

Vizualizace grafů matematických funkcí v systému LATEX

František Hanko

Bakalářská práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **František HANKO**

Osobní číslo: **A06146**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Téma práce: **Vizualizace grafů matematických funkcí v systému LATEX**

Zásady pro vypracování:

1. Podrobně se seznámte se sázecím systémem LATEX.
2. Vypracujte přehled nástrojů, které je možno použít v systému LATEX pro tvorbu vektorových schémat obsahující grafy matematických funkcí.
3. Stručně nástroje popište a porovnejte jejich možnosti.
4. Vyberte nejvhodnější z nich a s jeho pomocí realizujte zadaná grafická schémata.
5. Zpracujte přehledný návod pro tvorbu zadaných grafických schémat v LATEXu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. GOOSSENS, Michel, RAHTZ, Sebastian, MITTELBACH, Frank. The LaTeX graphics companion : illustrating documents with TeX and PostScript. Boston : Addison-Wesley, 1997. 554 s. ISBN 0201854694.
2. KOPKA, Helmut, DALY, Patrick W. Latex – kompletní průvodce. Brno : Computer Press, 2004. 584 s. ISBN 80-7226-6973-9.
3. OLŠÁK, Petr. TEXbook naruby. 2. vyd. Brno : Konvoj, 2001. 468 s. ISBN 80-7302-007-6.
4. RYBIČKA, Jiří. LATEX pro začátečníky. 3. vyd. Brno : Konvoj + CSTUG, 2003. 240 s. ISBN 80-7302-049-1.
5. TANTAU, Till. TikZ & PGF. [s.l.] : [s.n.], 2008. 560 s. Dostupný z WWW: <http://ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>.

Vedoucí bakalářské práce: **Mgr. Lubomír Sedláček, Ph.D.**
Ústav matematiky

Datum zadání bakalářské práce: **5. března 2010**

Termín odevzdání bakalářské práce: **1. června 2010**

Ve Zlíně dne 5. března 2010


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce bylo vytvořit zadaná grafická schémata v sázecím programu \LaTeX . Teoretická část se zabývá vytvořením přehledu grafických možností v \LaTeX u a porovnáním těchto nástrojů. Praktická část je tvořena základním přehledem tvorby grafických schémat pomocí zvoleného nástroje a zpracovanými grafickými schématy s popisem jejich tvorby. Výstupem jsou spustitelné zdrojové kódy jednotlivých grafických schémat.

Klíčová slova: \LaTeX , \TeX , grafická schémata, tvorba grafických schémat, `picture`, `PSTricks`, `TikZ & PGF`

ABSTRACT

The aim of this bachelor work was to create assigned graphical schemes in typesetting computer program \LaTeX . Theoretical part deals with creation of overview of graphical possibilities in \LaTeX and comparison of these tools. Practical part consists of elementary overview of creation of graphical schemes with help of selected tool and of processed graphic schemes with description of their creation. The output are executable source codes of individual graphical schemes.

Keywords: \LaTeX , \TeX , graphical schemes, creation of graphical schemes, picture inserting, `picture`, `PSTricks`, `TikZ & PGF`

Děkuji vedoucímu bakalářské práce Mgr. Lubomíru Sedláčkovi Ph.D. za pedagogickou pomoc a konzultace a Mgr. Vladimíru Poláškov, Ph.D. za konzultace během vypracovávání bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo - bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....
podpis diplomanta

Obsah

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 PRVNÍ KROKY V L^AT_EXU	10
2 NÁSTROJE PRO TVORBU GRAFICKÝCH SCHÉMAT	13
2.1 VKLÁDÁNÍ OBRÁZKŮ TO TEXTU	14
2.2 GRAFICKÉ PROSTŘEDÍ PICTURE	16
2.3 GRAFICKÉ PROSTŘEDÍ PSTricks	19
2.4 GRAFICKÉ PROSTŘEDÍ TikZ & PGF	21
3 SROVNÁNÍ GRAFICKÝCH PROSTŘEDÍ	22
II ANALYTICKÁ ČÁST	24
4 GRAFICKÁ SCHÉMATA	25
4.1 ÚVOD DO TVORBY GRAFICKÝCH SCHÉMAT	25
4.2 VRSTEVNICE FUNKCE $z = 8 - \frac{x^2}{2} - \frac{y^2}{2}$	32
4.3 ŘEZ PARABOLOIDU $z = a - (x^2 + y^2)$ ROVINOU $z = z_0$ ($a > z_0$)	38
4.4 DVOJNÝ INTEGRÁL NA OBDÉLNÍKU	39
4.5 DEFINIČNÍ OBOR FUNKCE DVOU PROMĚNNÝCH $z = f(x, y)$	40
4.6 ŘEZ KVÁDRU OBECNOU ROVINOU ρ	41
4.7 PARCIÁLNÍ DERIVACE FUNKCE DVOU PROMĚNNÝCH	42
4.8 VÁLCOVÉ (CYLINDRICKÉ) SOUŘADNICE	43
4.9 KULOVÉ (SFÉRICKÉ) SOUŘADNICE	45
ZÁVĚR	47
CONCLUSION	48
SEZNAM POUŽITÉ LITERATURY	49
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	50
SEZNAM OBRÁZKŮ	51
SEZNAM PŘÍLOH	52

ÚVOD

Problematika vkládání obrázků, grafů a schémat do textu patří k základním funkcím textových procesorů. Každý software tohoto typu má své specifické vlastnosti. Jedním z nich je program \LaTeX .

\LaTeX je sázecí program pro tvorbu vědeckých a technických dokumentů ve vysoké typografické kvalitě. Formátovací systém \LaTeX umožňuje vytvářet velké množství tiskovin od dopisů po složité technické příručky a knihy. Jedno z největších uplatnění nachází při tvorbě matematických textů.

\LaTeX vychází ze svého předchůdce \TeX u, který naprogramoval v 70. letech 20. století německý profesor Donald Erwin Knuth, jež byl nespokojen především se sazbou matematických vzorců v běžně používaných programech té doby. Název je odvozen z prvních tří písmen řeckého slova „technika“ čili $\tau\epsilon\chi$. Z tohoto důvodu vyslovujeme „tex“ jako „tech“ (v angličtině také „tek“). Samotný \TeX je pouze sada elementárních příkazů, pomocí nichž se vytvářejí komplexnější instrukce vyšší úrovně, které nazýváme makra.

\LaTeX je tedy nadstavba obsahující pro uživatele přijatelnější příkazy. Tento formát sestavil americký odborník Leslie Lamport. Od svého vzniku prochází neustálým vývojem. Rozvojem informačních technologií došlo ke vzniku velkého množství aplikací postavených na tomto formátu. K těm významnějším patří vyšší verze \LaTeX u $\LaTeX 2\epsilon$ čteno „latech tú í“.

Další software pracující s kompilátorem \LaTeX u a \TeX u je $\text{MiK}\TeX$, který vyvíjí Christian Schenk. Takle distribuce je známa svým jednoduchým rozšiřováním balíčky (tzv. packages) volně stažitelnými z internetu. Bakalářská práce je také sázena v tomto softwaru.

Sázecí systém neustále rozšiřuje své možnosti. Cílem mého zkoumání je vytváření obrázků, grafů a schémat v \LaTeX u. Kreslení obrázků v systému \LaTeX se provádí generováním napsaného zdrojového kódu.

I. TEORETICKÁ ČÁST

1 PRVNÍ KROKY V L^AT_EXU

Tvorba dokumentů v L^AT_EXu je jiná než u běžných editorů, protože nepatří mezi WYSIWYG aplikace. WYSIWYG je akronymum z anglické věty „What you see is what you get“, což česky znamená „co vidíš, to dostaneš“. Tento termín označuje způsob editace textu, při níž je zobrazený výstup neustále obnovován a editor, jakožto člověk, vidí jednotlivé změny, které provedl, takřka ihned. WYSIWYG aplikace zaměstnávají procesor neustále. Pro uživatele, kteří jsou zvyklí na tvorbu textových dokumentů v aplikacích pracujících na WYSIWYG systému, jakou je např. distribuce OpenOffice, mohou být první kroky v T_EXu velmi nezvyklé.

V L^AT_EXu je možné zobrazit výstup až po překladu. Pro zjednodušení práce je vhodné používat editor pro tvorbu dokumentů v systému L^AT_EX. V bakalářské práci použitý TeXnicCenter¹⁾, který je zobrazen na obrázku 1, poskytuje přívětivé prostředí. Jeho součástí je builder, který přeloží aktivní dokument a výstup konvertuje do předem zadaných formátů. Nejčastěji se jedná o formát PDF (Portable Document Format – přenosný formát dokumentů) nebo formát PS (PostScript), popřípadě přímo T_EXový formát DVI (DeVice Independent), který obsahuje odkazy na obrázky a je jej možné standardními nástroji převést do formátu PDF. Poté je možné si výstupní soubor prohlédnout a provést úpravy, pokud výsledek nesplňuje požadované vlastnosti. Tento pracovní přístup může nováčkům činit značné problémy. Nicméně trpělivost v začátcích v L^AT_EXu jistě přinese cílený efekt.

„Text, který nám poslouží jako vstup pro zpracování L^AT_EXem, připravíme pomocí *textového editoru* a uložíme do souboru s příponou `.tex`. Tento soubor by měl obsahovat pouze tisknutelné znaky – neměl by zahrnovat skryté znaky nebo příkazy, které nejsou běžně viditelné na obrazovce. Vstupní text však musí obsahovat i příkazy zadávající instrukce L^AT_EXu. Tyto příkazy jsou viditelné ve vstupním textu, ale nezobrazí se v jeho literální formě ve výstupu. Je velmi důležité vědět, jak se příkazy odlišují od textu, který se má vytisknout, a samozřejmě také to, jak fungují.“ (Kopka, 2004, s.15).

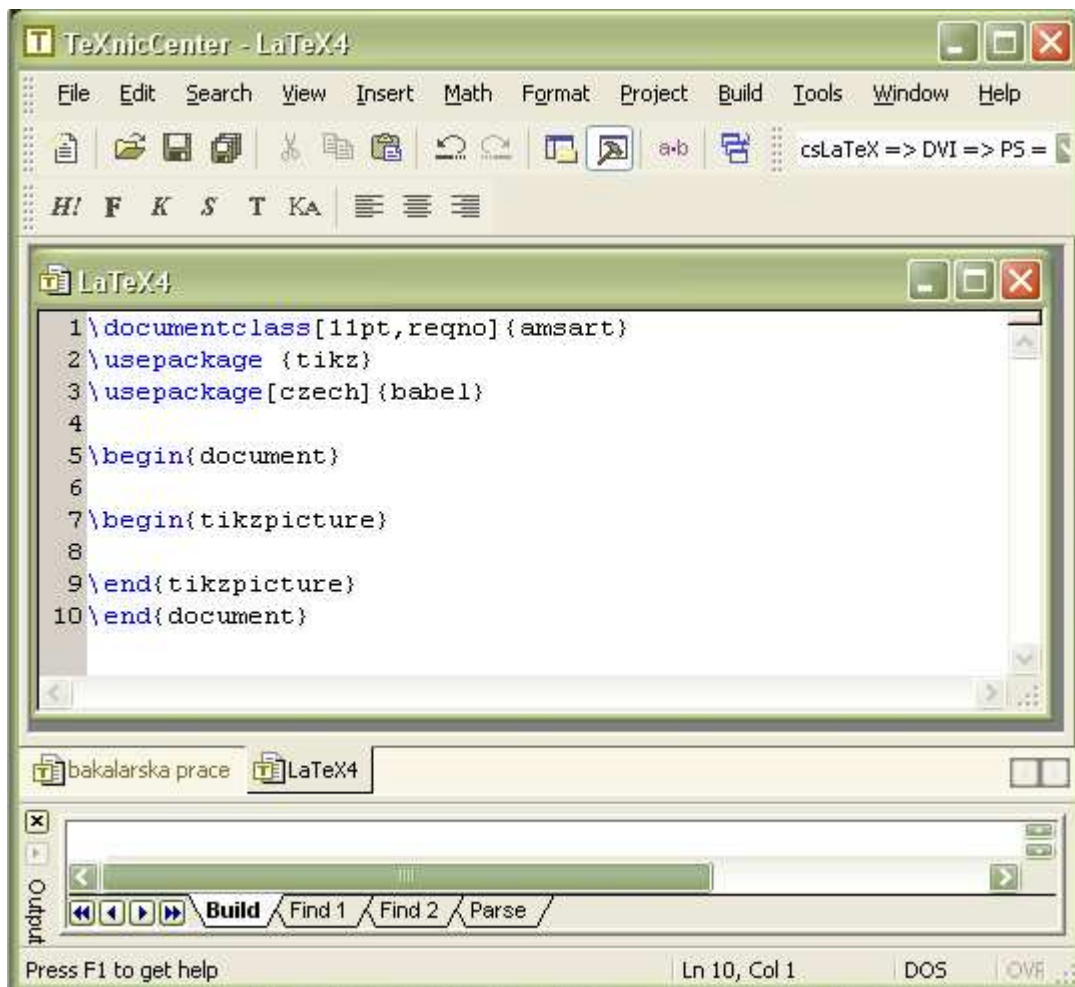
Instrukce představují pro překladač důležité informace o tom, jak se má daný úsek textu vysázet. Příkazy tvoříme dle obecné syntaxe, která vypadá následovně:

$$\backslash \textit{n\acute{a}zev_p\acute{r}\acute{i}kazu} [\textit{voliteln\acute{e}_argumenty}] \{ \textit{povinn\acute{y}_argument} \}$$

Některé příkazy mohou být bez argumentů. Příkladem může být vysazení některého ze speciálních znaků jako jsou #, &, \$, %, -, {, }. Každý z těchto znaků má svůj specifický význam. Příkaz pro procento vypadá například takto: `\%` a symbol % vytiskne v literární podobě.

Pokud se v syntaxi příkazu objevuje povinný argument, musí se vždy zadat. Taková instrukce může vypadat například takto:

¹⁾<http://www.texniccenter.org/>



Obrázek 1. Uživatelské rozhraní Technic Center

`\textbf{tento text bude tučně}` **tento text bude tučně**

Argumentem u této instrukce je text, který chceme vysázet tučně. Jak uvádí Kopka(2004), u instrukce se může vyskytovat i více povinných argumentů, každý z nich musíme vložit do páru složených závorek a také musíme zachovat pořadí těchto argumentů, jak je uvedeno v popisu zadávaného příkazu.

Volitelným argumentem myslíme část příkazu v hranatých závorkách. Tato část je nepovinná a zadává se v případě, když chceme změnit standardní vlastnosti sázeného objektu. Muže se jednat o změnu barvy, velikosti či jiných parametrů. Tyto argumenty se sázejí do jediných hranatých závorek a oddělují se čárkou.

