`\usepackage`, 6, 9, 13
`\varinjlim`, 28
`\varliminf`, 28
`\varlimsup`, 28
`\varprojlim`, 28
`\vector`, 47
`\widehat`, 25
`\widetilde`, 25
`\xleftarrow`, 24
`\xrightarrow`, 24
`\xymatrix`, 54