„Každý L^AT_EXový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo“ (Kopka, 2004, s. 11).

Preambulí se myslí nastavení parametrů pro celý dokument. Tato část zdrojového textu musí obsahovat `\documentclass[volitelné nastavení]{volba třídy}`. Tento příkaz se nachází na začátku každého dokumentu v L^AT_EX_{2 ϵ} a určuje o jaký druh dokumentu se bude jednat. Do preambule začleňujeme rozšiřující balíčky, které nám rozšiřují

použitelnou sbírku maker. Příkaz, který přidává do dokumentu balíček vypadá takto:

```
\usepackage{název_balíčku}
```

Vlastní tělo dokumentu ohraničujeme nahoře příkazem `\begin{document}` a ukončujeme jej příkazem `\end{document}`. Mezi tímto ohraničením se nachází zdrojový text s formátujícími příkazy. Tělo dokumentu může obsahovat těla dalších lokálních těl objektů. Těmito objekty myslíme těla obrázků, rovnic, tabulek a dalších. Ohraničení bude vypadat obdobně jako u těla dokumentu, avšak je na místě zaměnit argument ve složených závorkách. Například u rovnic se bude jednat o argument `equation`.

2 NÁSTROJE PRO TVORBU GRAFICKÝCH SCHÉMAT

„Diagramy, fotografie, snímky obrazovky, grafické znázornění experimentálních dat, to vše patří k netextovému materiálu, který je standardní součástí dnešních dokumentů, ať již mají vědeckou povahu či nikoli. Protože se \LaTeX zajímá pouze o textový aspekt těchto materiálů, musí poskytovat nějakou možnost, jak integrovat grafické materiály, které byly připraveny jinými programy“ (Kopka, 2004, s. 147).

Původní \LaTeX byl vybaven jen primitivními nástroji k vytváření grafických schémat. Postupem času, jak stále rostla potřeba integrovat rozmanitější grafický materiál, byly vyvíjeny prostředky k těmto účelům. Do systému se integrují pomocí doplňkových balíčků.

Jak uvádí Rybička (2003), můžeme do textu vkládat obrázky ve dvou zcela rozdílných formách počítačového zobrazení – vektorové a rastrové. Vektorovou formu tvoříme posloupností příkazů pro vykreslení jednotlivých částí zobrazovaného grafického útvaru. Naopak u rastrového obrázku myslíme posloupnost informací o každém jednotlivém pixelu.

Nejen pro práci s obrázky je potřeba znát použitelné délkové jednotky v \LaTeX u. Jednotlivé dimenze tvoříme desetinným číslem, před kterým můžeme zapsat znaménka (+ nebo –). Lze použít u nás běžně známé délkové míry metrické soustavy i méně známé, které se užívaly v minulosti ve spojitosti s tiskem, či jednotky atypické. Jednotkami z metrické soustavy jsou milimetry (užíváme zkratku **mm**) a centimetry (**cm**), anglosaský palec (**in**, $1 \text{ in} = 2,54 \text{ cm}$), anglosaský typografický bod (**pt**, $1 \text{ pt} = 2,54/72.27 = 0,351\ 46 \text{ mm}$), velký bod (**bp** „big point“, $1 \text{ bp} = 2,54/72 = 0,352\ 78 \text{ mm}$) a pica (**pc**, $1 \text{ pc} = 2,54/6 = 4,233\ 6 \text{ mm}$). Mezi historické jednotky, které můžeme použít, řadíme Didôtův bod (**dd**, $1 \text{ dd} = 0,375\ 9 \text{ mm}$), jenž je odvozen z bývalé francouzské stopy. Z Didôtova bodu odvozujeme jednotku cicero (**cc**, $1 \text{ cc} = 12 \text{ dd}$). Specifickými délkovými jednotkami jsou **em** a **ex**. Jednotka **em** je šířka velkého písmene M a **ex** je výška malého písmene x.

2.1 Vkládání obrázků do textu

Umísťování obrázků do textu patří mezi obecné problémy tvoření různých publikací. Snad nejviditelnější komplikace sázení obrázků nastává, pokud chceme vysázet obrázek na konci stránky a zbývající prostor vespod stránky není dostačující pro obrázek. Řešením může být umístění obrázků na další stránku, avšak tím to způsobem dojde k vynechání prostoru.

„Vhodným řešením tohoto problému je to, že se obrázek či tabulka, které se již na aktuálně sázenou stránku nevejdou, prostě na této stránce nevysází a s jejich sazbou se posečká až na další vhodné místo na některé z následujících stránek“ (Oetiker, 1996, s. 28).

V \LaTeX u se pro osazování obrázků užívá plovoucího prostředí (Floating Bodies). Tímto termínem rozumíme prostředí ohraničené následujícími příkazy:

```
\begin{figure}[argumenty umístění prostředí]
  příkazy umísťovaného objektu
\end{figure}
```

Každý objekt, který umístíme do prostředí `figure`, je chápán jako plovoucí objekt. Volitelnými argumenty specifikujeme povolené umístění obrázku. Argumentem může být jedno z písmen `h`, `t`, `b`, `p` a vykřičník. Parametr `h` (here) umísťuje plovoucí objekt přednostně do místa, kde je vysázen ve zdrojovém kódu. Parametr `t` (top) umístí objekt nahoru stránky, na které se nachází nebo na následující stránku. Parametr `b` (bottom) umístí objekt vespod stránky. Parametr `p` (page) umístí objekt na speciální stránku, na které se nachází, jen plovoucí objekty. Tato stránka se nazývá plovoucí stránka (floating page). Parametr vykřičník `!` dává plovoucímu objektu specifické vlastnosti. Vykřičník potlačuje omezení počtu plovoucích objektů na jedné stránce a současně explicitní nastavení prostoru textové stránky, který musí zabírat text. Tento parametr musí být užit alespoň s jedním ze zmíněných písmenných argumentů.

Před vkládáním obrázků do textu je důležité pro rozšíření možností vložení zapsat do preambule balík `graphicx` příkazem `\usepackage{graphicx}`. Pomocných balíků pro vkládání grafických objektů můžeme použít více. Obrázky do textu vkládáme následující instrukcí, která má dvě možné syntaxe:

```
\includegraphics[llx,lly][urx,ury]{název_souboru}
\includegraphics[vlastnosti objektu]{název_souboru}
```

Můžeme tedy vložit obrázek do ohraničujícího rámečku, který je určen pomocí dvou souřadnic. První souřadnice `llx,lly` nám určuje dolní levý roh a druhá souřadnice `urx,ury` pravý horní roh. K souřadnicím můžeme připsat jednotky, předdefinovanou jednotkou je `bp` (big points). Souřadnice jsou nepovinným argumentem, můžeme je s úspěchem užit jen v případě, vkládáme-li objekt v zapouzdřeném PostScriptovém

formátu. Jinou možností vložení nám umožňuje již zmíněný balík `graphicx`. Zadáním klíčových slov s přiřazením požadované hodnoty v hranatých závorkách, přiřadíme objektu nové vlastnosti. Těmito operacemi můžeme obrázek např. rotovat o stanovený úhel, změnit měřítko, obrázek ořezat nebo s ním provést i několik dalších operací.

Samotné zadání objektu se provádí v povinném argumentu. \LaTeX neumí rozeznat všechny grafické formáty. Standardně pracuje se zapouzdřenými PostScriptovými soubory, jimiž jsou `.eps` či `.ps`. Pokud chceme vložit bitmapu, je vhodné v jakémkoliv grafickém programu přeložit objekt ve formátu `.eps`. Formát `.eps` (Encapsulated PostScript) se používá pro rastrové, vektorové i kombinované obrázky. Obsahuje jen obrázek ohraničený obdélníkem pro tisknutelnou oblast.

Obrázky, které chceme vkládat, musí být v adresáři, kde máme uložen projekt nebo můžeme definovat nový adresář pomocí instrukce `\graphicspath{seznam adresářů}`, ve kterém bude \LaTeX při překladu hledat. Při přenášení `.tex` souboru na jiný PC je nutné zabezpečit aby překladač našel tyto soubory.

Vkládání obrázků má svá úskalí. U bitmapového obrázku dochází s jeho úpravou pro ideální vložení do textu k deformacím. Úprava velikosti nebo rotace objektu vede k rozostření čar, písmen či barev. Například při zmenšení grafického objektu, který obsahuje popisná data, o padesát procent, dojde i ke zmenšení písmen a mohou se stát takřka nečitelná.

2.2 Grafické prostředí picture

„Prostředí je určeno pro kreslení vektorových obrázků složených z přímých čar, šipek, kružnic, oválů a libovolných textů. Uvnitř tohoto prostředí se mohou objevit pouze příkazy pro vložení obrazového objektu (`\put` a `\multiput`) a příkazy pro nastavení velikosti a typu písma a délek“ (Rybička, 2003, s. 117.).

Prostředí `picture` vypadá následovně:

```
\begin{picture}(hor. rozměr, ver. rozměr)(souř. pos. x, souř. pos. y)
    příkazy pro tvorbu objektu
\end{picture}
```

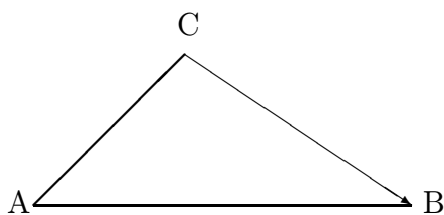
V prostředí `picture` vytváříme tzv. box. Jeho velikost je stanovena horizontálním a vertikálním rozměrem, které zapisujeme do prvních kulatých závorek. Do druhých kulatých závorek můžeme zapsat posun levého dolního rohu boxu po ose x a po ose y . Tímto posunem máme zabezpečeno umístění objektu na libovolné námi stanovené místo na stránce. Oba tyto parametry jsou zadávány bez jednotek. Jednotku nastavíme příkazem:

```
\setlength{\unitlength}{5mm}
```

Při nastavení jednotky na 5 mm bude čára o délce 4 jednotky dlouhá 2 cm, o délce 8 jednotek dlouhá 4 cm. Bez stanovení velikosti jednotky bude užito implicitní hodnoty 1 pt. Prostředí `picture` s nastavenou jednotkou a s příkazy pro vykreslení trojúhelníku pomocí čar bude vypadat takto:

```
\begin{picture}(50,40)\thicklines % tloušťka čáry
    \setlength{\unitlength}{5mm}
    \put(10,0){\line(-1,0){10}} % nakreslí čáru
    \put(0,0){\line(1,1){4}}
    \thinlines % tloušťka čáry
    \put(4,4){\vector(3,-2){6}} % nakreslí vektor
    \put(-0.7,-0.2){A} % umístí písmeno na zadanou souřadnici
    \put(10.3,-0.2){B}
    \put(3.8,4.5){C}
\end{picture}
```

Tento kód nám po překladu vykreslí trojúhelník ABC, jak se můžeme přesvědčit na Obrázku 1. Pro kreslení čar jsem zvolil předdefinovanou tlustou čáru příkazem `\thicklines`, která má šířku 8 pt, pro vektor pak čáru příkazem `\thinlines` o šířce 4 pt. Instrukcí `\linethickness{tloušťka včetně jednotky}` si můžeme definovat svou vlastní tloušťku čáry.

Obrázek 1. Trojúhelník v prostředí `picture`

Čáry kreslíme instrukcí `\line`, která má několik parametrů. Tato instrukce bude ve složených závorkách, protože je součástí instrukce `\put`, za kterou je v kulatých závorkách umístěn referenční bod (tzv. origin). Origin nám určuje, z kterého místa bude čára vedena. První parametr instrukce `\line` v kulatých závorkách určuje směr vykreslení, které je určeno dvojicí čísel, která jsou z intervalu celých čísel od -6 do 6 . Tato čísla nesmí mít společného dělitele většího než 1 . Druhý parametr je ve složených závorkách a určuje o jakou hodnotu bude změněna jeho x -ová souřadnice, vzhledem k referenčnímu bodu.

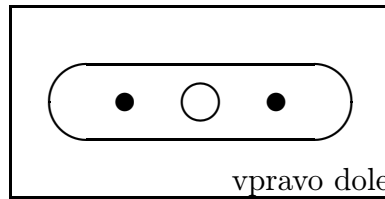
Instrukce pro vektor vychází ze stejné filozofie, navíc je opatřen šipkou, která směřuje od referenčního bodu. Na obrázku 1 vektor směřuje z bodu C do bodu B . Ještě jednodušeji můžeme umísťovat do obrázku text. Z kódu vyplývá, že text zadaný ve složených závorkách se zakreslí na místo, určené referenčním bodem.

Prostředí `picture` pracuje s několika předdefinovanými tvary, které můžeme libovolně kombinovat. Předdefinovanými tvary myslíme kružnici, ovál, rámeček, bezierovu křivku a další.

Instrukci pro kružnici můžeme použít i pro vykreslení kruhu pomocí tzv. hvězdičkové syntaxe. Kružnici vykreslíme příkazem `\circle{průměr}` a kruh příkazem `\circle*{průměr}`. Ovál je tvořen dvěma půlkružnicemi, které jsou propojeny čarami. V instrukci je zadán výškou a šířkou, má volitelný argument, s nímž můžeme stanovit, kterou část křivky oválu chceme vykreslit. Ovál vykreslíme příkazem `\oval{šířka, výška}[t-nahoru, b-dolu, l-vlevo, r-vpravo]`. V následujícím kódu jsou tyto příkazy použity.

```
\begin{picture}(50,40)\thicklines
  \setlength{\unitlength}{5mm}
  \put(5,2.5){\oval(8,2)}
  \put(5,2.5){\circle{1}}
  \put(3,2.5){\circle*{0.5}}
  \put(7,2.5){\circle*{0.5}}
```

```
\put(0,0){\framebox(10,5)[rb]{vpravo dole}}  
\end{picture}
```



Obrázek 2. Použití vybraných příkazů

Obrázek 2 je utvořen z oválu, kružnice, dvou kruhů, textu a ohraničením, které bylo vytvořeno příkazem `frame`. `Frame` je zadán šířkou a výškou, podobně jako u oválu. Dvnitř je umístěn text vpravo dolů dle parametru `rb`. Dalšími instrukcemi v prostředí `picture` jsou:

```
\dashbox{délka čárky}(šířka,výška)[pozice textu]{text}  
\frame{text k orámování}  
\makebox(šířka, výška)[pozice textu]{text}
```

Prostředí `picture` poskytuje několik nástrojů pro tvorbu jednoduché grafiky, ale není vhodné pro tvorbu složitějších grafických konstrukcí, protože nedisponuje tak rozsáhlou sadou příkazů.

2.3 Grafické prostředí PSTricks

PSTricks(PostScript Tricks) je soubor balíčků Post-Scriptových maker vytvořených na \TeX ovém základě. PSTricks poskytuje širokou paletu nástrojů pro kreslení vektorové grafiky přímo v \TeX ovém kódu. Tento balík disponuje makry pro barvy, kreslení čar a křivek, rotaci objektů, pro tvorbu stromových struktur a barevné výplně grafických útvarů.

Prostředí PSTricks vytvořil Timothy Van Zandt a k tomu i velmi obsáhlý manuál. Tento manuál poskytne uživateli seznámení se základními funkcemi. V této době se na vývoji PSTricks podílejí Denis Girou, Sebastian Rahtz a Herbert Voss.

Můžeme kreslit přímo v textu nebo v PSTrickovém prostředí stejně jako u prostředí `picture`. Přímou v textu lze snadno nakreslit jednoduché grafické konstrukce. Složitější grafická schémata je mnohem výhodnější vysázet do prostředí ohraničeného příkazy `\begin{pspicture}` a `\end{pspicture}`.

Prostředí PSTricks disponuje několika balíky, které obsahují další instrukce pro kreslení. Každý použitý balík musí být zapsán v preambuli v již zmiňované formuli `usepackage`. Názvy balíčků vždy začínají `pst-`. Za pomlčkou uvádíme název balíčků, které chceme přidat. Můžeme použít např. tyto balíčky:

Balík (package)	Použití
<code>pst-node</code>	pro kreslení s uzly
<code>pst-plot</code>	pro kreslení dvojrozměrných grafů
<code>pst-3dplot</code>	pro kreslení trojrozměrných grafů
<code>pst-func</code>	pro vykreslování matematických funkcí
<code>pst-fill</code>	pro vyplňování a ohraničování ploch a znaků
<code>pst-abspos</code>	pro kreslení v absolutních souřadnicích
<code>pst-bezier</code>	pro kreslení bezierových křivek
<code>pst-text</code>	pro manipulaci s textem a znaky
<code>pst-grad</code>	pro vyplňování ploch s barevnými přechody
<code>pst-circ</code>	pro kreslení elektronických součástek

Makra v PSTricks užívají závorek podobně jako \LaTeX u k různým účelům. Složené závorky `{ }` pro povinné argumenty, hranaté závorky `[]` pro volitelné argumenty, kulaté závorky `()` pro souřadnice (*x-ová souřadnice*, *y-ová souřadnice*). Symbolu „`=`“ se užívá pro přiřazení hodnoty.

Implicitně nastavenou jednotkou jsou centimetry. Příkazem `psset` můžeme kdykoliv změnit atributy jednotlivých objektů a jejich vlastnosti. Můžeme změnit jednotku, nastavení tloušťky čar, barvy a další vlastnosti prostředí.

V PSTricks je předdefinováno šest základních barev `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` ze soustav RGB-barevného modelu a modelu CMY založeného na subtraktivním míchání barev a pět barev ve stupních šedi `black`, `darkgray`, `gray`, `lightgray`, a `white`. Můžeme si dle potřeby definovat další barvy. Ve stupních šedi bude příkaz vypadat následovně:

```
\newgray{color}{cislo}
```

Color je název námi vytvořené barvy. *Cislo* nabývá hodnot od 0 do 1. Hodnota 0 je pro černou barvu a 1 pro bílou. Šedesátiprocentní černou barvu můžeme definovat takto:

```
\newgray{mojeseda}{.4}
```

Podobné tvoření barev je možné v RGB soustavě barev i v soustavě CMY. Díky tomu je možné používat jakoukoliv barvu z celého barevného spektra.

PSTricks oplývá širokým uplatněním na poli matematiky, samotné začlenění primitivních matematických funkcí pro tvorbu grafů umožňuje jednoduše vykreslit např. goniometrické funkce. Prostředí disponuje přídatnými balíky, které umožňují sázet v L^AT_EXu složitá grafická schémata použitelná v technických a přírodovědných oborech jako je fyzika, chemie, automatizace, elektronika a další. Uživatelé se mohou podílet na vývoji nových balíků a autoři jsou ochotni jimi vytvořené komponenty začlenit do systému. Přehled toho co dokáže rozhraní PSTricks je na oficiálních stránkách v záložce *Examples* <http://tug.org/PSTricks/main.cgi/>.

PSTricks je postaven na širokém podpůrném základu. Tento balík je pravděpodobně nejsilnější nadstavbový balík pro tvorbu grafických schémat. Vzhledem ke složité syntaxi může činit začínajícím uživatelům značné problémy se s balíkem PSTricks naučit pracovat.

2.4 Grafické prostředí TikZ & PGF

Grafický balík TikZ & PGF je nástroj k vytváření vysoce kvalitních ilustrací a grafiky v multiplatformním TeXovém prostředí. Jedná se o vektorové ilustrace, proto při manipulaci s nimi nedochází k deformacím a ke snižování kvality vykreslení. Balík TikZ je komplexním nástrojem s širokým uplatněním. Obsahuje několik základních instrukcí, které mají nepřeberné množství použití díky svým podinstrukcím, povinným argumentům a volitelným parametrům.

Balík Tikz vytvořil německý univerzitní profesor Till Tantau. TikZ je akronym z německého „TikZ ist kein Zeichenprogramm“. To se dá volně přeložit TikZ není žádný kreslicí program. Ačkoliv název tvrdí, že se nejedná o grafický program, výsledná grafika někdy může působit, jako by byla vytvořena v profesionálním grafickém programu pracujícím s vektory. PGF je akronymem „portable graphics format“. PGF formát je přenositelný grafický formát. TikZ je pro PGF uživatelskou sázecí syntaktickou vrstvou.

Makra v TikZ jsou propracovanější instrukce a jsou na vyšší úrovni než jsou příkazy v TeXu, L^ATeXu či PSTricks. V tomto prostředí se uživateli pracuje mnohem příjemněji. Instrukce jsou stavěné podobně jako v jiných programovacích jazycích. S nadsázkou můžeme tvrdit, že díky středníkovým příkazům může výsledný kód působit na první pohled jako kód v programovacím jazyku C. L^ATeXová syntaxe s použitím zpětného lomítka před instrukčními hesly však zůstává.

Samotné užívání tohoto balíku maker je velmi podobné používání balíku PSTricks. TikZ & PGF tvoří výstup buď ve formátu .ps nebo .pdf. Můžeme si k základnímu balíku maker TikZ & PGF naimportovat velké množství vytvořených balíčků od různých autorů, někdy se může stát, že nebudou balíky mezi sebou plně spolupracovat.

Po krátkodobém zkoušení psaní kódů pro grafická schémata se uživatel snadno seznámí s mnoha možnostmi kreslení, které poskytuje balík maker TikZ & PGF. Balík umožňuje uživateli nakreslit i 3D grafiku.

3 SROVNÁNÍ GRAFICKÝCH PROSTŘEDÍ

V \LaTeX u existuje několik grafických nástrojů, které jsou přímo součástí jeho distribuce, popřípadě distribuce MikTeX u. Mimo to je k dispozici široká škála grafických doplňkových balíčků, které můžeme do tohoto programu naimportovat. V teoretické části bakalářské práce bylo zkoumáno použití čtyř ze všech možností, jak dostat do textového souboru grafická schémata. Jednalo se o vložení grafického souboru a grafická prostředí `picture`, `PSTricks` a `TikZ & PGF`.

Každé prostředí má své výhody pro specifické úkony, které můžeme chtít provést. Nejjednodušším způsobem, jak dostat do textu grafiku je rozhodně vložení obrázku na cílené místo v textu. Vložení obrázku do textu není pro \LaTeX žádný problém. Avšak před vložением musíme provést převod obrázku na požadovaný formát. Komplikací vkládání je vždy velikost obrázku. Pokud jej nemáme v požadované velikosti, dojde k deformacím při jeho úpravě. Tím obrázek ztrácí na kvalitě. Ukázkou vloženého bitmapového obrázku vidíme na obrázku 1. Pro obrázky s textem je zmenšení či zvětšení poznatelná změna na první pohled, text se také deformuje a stává se nečitelným. Pro matematické a technické účely je vkládání obrázků v našem případě nevyhovující.

Prostředí `picture` je uživatelsky přívětivé pro tvorbu primitivní grafiky, která se skládá ze základních grafických útvarů, jimiž máme na mysli čáry, kružnice, obdélníky a podobné. Vytváříme v tomto prostředí vektorový obrázek, který přechází na bitmapu až po překladu a ve výsledku působí jako perokresba. Můžeme používat text, aniž by došlo k jeho k jakýmkoliv deformacím. Použití tohoto prostředí je vhodné pro tvorbu jednoduchých náčrtů, pro složitější grafická schémata se stává velmi nepraktické.

Prostředí `PSTricks` je jednoznačně nejpropracovanější grafická vrstva pro \LaTeX , která prošla a nadále prochází vývojem, na kterém se podílejí i běžní uživatelé \LaTeX u. Dá se konstatovat, že se v tomto balíku maker najdou takové nástroje, které nám dopomohou vytvořit různé obrázky. Kód obrázku se stejně jako u prostředí `picture` tvoří přímo v textu. Často nám vynaložená námaha a čas nemusí korespondovat s výsledkem. Zlatým pravidlem jistě je, pokud už před námi někdo podobného tvořil, je více než vhodné použít části těchto kódů a usnadnit si tak práci. Kódy s různým grafickým snažením uživatelů jsou dohledatelné a volně k dispozici na oficiálních internetových stránkách. `PSTricks` obsahuje i balíky pro tvorbu 3D grafiky. Výsledná grafická schémata jsou vektorově vykreslená, proto nedochází při jejich upravách k deformacím. K prostředí `PSTricks` vytvořil autor balíku obsáhlý manuál pro pokročilejší tvorbu grafiky. `PSTricks` hodnotíme jako velmi propracovaný balík maker, který přivítá začínajícího tvůrce grafiky velkou složitostí a časovou náročností.

Posledním členem ze zkoumaných nástrojů je `TikZ & PGF`. Je jedním z nejmladších součástí \LaTeX u. Přívětivější psaní zdrojového kódu zvládne každý začínající uživatel \LaTeX u. Postupnou cestou zkoušení kreslení nových grafických prvků se uživatel rychle

zdokonalí v samotné tvorbě a od počátku své práce s programem si může pomocí poskytnutých nástrojů vyzkoušet nakreslit i něco složitějšího. TikZ & PGF je silným nástrojem ve tvoření vektorové grafiky, ačkoliv název, který mu dal jeho autor, tomu neodpovídá. Podobně jako PSTricks disponuje makry, která nás neomezují v kresbě a můžeme se pokoušet s kvalitním výsledkem nakreslit různá grafická schémata.

II. ANALYTICKÁ ČÁST

4 GRAFICKÁ SCHÉMATA

4.1 Úvod do tvorby grafických schémat

V analytické části se zaměřím přímo na tvorbu grafických schémat. K jednotlivým schématům je na přiloženém CD ve složce *graficka schemata* funkční zdrojový kód. U prvního zadaného obrázku je podrobně popsán postup tvorby s vysvětlením toho, co jednotlivé instrukce provádějí. Objasnění příkazů bude vodítkem pro kreslení grafických schémat.

Pro tvorbu grafických schémat byl zvolen balík maker **TikZ & PGF** z několika důvodů. Jedním z nich je určitě přehledný manuál, který zpracoval německý autor Till Tantau. Manuál poskytuje podrobné informace o tvorbě různých grafických útvarů. Dalším důvodem je má rychlá adaptace na psaní zdrojového kódu tohoto balíku. Balík poskytuje dostatek maker, kterým můžeme přiřadit pomocí volitelných parametrů spoustu zajímavých vlastností. A v neposlední řadě je předností **TikZ & PGF**, že se jedná o jeden obsáhlý balík s mocným souborem nástrojů, do kterého lze ještě přidávat pomocné balíky a knihovny.

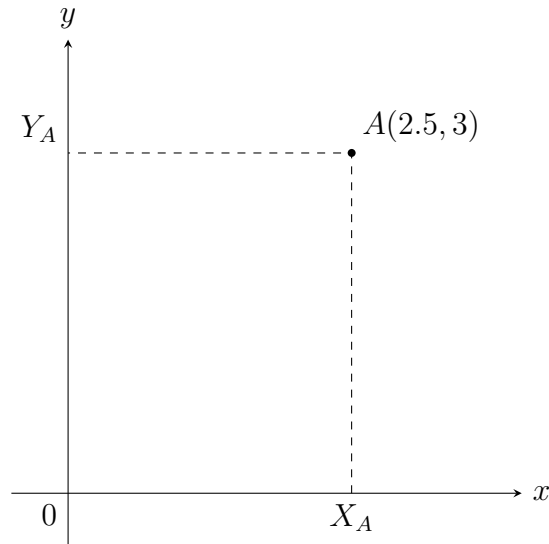
Základní předdefinované prostředí s preambulí, do kterého budeme vkládat kódy obrázků z analytické části, bude mít tento tvar a kód bude sázen mezi instrukce `\begin{tikzpicture}` a `\end{tikzpicture}`.

```
\documentclass[11pt,reqno]%definovana trida
%preambule
\usepackage{tikz}%balík maker TikZ
\usepackage[czech]{babel}
\begin{document}%začátek dokumentu
\begin{tikzpicture}
%kód obrázku
\end{tikzpicture}
\end{document}%konec dokumentu
```

Všechny instrukce v tomto prostředí musí být ukončeny středníkem. Jeden příkaz může být i přes několik řádků. Na jednom řádku se může také vyskytovat více příkazů ukončených středníkem. Základní jednotkou v prostředí jsou centimetry, pokud chceme u některého příkazu použít jednotku jinou, napíšeme označení jednotky za číslo určující požadovaný rozměr.

Na začátku samotné tvorby grafických schémat v **TikZ & PGF** by si měl autor rozhodnout, zda bude tvořit ve dvojrozměrném nebo trojrozměrném prostoru. Před psaním kódu je ideální předkreslení složitějšího obrázku na papír nebo v jiném grafickém programu. Při tvorbě 3D obrázku se bez prostorové orientace neobejdete.

Prvním krokem by mělo být vytvoření souřadného systému. V TikZ & PGF jsou souřadnice definovány kartézským systémem souřadnic ve 2D i ve 3D. Bod zavedený v rovinné kartézské soustavě souřadnic vidíme na obrázku 1, bod zavedený v prostorové kartézské soustavě souřadnic na obrázku 2.














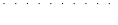
Obrázek 1. Bod v rovinné kartézské soustavě souřadnic

Zdrojový kód pro obrázek 1

```
\begin{tikzpicture}[scale=1.5]
\draw[thin] [->] (-0.5,0) -- (4,0) node[right]{$x$};
\draw[thin] [->] (0,-0.5) -- (0,4) node[above]{$y$};
\node[below left](0,0){$0$};
\draw[dashed] (2.5,0) coordinate (xa) node[below]{$X_A$}--(2.5,3)
coordinate (a)node [above right]{$A$}--(0,3) coordinate
(ya) node[above left]{$Y_A$};
\fill (a) circle (1pt);
\end{tikzpicture}
```

Prostředí `tikzpicture` může mít několik volitelných argumentů, jedním z nich je `scale`. `Scale` je měřítko obrázku, je výhodné jej uvést vždy, i když obrázek chceme v poměru 1 : 1. Třemi nejdůležitějšími instrukcemi TikZu jsou `\draw`, `\fill` a `\node`. Pomocí příkazu `draw` nakreslíme libovolné konstrukce, je možné pomocí něj i vybarvovat a má mnoho funkcí, několik z nich použijeme v následných obrázcích. Příkaz `fill` se soustředí především na vyplňování ohraničených oblastí barvou, podobnou instrukcí je `shade`. Existují i složené instrukce `filldraw` a `shadedraw`, které určenou oblast současně ohraničují čarou a barevně vyplňují, případně stínují. Instrukce `node` nám poskytuje přesné určení bodu, kterému můžeme přiřadit popis.

Čáry kreslíme jako spojnice dvou a více bodů. Plná čára z bodu (0,0) do bodu (1,1) bude nakreslena příkazem `\draw (0,0) -- (1,1)`. Bod, o kterém víme, že se na něj budeme v budoucích příkazech odkazovat, si můžeme pracovníčně označit příkazem `coordinate`. Takto máme označen např. bod *A* na souřadnicích (2.5, 3), za kterým je zmiňovaný příkaz s označením (a). Při následujících použitích tohoto bodu můžeme místo souřadnic (2.5,3) psát (a). Na našem obrázku máme na koncích os přidanou šipku s jednoduchým argumentem `[->]`, v dalším textu užíváme i argument `[-stealth]` pro jiný tvar šipky. Čarám můžeme přidat různé vlastnosti, popřípadě si definovat své vlastní. V TikZ je předdefinováno několik typů čar, které píšeme do hranatých závorek:

Typ čáry	Volitelné argumenty čar	
	<code>thin</code>	tenká čára (základní)
	<code>very thin</code>	velmi tenká čára
	<code>ultra thin</code>	nejvíce tenká čára
	<code>thick</code>	tlustá čára
	<code>very thick</code>	velmi tlustá čára
	<code>ultra thick</code>	nejvíce tlustá čára
	<code>dashed</code>	čárkovaná čára
	<code>loosely dashed</code>	řídce čárkovaná čára
	<code>densely dashed</code>	hustě čárkovaná čára
	<code>dotted</code>	tečkovaná čára
	<code>loosely dotted</code>	řídce tečkovaná čára
	<code>densely dotted</code>	hustě tečkovaná čára

Každému bodu můžeme příkazem `node` přiřadit popisek. Pro počátek systému souřadnic s označením 0 vypadá příkaz takto `\node[above left](0,0,0){0};`. V hranatých závorkách specifikujeme umístění popisku, můžeme použít např. `above` pro umístění nad bod, `below` pro umístění pod bod, `left` pro umístění vlevo od bodu a `right` pro umístění vpravo od bodu. Samotný popisek umísťujeme mezi znaky dolaru ve složených závorkách `{$popisek$}`. Syntaxe příkazu je jiná, je-li `node` součástí jiného příkazu například `draw`. V tomto případě se píše `node` za bod v kulatých závorkách a hned za tím popisek se stejnou syntaxí (viz obrázek 1).

V TikZ jsou definovány základní tvary, jimiž jsou např. kružnice, elipsa, čtverec a další, které jsou součástí příkazu `draw` či `fill`. U našeho obrázku máme použití kruhu, který je vybarven implicitní barvou, která je černá. Kružnice má jednoduchou syntaxi skládající se z bodu určujícího střed kružnice, klíčového slova `circle` a poloměr v kulatých závorkách.

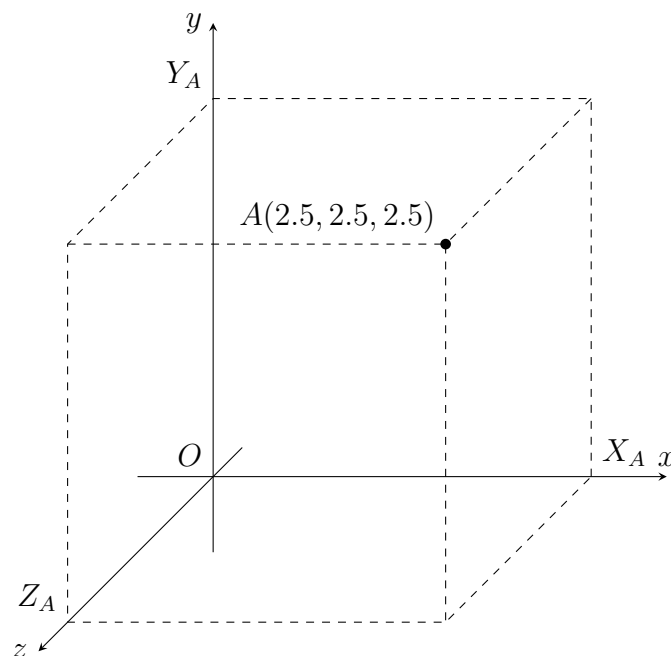
Zdrojový kód pro obrázek 2

```
\begin{tikzpicture} [scale=2]
\draw[thin] [->] (-0.5,0,0)--(3,0,0)node[above]{$x$};
```

```

\draw[thin] [->] (0,-0.5,0)--(0,3,0)node[left]{$y$};
\draw[thin] [->] (0,0,-0.5)--(0,0,3)node[left]{$z$};
\node[above left](0,0,0){$0$};
\draw[dashed] (2.5,0,0) coordinate (xa)node[above right]{$X_A$}
--(2.5,2.5,0)coordinate (xya)--(0,2.5,0) coordinate (ya)
node[above left]{$Y_A$}--(0,2.5,2.5)coordinate(yza)--(0,0,2.5)coordinate
(ya)node[above left]{$Z_A$}--(2.5,0,2.5)coordinate(xza)--cycle;
\draw[dashed] (xya)--(2.5,2.5,2.5)coordinate(a)node[above]{$A$}--(xza);
\draw[dashed] (yza)--(a);
\fill (a) circle (1pt);
\end{tikzpicture}

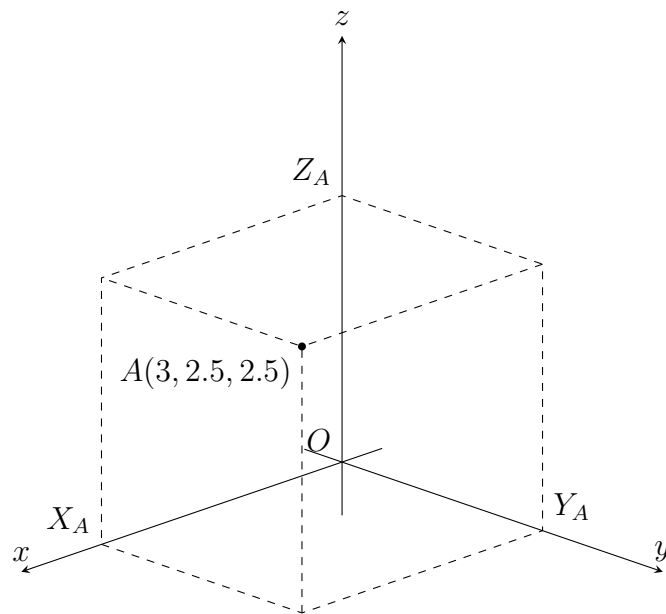
```



Obrázek 2. Bod v prostorové kartézské soustavě souřadnic

Pokud chceme kreslit ve 3D, musíme zadávat 3 souřadnice. Systém TikZ automaticky přepočítává třetí rozměr, pomocí něj dostává obrázek prostorový charakter. Při kreslení čar můžeme užít příkazu `cycle`, který způsobí spojení čarou poslední zadané souřadnice s první zadanou.

Pro kreslení ve 3D můžeme použít i specifickou soustavu souřadnic. Musíme si však do systému \LaTeX přidat balíček `tikz--3dplot` a zapsat jej do preambule příkazem `\usepackage{tikz--3dplot}`. Soustavu v tomto případě určíme pomocí dvou úhlů, jak lze vidět na následujícím obrázku.



Obrázek 3. Bod v prostorové soustavě souřadnic 3Dplot

Zdrojový kód pro obrázek 3.

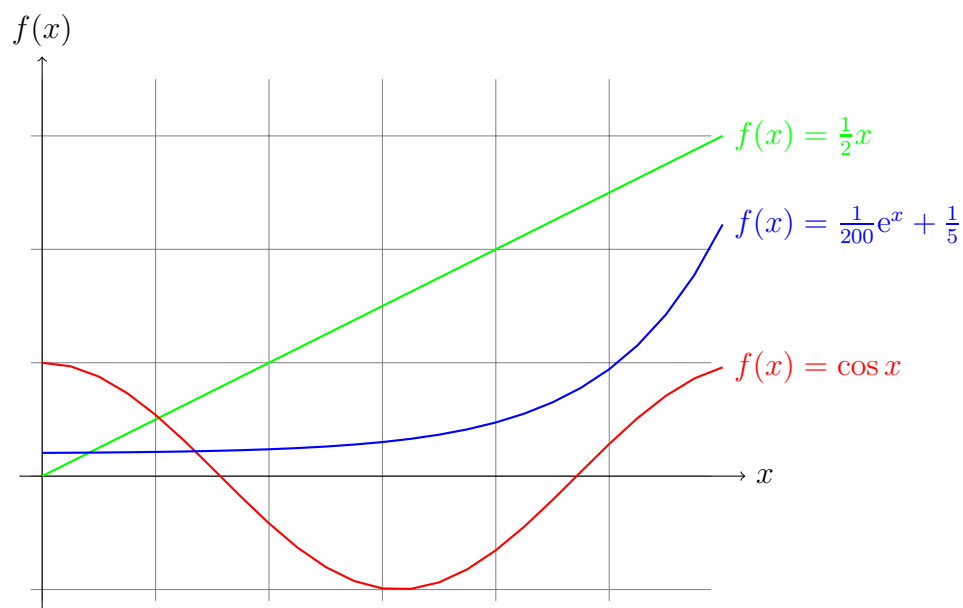
```
\tdplotsetmaincoords{70}{135}
\begin{tikzpicture}[scale=1.5,tdplot_main_coords]
\draw[thin][->] (-0.5,0,0) -- (4,0,0) node[above]{$x$};
\draw[thin][->] (0,-0.5,0) -- (0,4,0) node[above]{$y$};
\draw[thin][->] (0,0,-0.5) -- (0,0,4) node[above]{$z$};
\node[above left](0,0,0){$0$};
\draw[dashed] (3,0,0) coordinate (xa)node[above left]{$X_A$}--(3,2.5,0)
coordinate (xya)--(0,2.5,0) coordinate (ya)node[above right]{$Y_A$}--
(0,2.5,2.5)coordinate(yza)--(0,0,2.5)coordinate (ya)node[above left]
{$Z_A$}--(3,0,2.5)coordinate(xza)--cycle;
\draw[dashed] (xya)--(3,2.5,2.5)coordinate(a)node[above]{$A$}--(xza);
\draw[dashed] (yza)--(a);
\fill (a) circle (1pt);
\end{tikzpicture}
```

Příkaz `\tdplotsetmaincoords{rotace kolem osy x o daný úhel}{rotace kolem osy z o daný úhel}` nám provede rozložení os.

V prostředí je několik možností, jak lze definovat barvy ke kreslení. TikZ má předdefinovány základní barvy. Implicitně je nastavena černá barva. Dalšími barvami jsou `white` – bílá, `red` – červená, `green` – zelená, `blue` – modrá, `cyan` – azurová, `magenta` – purpurová, `yellow` – žlutá, `orange` – oranžová, `brown` – hnědá, `pink` – růžová, `gray` – šedá a další. Základní barvy si můžeme libovolně namíchat (předdefinovat vlastní)

na začátku prostředí `tikzpicture` a pak je libovolně používat stejně jako základní barvy. Předdefinování provedeme příkazem `\definecolor {moje barva}{rgb}{poměr červené barvy, poměr zelené barvy, poměr modré barvy}`. Poměr se udává v rozmezí od 0 do 1. Míchání barev můžeme rovněž provést i přímo v místě použití „vykřičníkovou“ metodou. Touto metodou můžeme poměrově oslabit základní barvu nebo ji oslabit a přidat další barvy. Oslabená zelená barva na padesát procent bude zapsána `green!50`. Oslabená zelená s přimícháním červené barvy bude zapsána `green!50!red`.

Prostředí `TikZ & PGF` nám umožňuje vykreslit 2D grafy základních matematických funkcí. `PGF` obsahuje matematické jádro, které nám umožňuje vložit matematický výraz a `TikZ` potom může vygenerovat požadované souřadnice automaticky. `TikZ & PGF` poskytuje více způsobů jak vytvořit grafy matematických funkcí. Nejjednodušší z nich je zadáním předpisu matematické funkce. Jinou možností může být tabelování vypočítaných hodnot. Pro zadání předpisu matematické funkce můžeme použít několik předdefinovaných matematických funkcí např. `sin`, `cos`, `sqrt`, `exp` a další. Na následujícím obrázku jsou vykresleny tři grafy matematických výrazů.



Obrázek 4. Ukázka 2D grafů v `TikZ & PGF`.

Zdrojový kód pro obrázek 4

```
\begin{tikzpicture}[domain=0:6, scale=1.5]
\draw[very thin,color=gray] (-0.1,-1.1) grid (5.9,3.5);
\draw[->] (-0.2,0) -- (6.2,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,3.7) node[above] {$f(x)$};
\draw[color=green, thick] plot (\x,{0.5*(\x)}) node[right]
{$f(x) = \frac{1}{2} x$};
\draw[color=red, thick] plot (\x,{cos(\x r)}) node[right]
{$f(x) = \cos x$};
\draw[color=blue, thick] plot (\x,{0.005*exp(\x)+0.2}) node[right]
```

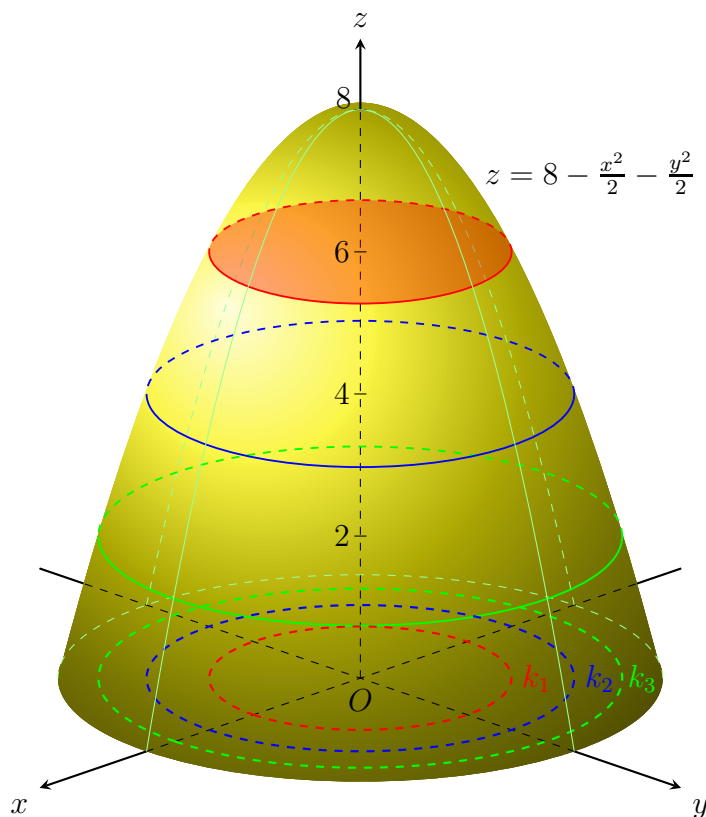
```
{f(x) = \frac{1}{200} \mathrm{e}^{\{x\}} + \frac{1}{5}};  
\end{tikzpicture}
```

Z kódu obrázku 4 vidíme, jaká je syntaxe příkazu pro vykreslení matematických funkcí. Obecná skladba příkazu pro vykreslení grafu vypadá následovně:

```
\draw[parametry čáry] plot(\proměnná, {funkce s proměnnou});
```

Proměnnou volíme nejčastěji jako x . K příkazu můžeme připojit popis podinstrukcí `node`. Důležitým parametrem je interval, na kterém bude funkce vykreslena. Interval určujeme ve volitelném argumentu prostředí. Interval nám určuje parametrický příkaz `domain`, kterému přiřadíme dolní a horní mez oddělenou dvojtečkou. Např. `domain=0:6` nám zaručí, že funkce zadané v prostředí budou vykresleny v intervalu od 0 do 6.

Vykreslování 2D grafů v TikZ & PGF není natolik obtížné, proto byla zadaná schémata volena ve 3D.

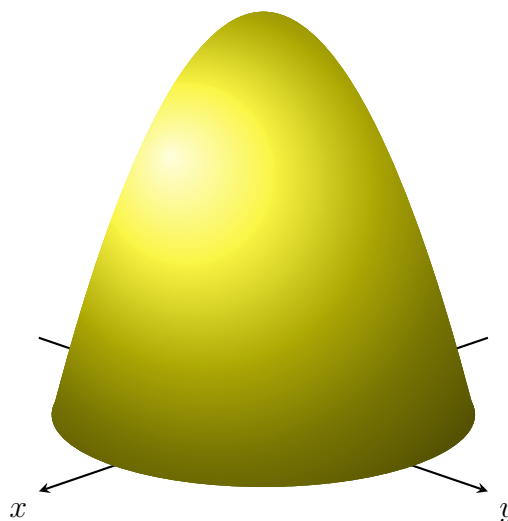
4.2 Vrstevnice funkce $z = 8 - \frac{x^2}{2} - \frac{y^2}{2}$ Obrázek 5. Vrstevnice funkce $z = 8 - \frac{x^2}{2} - \frac{y^2}{2}$

Zdrojový kód grafu je rozdělen do pěti fází. Každá fáze je opatřena obrázkem, který zobrazuje současný stav. Kód je podrobně popsán.

Popis zdrojového kódu v jednotlivých fázích

- `\tdplotsetmaincoords{70}{135}`
 - definice souřadnicového systému - v základní orientaci $\{0\}\{0\}$ osa x směřuje doprava, osa y nahoru a osa z „mimo stránku“, zde $\{70\}$ udává rotaci základního systému kolem osy x o úhel 70° a $\{135\}$ rotaci kolem osy z o úhel 135° (po směru hodinových ručiček)
- `\begin{tikzpicture}[tdplot_main_coords]`
 - instrukce pro zavedení prostředí `tikz` se zvoleným souřadnicovým systémem
- `\draw[thick,-stealth] (-6,0,0) -- (6,0,0)`
`node[anchor=north east] {x};`
`\draw[thick,-stealth] (0,-6,0) -- (0,6,0)`
`node[anchor=north west] {y};`

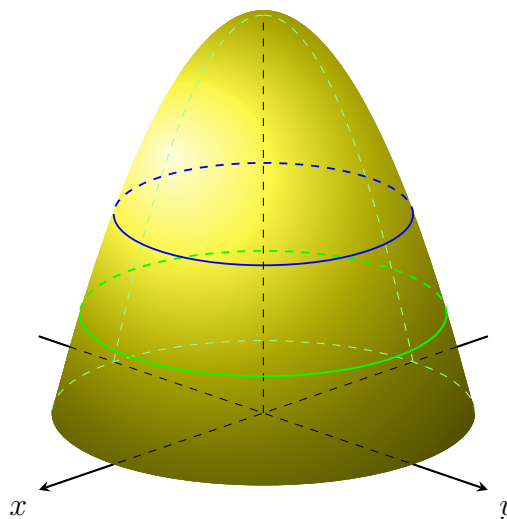
- vizualizace os x , y
- protože dochází k překrývání současného stavu obrázku následujícími elementy, jsou viditelné jen ty části os, které nepřekryje graf zadané funkce
- instrukce `node` umožňuje umístit uzel na stanovenou souřadnici a přiřadit jí popis, který je vepsán do složených závorek a pomocí volitelného argumentu v hranatých závorkách určit směr, kde bude vypsán
- `\coordinate (O) at (0,0,0);`
 - počátek $(0, 0, 0)$ označen jako pracovní bod (O) , který lze v průběhu tvorby použít
- `\shade[ball color=yellow!90!brown] (O) circle (4) -- (-2.828,2.828,0)coordinate(A) parabola bend (0,0,8.1) (2.828,-2.828,0)coordinate(B);`
 - graf zadané funkce - obrys paraboloidu je tvořen kružnicí `circle` se středem v bodě (O) a poloměrem (4) a parabolou ležící v rovině $x-y = 0$, s vrcholem v bodě `parabola bend` $(0,0,8.1)$, parabola má průsečíky s rovinou xy v bodech (A) , (B) (souřadnice lze vypočítat z rovnice zadaného paraboloidu jako průsečíky přímky $y = x$ a uvedené kružnice), vzniklá oblast se vystínuje příkazem `shade` s volitelným argumentem `ball`, který dodá efekt kulatosti a argumentem pro barvu složenou ze žluté a hnědé
 - kvůli perspektivě volíme vrchol paraboloidu v bodě $(0, 0, 8.1)$, nikoliv v bodě $(0, 0, 8)$, jak vyplývá z předpisu funkce



Obrázek 6. Tvorba obrázku 5 - fáze 1

- `\draw[dashed] (-6,0,0) -- (4,0,0);`
`\draw[dashed] (0,-6,0) -- (0,4,0);`
`\draw[dashed] (0,0,0) -- (0,0,8);`

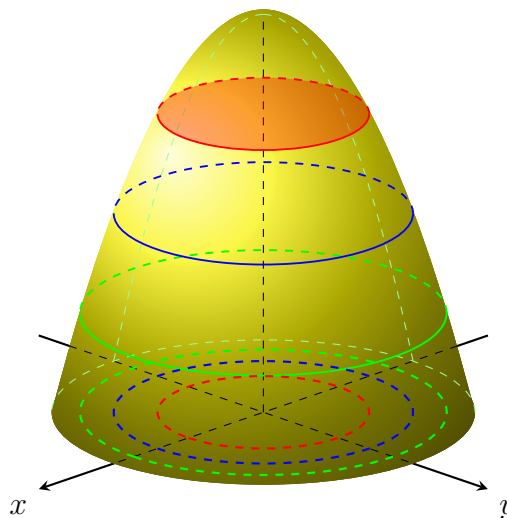
- části souřadných os, které jsou z hlediska viditelnosti překryty nebo leží uvnitř předcházejícího grafu
- ```
\draw [dashed,green!40](-4,0,0) parabola bend (0,0,8) (0,0,8);
\draw [dashed,green!40](0,-4,0) parabola bend (0,0,8) (0,0,8);
```
- čárkované paraboly nad zápornými částmi os  $x$ ,  $y$  - parabolám je ve volitelném argumentu přiřazena zelená barva dané sytosti
- ```
\draw[green!40,dashed] (-2.828,2.828,0) arc (135:315:4 and 4);
\draw[green,line width=0.7pt] (-2.449,2.449,2)
arc (135:-45:3.464 and 3.464);
\draw[green,dashed,line width=0.7pt] (-2.449,2.449,2)
arc (135:315:3.464 and 3.464);
\draw[blue,line width=0.7pt] (-2,2,4)
arc (135:-45:2.828 and 2.828);
\draw[blue,dashed,line width=0.7pt] (-2,2,4)
arc (135:315:2.828 and 2.828);
```
- vrstevnice o kótách 0, 2, 4, kde plnou čarou jsou zobrazeny pouze jejich viditelné části
- kódem např. $(-2,2,4)\text{arc}(135:-45:2.828 \text{ and } 2.828)$ je definován oblouk arc kružnice o středu v bodě $(-2,2,4)$, poloměru 2.828 a v rozmezí 135° až -45° , což odpovídá viditelné části vrstevnice o kótě 4
- ve volitelném argumentu je zvolena barva a šířka čáry přiřazením hodnoty příkazu `line width=0.7pt`



Obrázek 7. Tvorba obrázku 5 - fáze 2

- ```
\fill[red,opacity=.35] (0,0,6)coordinate(06) circle (2);
\draw[red,line width=0.7pt] ({-sqrt(2)},{sqrt(2)},6)
arc (135:-45:2 and 2);
\draw[red,dashed,line width=0.7pt] ({-sqrt(2)},{sqrt(2)},6)
arc (135:315:2 and 2);
```

- příkaz `fill` vyplní ohraničenou plochu, kterou je kruh se středem v pracovním bodě (06) a poloměrem 2, červenou barvou s průhledností 65 procent
- ```
\draw[dashed,thick,green] (0) circle (3.464);
\draw[dashed,thick,blue] (0) circle (2.828);
\draw[dashed,thick,red] (0) circle (2);
```
- průměty vrstevnic do roviny xy - tvořeny jako čárkované kružnice se středem v pracovním bodě (0) příslušnou barvou

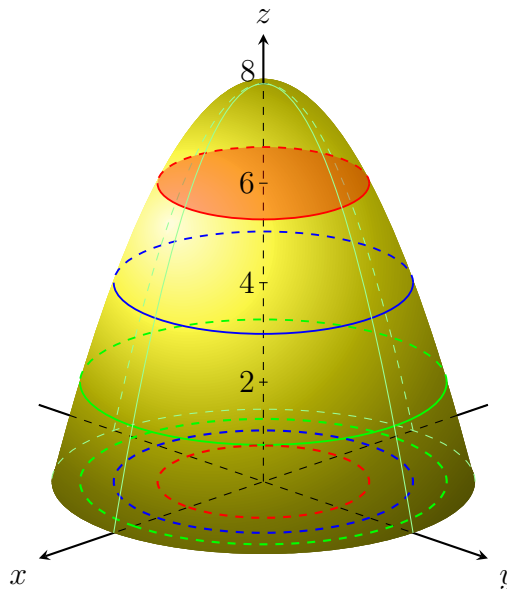


Obrázek 8. Tvorba obrázku 5 - fáze 3

- ```
\begin{scope}
\tdplotsetrotatedcoords{0}{0}{-45}
\draw[tdplot_rotated_coords,inner sep=1.5pt] (-0.08,0,8) --
(0.08,0,8);
\draw(0.08,0,7.98) node[above left,inner sep=1.5pt] {8};
\end{scope}
\begin{scope}
\tdplotsetrotatedcoords{0}{0}{-45}
\draw[tdplot_rotated_coords,inner sep=1.5pt] (-0.08,0,6) --
(0.08,0,6)node[left] {6};
\end{scope}
\begin{scope}
\tdplotsetrotatedcoords{0}{0}{-45}
\draw[tdplot_rotated_coords,inner sep=1.5pt] (-0.08,0,4) --
(0.08,0,4)node[left] {4};
\end{scope}
\begin{scope}
\tdplotsetrotatedcoords{0}{0}{-45}
\draw[tdplot_rotated_coords,inner sep=1.5pt] (-0.08,0,2) --
(0.08,0,2)node[left] {2};
\end{scope}
```

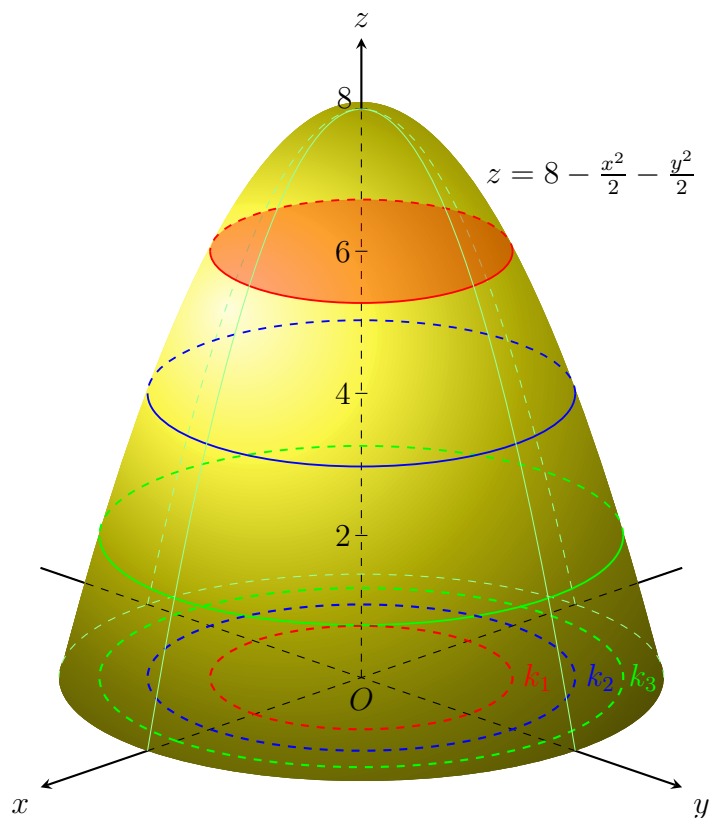
- označení jednotek na ose  $z$  - označení je provedeno vodorovnou čarou v systému souřadnic otočeném kolem osy  $z$  o  $-45^\circ$ , otočení bylo provedeno příkazem `tdplotsetrotatedcoords{0}{0}{-45}`

- každá čára je provedena ve vlastním otočeném prostoru uzavřeném příkazy `begin{scope}` a `end{scope}`
- popis je proveden vlevo od uzlu určeného souřadnicí
- `\draw[thick,-stealth] (0,0,8) -- (0,0,9) node[anchor=south] {$z$};`
  - viditelná část osy  $z$  - začíná ve vrcholu paraboloidu  $(0,0,8)$  a zároveň je nakreslena před samotným grafem
  - čára je vykreslena silně se šipkou
- `\draw [green!40] (4,0,0) parabola bend (0,0,8) (0,0,8);`  
`\draw [green!40] (0,4,0) parabola bend (0,0,8) (0,0,8);`
  - paraboly nad kladnými částmi os  $x, y$  na povrchu paraboloidu



Obrázek 9. Tvorba obrázku 5 - fáze 4

- `\node[below] at (0) {$0$};`  
`\node[right,red] at (-1.41,1.41,0) {$k_1$};`  
`\node[right,blue] at (-2,2,0) {$k_2$};`  
`\node[right,green] at (-2.4,2.4,0) {$k_3$};`  
`\node at (-1,3.3,7.7) {$z=8-\frac{x^2}{2}-\frac{y^2}{2}$};`
  - označení počátku  $(0)$  písmenem  $O$ , označení kružnic  $k_1, k_2, k_3$  (umístění vpravo od jednotlivých kružnic) a označení celého grafu
  - barva popisu kružnic odpovídá barvě vykreslení kružnic
- `\end{tikzpicture}`
  - instrukce pro ukončení prostředí `tikz`

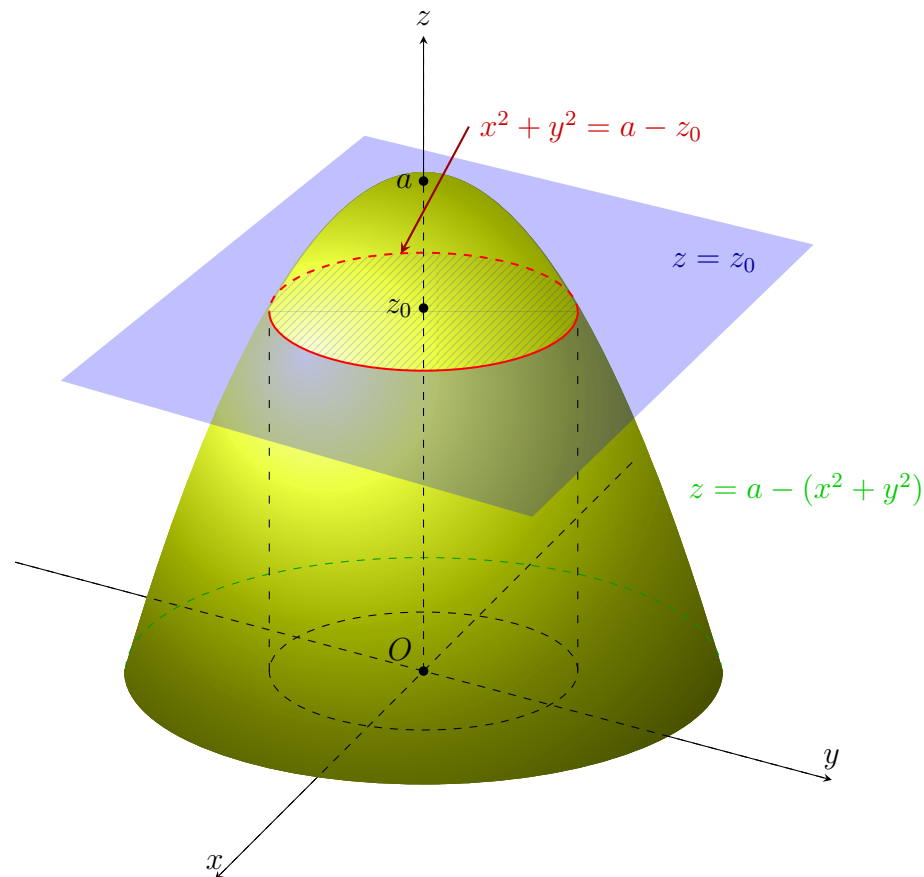


Obrázek 10. Tvorba obrázku 5 - fáze 5

Rozepsaný kód obrázku 5 popisuje fázově průběh tvorby grafu. Podobným postupem byla tvořena všechna následující grafická schémata. Jedná se o využití modifikovatelných instrukcí prostředí TikZ & PGF, z toho důvodu nevidíme potřebu rozepisovat zdrojové kódy ostatních obrázků.

### 4.3 Řez paraboloidu $z = a - (x^2 + y^2)$ rovinou $z = z_0$ ( $a > z_0$ )

Řez paraboloidu  $z = a - (x^2 + y^2)$  rovinou  $z = z_0$  ( $a > z_0$ ) byl tvořen ve dvojrozměrném systému souřadnic (jednotlivé souřadnice bodů mají dva rozměry), ve kterém byl zaveden obecný trojrozměrný systém. Paraboloid je vykreslen pomocí odhadnutých souřadnic podle předlohy. Rovina je umístěna ve výšce  $z_0$ .



Obrázek 11. Řez paraboloidu  $z = a - (x^2 + y^2)$  rovinou  $z = z_0$  ( $a > z_0$ )

Pro správnou funkčnost je důležité do obecné preambule, která je uvedena v předchozí kapitole, přidat knihovnu pro šrafování příkazem:

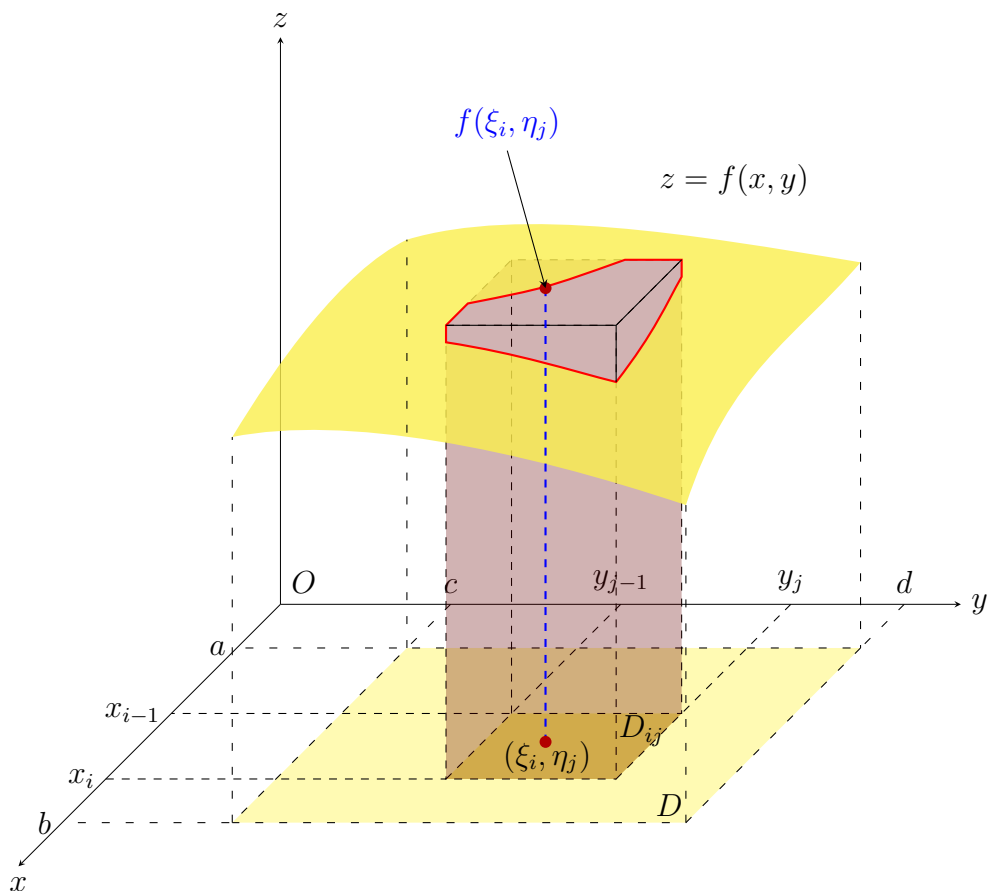
```
\usetikzlibrary[patterns]}
```

Ze zdrojového kódu můžeme vyčíst velmi málo složitých instrukcí, nejpodstatnější z nich je příkaz pro výřez viditelné části paraboloidu do modré roviny.

```
\fill[fill=blue,fill opacity=0.25, even odd rule]
(1.2,1.7)--(4.3,4.7)--(-.65,5.9)--(-4,3.2)--cycle
(-1.7,3.96) arc (180:360:1.7cm and .65cm)--(-1.72,3.96)
parabola bend (0,5.5) (1.72,3.96);}
```

Výřez se provádí volitelným argumentem `even odd rule`, který z první zadané plochy, odebere druhou zadanou plochu.

## 4.4 Dvojný integrál na obdélníku.



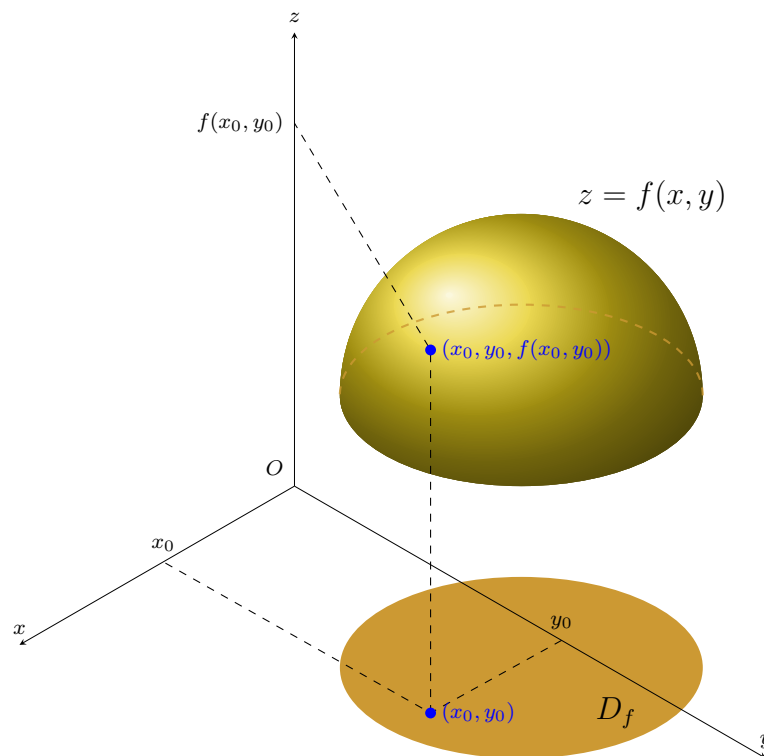
Obrázek 12. Dvojný integrál na obdélníku

Obrázek dvojného integrálu je umístěn v trojrozměrném prostoru, body jsou prostorově umístěny. Konstrukce je tvořena převážně rovnými čarami. Výjimkou je obrys řezné plochy a obvod vyříznuté plochy, které jsou tvořeny příkazy pro tvoření křivek. Konkrétně podinstrukcí `controls`, s následující syntaxí:

```
\draw [nastavení čáry] (referenční bod) .. controls
(bod, ke kterému křivka směřuje) .. (bod konce křivky);
```

Instrukce `controls` může obsahovat i více bodů, ke kterým se bude křivka přibližovat, tyto body musí být odděleny příkazem `and`.

Důležitým parametrem pro výplně je průhlednost, která se zavádí jako volitelný argument `opacity`. Průhlednost se uvádí desetinným číslem od 0 do 1. Jednička je pro neprůhlednou plochu. Šedesátiprocentní průhlednost bude zapsána `opacity=0.4`.

4.5 Definiční obor funkce dvou proměnných  $z = f(x, y)$ Obrázek 13. Definiční obor funkce dvou proměnných  $z = f(x, y)$ 

Obrázek definičního oboru je psán ve stejném systému souřadnic jako obrázek 11. Rozdíl je jen v zadání os. V tomto obrázku jsou osy vyneseny pomocí polárních souřadnic, které se zadávají úhlem a vzdáleností.

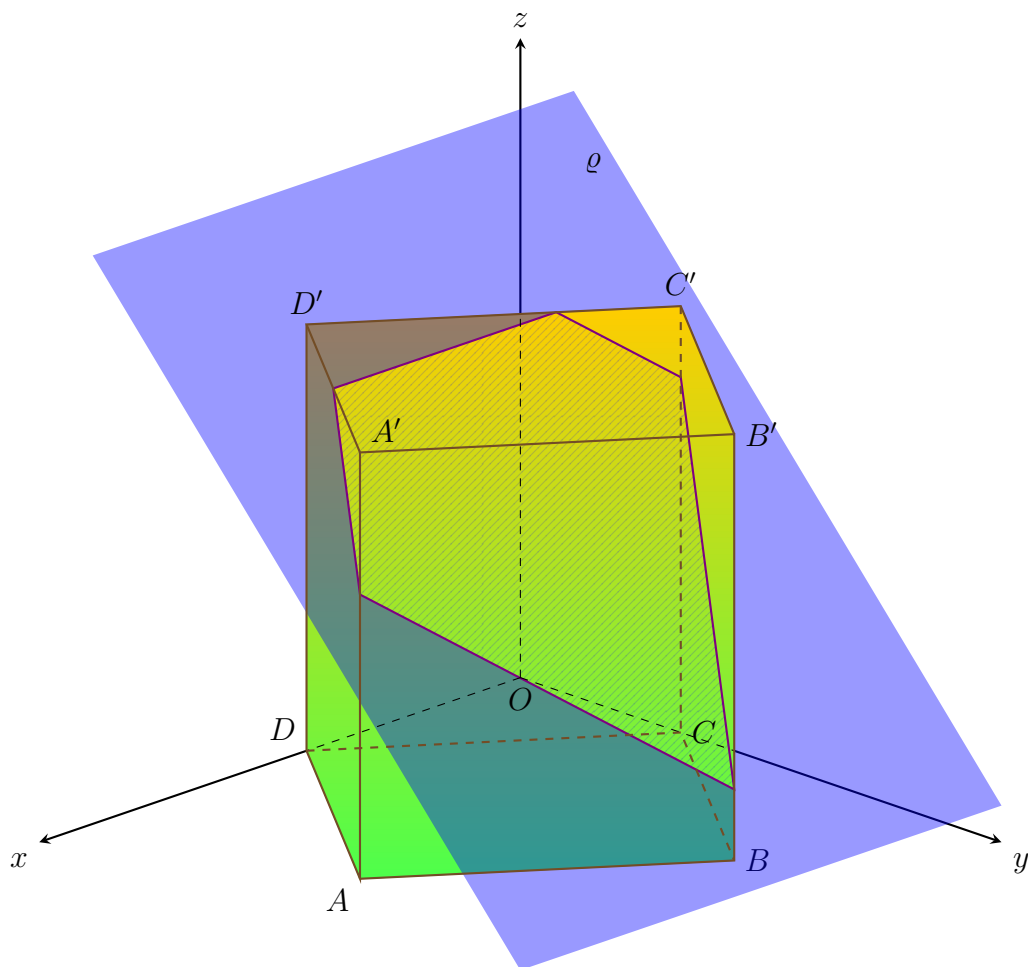
V kódu bylo použito složeného příkazu `shadedraw` (`shade` a `draw`), který umožňuje stínované výplně a zároveň obsahuje plochu čarou. Má své specifické volitelné parametry, jako je parametr `ball`, který vytvoří grafický efekt podobný jako můžeme vidět při pohledu na kouli.

V kódu bylo užito pro snadné vyznačení více bodů najednou následné dvojstručce:

```
\foreach \p in {p0,p}
\fill [red] (\p) circle (2pt);
```

Instrukce `foreach` přiřazuje zadaným bodům příkaz k vykreslení. První řádek určuje body, ve kterých chceme vyznačení provést a druhý řádek příkaz na to, co se s body provede. V našem příkladu se vyznačí kruhem o poloměru `2pt`.

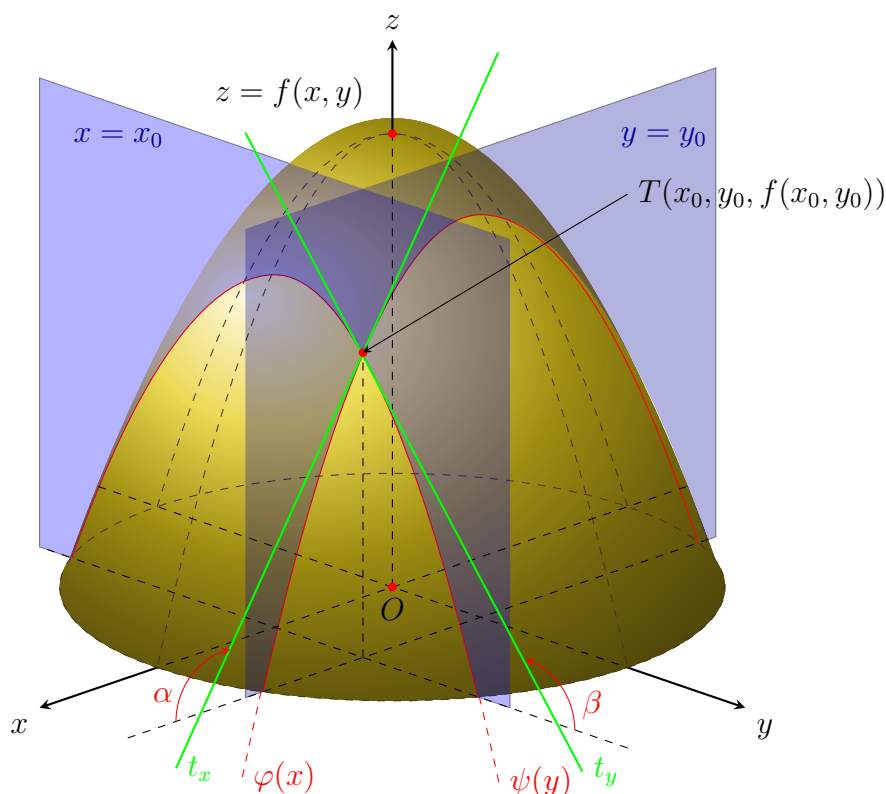


4.6 Řez kvádrů obecnou rovinou  $\varrho$ Obrázek 14. Řez kvádrů obecnou rovinou  $\varrho$ 

V obrázku 14 je užit nadstavbový balík `tikz-3dplot`, který musíme společně s knihovnou pro šrafování přidat do preambule.

V kódu bylo hojně užito instrukce `coordinate` pro jednodušší práci s určenými body. Plocha řezu je vyšrafována severovýchodními čarami.

## 4.7 Parciální derivace funkce dvou proměnných

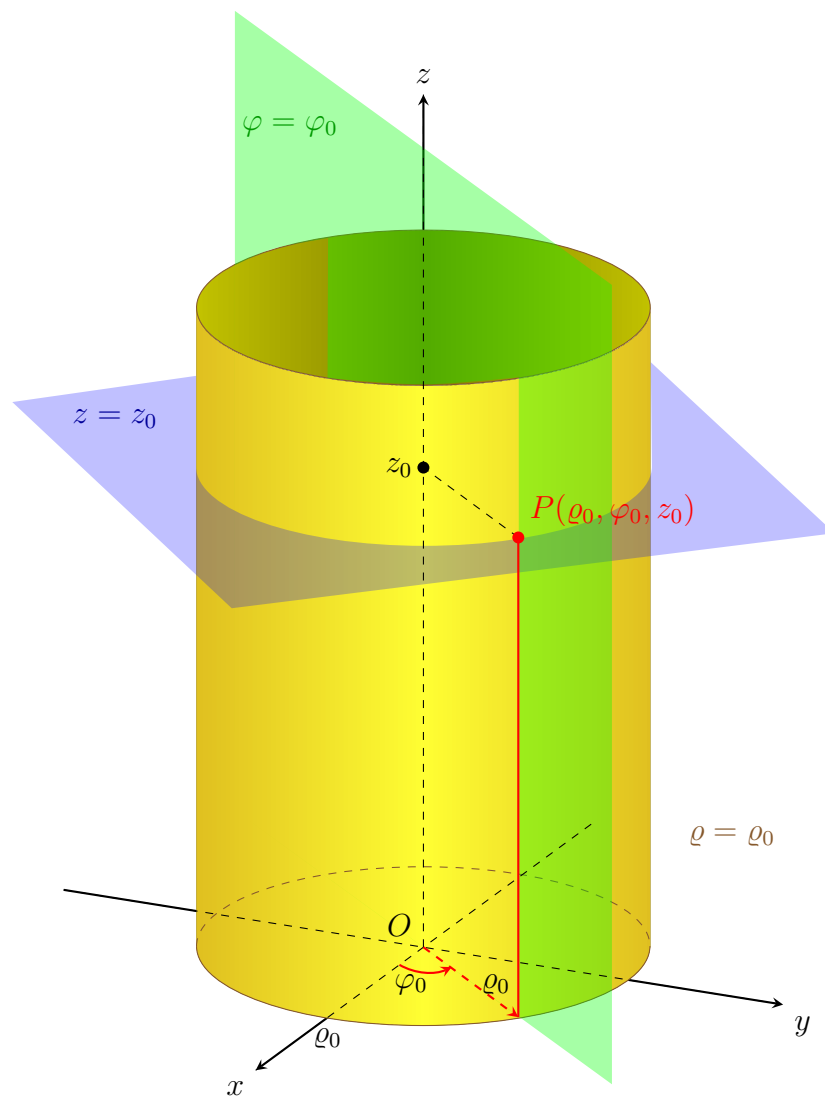


Obrázek 15. Parciální derivace funkce dvou proměnných

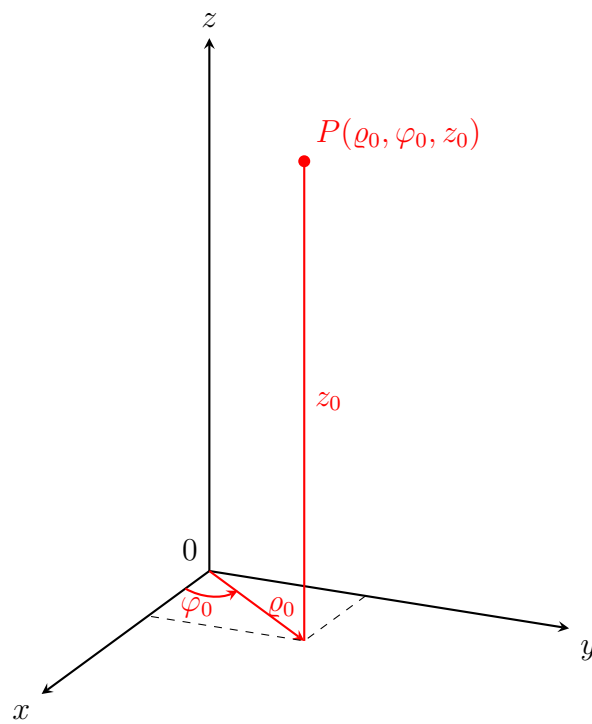
Obrázek 15 zobrazuje geometrický význam parciálních derivací funkce dvou proměnných. Obrázek je sázen pomocí balíku `tikz-3dplot`. Ve zdrojovém kódu obrázku jsou efektivně využity základní instrukce. Výsledný brázek poukazuje na sílu balíku `TikZ & PGF`, který umožňuje několika příkazy vykreslit složitý graf.

#### 4.8 Válcové (cylindrické) souřadnice

Grafické schéma se skládá ze dvou obrázků. Na prvním obrázku je zobrazen bod  $P(\varrho_0, \varphi_0, z_0)$  v cylindrických souřadnicích jako průsečík rovin  $z = z_0$ ,  $\varphi = \varphi_0$  a válcové plochy  $\varrho = \varrho_0$ . Na druhém obrázku je zobrazen pro lepší viditelnost tento bod ve zjednodušené formě.



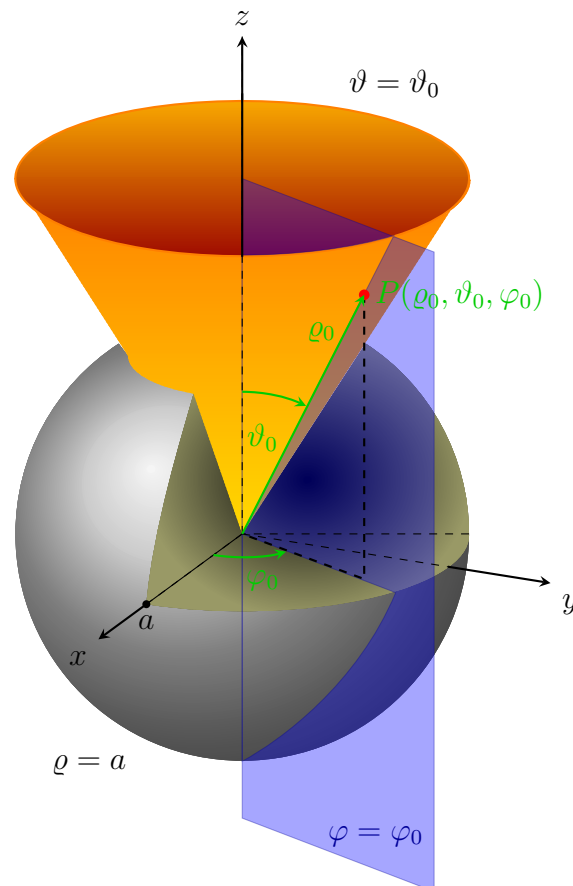
Obrázek 16. Válcové (cylindrické) souřadnice



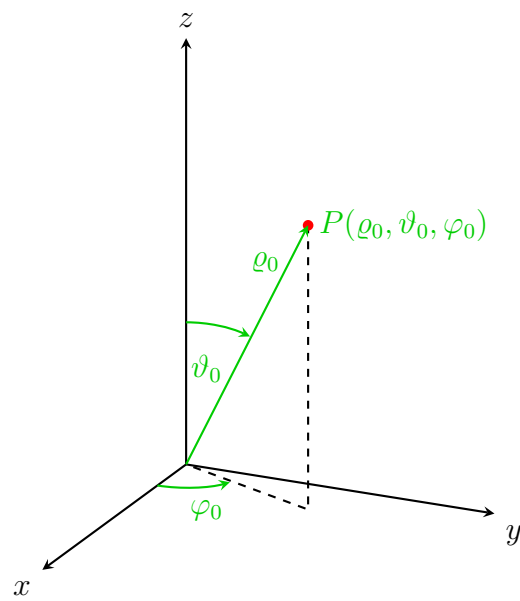
Obrázek 17. Umístění bodu  $P$  ve válcových (cylindrických) souřadnicích

#### 4.9 Kulové (sférické) souřadnice

Sférické souřadnice jsou podobně jako cylindrické souřadnice tvořeny dvojicí obrázků. Před tvorbou náročnějších grafických konstrukcí je vhodné vytvořit náčrt a odhadnout polohu důležitých bodů konstrukce.



Obrázek 18. Kulové (sférické) souřadnice



Obrázek 19. Umístění bodu P v kulových (sférických) souřadnicích

## ZÁVĚR

Cílem této práce bylo vytvořit přehled nástrojů, které je možno použít v systému  $\LaTeX$  pro tvorbu vektorových schémat obsahující grafy matematických funkcí a realizovat zadaná grafická schémata pomocí nejvhodnějšího z těchto nástrojů.

Pro vytvoření práce jsem zvolil profesionální sázečí systém  $\LaTeX$ , který se primárně používá pro tvorbu vědeckých a matematických dokumentů. S  $\LaTeX$ em jsem do doby začátku realizace bakalářské práce neměl žádnou zkušenost.

V teoretické části jsem podrobně shrnul mé poznatky z prvních setkání se systémem  $\LaTeX$ . Poté jsem se seznámil s grafickými nástroji  $\LaTeX$ u, z nich zvolil čtyři významné nástroje pro tvorbu grafických schémat (vkládání obrázků, prostředí `picture`, `PSTricks` a `TikZ & PGF`) a nastínil jejich možnosti použití. Na konci této části jsem je porovnal.

V praktické části jsem z vytvořeného přehledu vybral jeden (`TikZ & PGF`), který jsem shledával nejvhodnějším pro tvorbu zadaných schémat. Základní informace o způsobu tvorby grafických schémat v tomto prostředí jsem shrnul v kapitole *Úvod do tvorby grafických schémat*. V této kapitole je také seznámení se základními instrukcemi. V následující části prezentuji vypracovaná grafická schémata dle zadání a komentuji jejich zdrojový kód. Spustitelné a komentované zdrojové kódy jsou obsaženy v příloženém CD ve složce grafická schémata.

Během tvorby grafických schémat jsem několikrát konzultoval psaní kódu se zkušenějšími sázeči v systému  $\LaTeX$ . Česká literatura ke grafickým prostředím `TikZ & PGF` a `PSTricks` takřka neexistuje, z toho důvodu jsem přednostně čerpal z anglicky psaných manuálů.

Bakalářská práce mi rozšířila možnosti tvorby grafiky a technických dokumentů. Zajímavá je samotná tvorba grafiky přímo ve zdrojovém kódu  $\TeX$ ového dokumentu, která v mnohém připomíná programování. Práce v  $\LaTeX$ u je pro mne cennou empirií.

## CONCLUSION

The aim of this work was create overview of tools that can be used in system  $\LaTeX$  for creating vectorial schemes containing graphs of mathematical functions and realize assigned graphical schemes with help of the most appropriate of these tools.

To create the work I have chosen professional typesetting system  $\LaTeX$  that is primarily used for creation of scientific and mathematical documents. Until the start of realization of bachelor work I did not have any experience with  $\LaTeX$ .

In theoretical part I summarized my knowledge from the first encounter with system  $\LaTeX$ . After that I familiarized with graphical tools of  $\LaTeX$ , from which I chose four significant tools for creation of graphical schemes (`picture` inserting, `picture` environment, `PSTricks` and `TikZ & PGF`) and outlined possibilities of their use.

In practical part I chose from created overview one (`TikZ&PGF`), which I found the most appropriate for creation of assigned schemes. Basic information about the way of creating graphical schemes in this environment I summarized in chapter introduction to creation of graphical schemes. In this chapter is also familiarization with basic instructions. In following part I present made graphical schemes according the assignment and comment on their source code. Executable and commented source codes are included on enclosed CD in file graphical schemes.

During the creation of graphical schemes I several times consulted typing of the code with more experienced typesetters in system  $\LaTeX$ . Czech literature to graphical environment `TikZ & PGF` and `PSTricks` virtually does not exist, for that reason I preferentially drew from english written manuals.

Bachelor work expanded to me possibilities of creation of graphics and technical documents. Interesting is creation of graphics itself directly in the source code of  $\TeX$  document that in many ways reminds programming. Work in  $\LaTeX$  is for me a valuable experience.



## Reference

- [1] GOOSSENS, Michel; SEBASTIAN, Rahtz; FRANK, Mittelbach. *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion : Illustrating documents with T<sub>E</sub>X and PostScript*. Boston : Addison-Wesley, 1997. 554 s. ISBN 0201854694.
- [2] KOPKA, Helmut; DALY, Patrick W. *L<sup>A</sup>T<sub>E</sub>X – Kompletní průvodce*. Brno : Computer Press, 2004. 584 s. ISBN 80-7226-6973-9.
- [3] MARTINEK, David. *L<sup>A</sup>T<sub>E</sub>Xové speciality* [online]. 2010-2-25 [cit. 2010-05-23]. Dostupné z WWW: <<http://www.fit.vutbr.cz/~martinek/latex/>>.
- [4] OETIKER, Tobias, et al. *Ne příliš stručný úvod do systému L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  : Neboli LaTeX 2 $\epsilon$  v 73 minutách* [online]. [s.l.] : [s.n.], 1996, 1998 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>>.
- [5] OLŠÁK, Petr. *T<sub>E</sub>Xbook naruby*. 2. Brno : Konvoj, 2001. 468 s. Dostupné z WWW: <<ftp://math.feld.cvut.cz/pub/olsak/tbn/tbn.pdf>>. ISBN 80-7302-007-6.
- [6] RYBIČKA, Jiří. *L<sup>A</sup>T<sub>E</sub>X pro začátečníky*. 3. Brno : Konvoj+CSTUG, 2003. 240 s. ISBN 80-7302-049-1.
- [7] SOPUCH, Pavel. *L<sup>A</sup>T<sub>E</sub>X v kostce* [online]. 2001-1-20 [cit. 2010-05-23]. Dostupné z WWW: <<http://www.it.cas.cz/manual/latex/>>.
- [8] TANTAU, Till. *TikZ & PGF* [online]. [s.l.] : [s.n.], 2008 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>>.
- [9] *T<sub>E</sub>XUsers Group (TUG) home page* [online]. 2010 [cit. 2010-05-23]. Dostupné z WWW: <<http://tug.org/>>.
- [10] *TikZ and PGF examples* [online]. 2010 [cit. 2010-05-23]. Dostupné z WWW: <<http://www.texample.net/>>.
- [11] THOMAS, JR, George B., et al. *Thomas' Calculus : eleventh edition*. Massachusetts Institute of Technology : [s.n.], 2004. 1380 s. ISBN 978-0-321-48987-6.
- [12] VAVŘÍČEK, Jan. *LaTeX - VavříčekWiki* [online]. 2010-3-10 [cit. 2010-05-23]. Dostupné z WWW: <<http://vavricek.cs.vsb.cz/index.php/LaTeX>>.
- [13] ZANDT, Timothy Van. *PSTricks : PostScript macros for Generic TeX*. [online]. Princeton University : [s.n.], 1993 [cit. 2010-05-23]. Dostupné z WWW: <[http://www.tex.uniya.ru/doc/pst\\_ug.pdf](http://www.tex.uniya.ru/doc/pst_ug.pdf)>.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

|         |                               |
|---------|-------------------------------|
| bp      | jednotka velký bod            |
| cc      | jednotka cicero               |
| cm      | jednotka centimetr            |
| CMY     | Cyan, Magenta, Yellow         |
| dd      | jednotka Didôtův bod          |
| DVI     | DeVice Independent            |
| em      | jednotka šířky písmene M      |
| ex      | jednotka výšky písmene x      |
| in      | jednotka palec                |
| mm      | jednotka milimetr             |
| např.   | například                     |
| pc      | jednotka pica                 |
| PDF     | Portable Document Format      |
| PGF     | Portable Graphics Format      |
| PS      | PostScript                    |
| pt      | jednotka typografický bod     |
| RGB     | Red, Green, Blue              |
| TikZ    | TikZ ist kein Zeichenprogramm |
| tzv.    | takzvaný                      |
| viz     | lze vidět                     |
| WYSIWYG | What you see is what you get  |

## Seznam obrázků

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Obr. 1. Uživatelské rozhraní Technic Center .....                                    | 11 |
| Obr. 1. Trojúhelník v prostředí <code>picture</code> .....                           | 17 |
| Obr. 2. Použití vybraných příkazů .....                                              | 18 |
| Obr. 1. Bod v rovinné kartézské soustavě souřadnic.....                              | 26 |
| Obr. 2. Bod v prostorové kartézské soustavě souřadnic.....                           | 28 |
| Obr. 3. Bod v prostorové soustavě souřadnic 3Dplot.....                              | 29 |
| Obr. 4. Ukázka 2D grafů v <code>TikZ &amp; PGF</code> .....                          | 30 |
| Obr. 5. Vrstevnice funkce $z = 8 - \frac{x^2}{2} - \frac{y^2}{2}$ .....              | 32 |
| Obr. 6. Tvorba obrázku 5 - fáze 1.....                                               | 33 |
| Obr. 7. Tvorba obrázku 5 - fáze 2.....                                               | 34 |
| Obr. 8. Tvorba obrázku 5 - fáze 3.....                                               | 35 |
| Obr. 9. Tvorba obrázku 5 - fáze 4.....                                               | 36 |
| Obr. 10. Tvorba obrázku 5 - fáze 5.....                                              | 37 |
| Obr. 11. Řez paraboloidu $z = a - (x^2 + y^2)$ rovinou $z = z_0$ ( $a > z_0$ ) ..... | 38 |
| Obr. 12. Dvojný integrál na obdélníku.....                                           | 39 |
| Obr. 13. Definiční obor funkce dvou proměnných $z = f(x, y)$ .....                   | 40 |
| Obr. 14. Řez kvádrů obecnou rovinou $\rho$ .....                                     | 41 |
| Obr. 15. Parciální derivace funkce dvou proměnných .....                             | 42 |
| Obr. 16. Válcové (cylindrické) souřadnice.....                                       | 43 |
| Obr. 17. Umístění bodu P ve válcových (cylindrických) souřadnicích.....              | 44 |
| Obr. 18. Kulové (sférické) souřadnice.....                                           | 45 |
| Obr. 19. Umístění bodu P v kulových (sférických) souřadnicích .....                  | 46 |
| Obr. 1. Grafické znázornění parciální derivace podle $x$ .....                       | 53 |
| Obr. 1. Grafické znázornění parciální derivace podle $y$ .....                       | 54 |

## SEZNAM PŘÍLOH

- P I. Ukázka učebního textu 1
- P II. Ukázka učebního textu 2

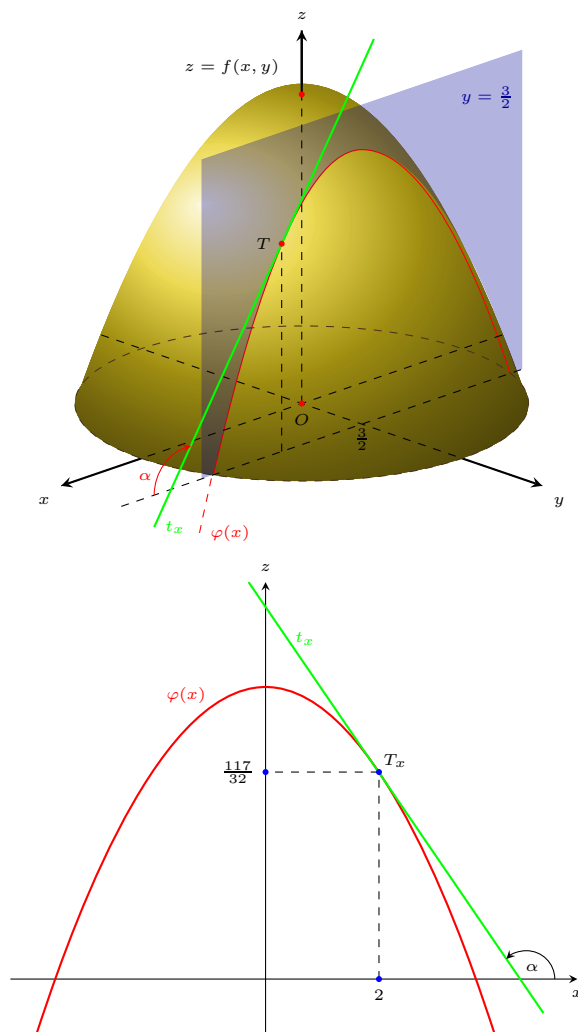
## Geometrický význam parciálních derivací funkce $z = f(x, y)$ podle $x$ v daném bodě

Geometrický význam parciální derivace funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  podle  $x$  v bodě  $(2, \frac{3}{2})$  je směrnice tečny ke křivce  $\varphi(x)$ , která je průnikem roviny  $y = \frac{3}{2}$  a funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  v bodě  $T = (2, \frac{3}{2}, \frac{117}{32})$ .

Parciální derivace v bodě  $(2, \frac{3}{2})$  podle  $x$ :

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x} \left[ 6 - \frac{3}{8}(x^2 + y^2) \right] = -\frac{3}{4}x, \quad \frac{\partial f}{\partial x} \left( 2, \frac{3}{2} \right) = -\frac{3}{2}.$$

Směrnice tečny v bodě  $(2, \frac{3}{2})$  má hodnotu  $-\frac{3}{2}$ , tzn.  $\text{tg } \alpha = -\frac{3}{2}$ , kde  $\alpha = 123.69^\circ$  je úhel, který svírá tečna s kladným směrem osy  $x$ . Průnikem roviny  $y = \frac{3}{2}$  a funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  je parabola o rovnici  $\varphi(x) = \frac{165}{32} - \frac{3}{8}x^2$ . Tečna  $t_x$  má rovnici  $z = -\frac{3}{2}x + \frac{213}{32}$ .



Obrázek 1. Grafické znázornění parciální derivace podle  $x$

## PŘÍLOHA P II. UKÁZKA UČEBNÍHO TEXTU 2

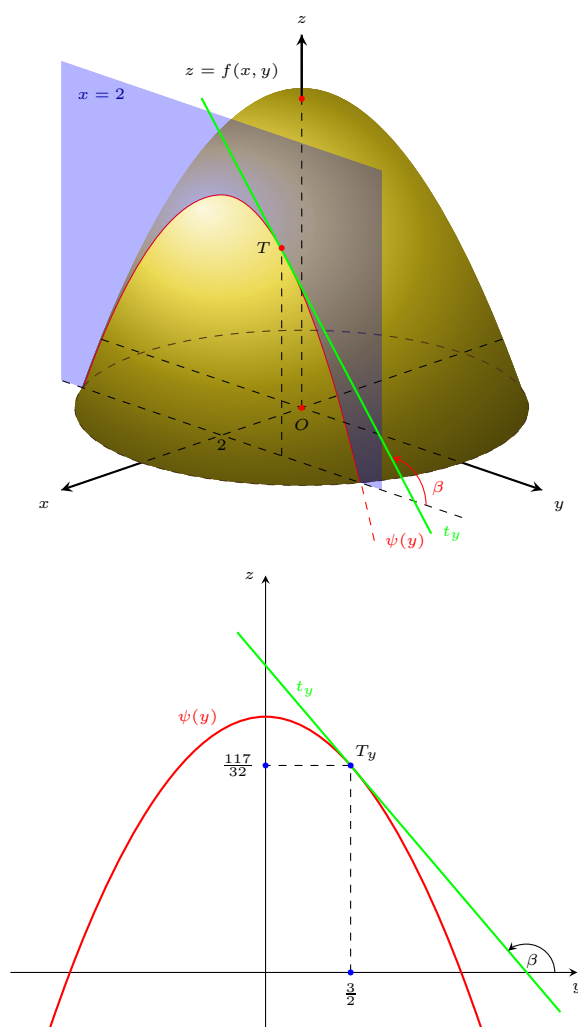
### Geometrický význam parciálních derivací funkce $z = f(x, y)$ podle $y$ v daném bodě

Geometrický význam parciální derivace funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  podle  $y$  v bodě  $(2, \frac{3}{2})$  je směrnice tečny ke křivce  $\psi(y)$ , která je průnikem roviny  $x = 2$  a funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  v bodě  $T = (2, \frac{3}{2}, \frac{117}{32})$ .

Parciální derivace v bodě  $(2, \frac{3}{2})$  podle  $y$ :

$$\frac{\partial f}{\partial y} = \frac{\partial}{\partial y} \left[ 6 - \frac{3}{8}(x^2 + y^2) \right] = -\frac{3}{4}y, \quad \frac{\partial f}{\partial y} \left( 2, \frac{3}{2} \right) = -\frac{9}{8}.$$

Směrnice tečny v bodě  $(2, \frac{3}{2})$  má hodnotu  $-\frac{9}{8}$ , tzn.  $\text{tg } \beta = -\frac{9}{8}$ , kde  $\beta = 131.63^\circ$  je úhel, který svírá tečna s kladným směrem osy  $y$ . Průnikem roviny  $x = 2$  a funkce  $f(x, y) = 6 - \frac{3}{8}(x^2 + y^2)$  je parabola o rovnici  $\psi(y) = \frac{9}{2} - \frac{3}{8}y^2$ . Tečna  $t_y$  má rovnici  $z = -\frac{9}{8}y + \frac{171}{32}$ .



Obrázek 1. Grafické znázornění parciální derivace podle  $y$