

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STAVEBNÁ FAKULTA

Evidenčné číslo: SvF-

NÁZOV PRÁCE

Bakalárska/Diplomová/Dizertačná práca

2022

Meno Priezvisko

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STAVEBNÁ FAKULTA

NÁZOV PRÁCE

Bakalárska/Diplomová/Dizertačná práca

Študijný program: Matematicko-počítačové modelovanie  
Študijný odbor: 9.1.9. Aplikovaná matematika  
Školiace pracovisko: Katedra matematiky a deskriptívnej geometrie  
Školiteľ: Ing. Mgr. Lukáš Tomek, PhD.

Bratislava 2022

Meno Priezvisko



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Mgr. Lukáš Tomek**  
ID študenta: 59517  
Študijný program: matematicko-počítačové modelovanie  
Študijný odbor: 9.1.9 aplikovaná matematika  
Vedúci práce: doc. Ing. Vladimír Kutiš, PhD.  
Pedagogický vedúci práce: doc. Ing. Vladimír Kutiš, PhD.  
Miesto vypracovania: Oddelenie aplikovanej mechaniky a mechatroniky FEI STU

Názov práce: **Analýza vlastných frekvencií dychového hudobného nástroja – fujary**

Špecifikácia zadania:

1. Vytvorenie geometrického a diskretizovaného modelu vzduchu vo fujare
2. Modálna analýza vzduchu vo fujare v programe ANSYS Multiphysics
3. Meranie frekvencií reálnej fujary
4. Porovnanie výsledkov modálnej analýzy a experimentu


Rozsah práce: 50 strán

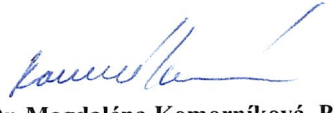
Riešenie zadania práce od: 01. 09. 2013

Dátum odovzdania práce: 22. 05. 2014

  
**Mgr. Lukáš Tomek**  
študent



  
**prof. Ing. Štefan Kozák, PhD.**  
vedúci pracoviska

  
**prof. RNDr. Magdaléna Komorníková, PhD.**  
garantka študijného programu

### **Čestné prehlásenie**

Prehlasujem, že som túto záverečnú prácu vypracoval/a samostatne pod vedením vedúceho záverečnej práce, s použitím literatúry uvedenej v zozname použitej literatúry.

Bratislava 6. 5. 2022

Meno Priezvisko

## **Podakovanie**

(Nepovinná, ale obvyklá časť práce)

Na tomto mieste môže byť podakovanie napr. školiteľovi práce, konzultantom, či iným ľuďom za pripomienky a odbornú pomoc pri vypracovaní záverečnej práce. Niekedy sa dáva aj na koniec Predhovoru.

Bratislava 6. 5. 2022

Meno Priezvisko

## Abstrakt

**Názov práce:** Názov práce v slovenčine

**Abstrakt:** Abstrakt má byť krátky, presný a zrozumiteľný text, ktorým predstavujete svoju prácu. Píše sa v prítomnom čase ako jeden odsek a mal by mať zhruba 50 – 300 slov. Najlepšie je písať ho nakoniec, keď už má človek všetko dobre uležané v hlave a venovať mu dostatočnú pozornosť. Pri odovzdaní práce sa slovenský aj anglický Abstrakt zadáva aj do AIS a dá sa chápať ako „upútavka“ na vašu prácu.

**Kľúčové slová:** 3 až 5 kľúčových slov/slovných spojení oddelených čiarkou

## Abstract

**Title:** Názov práce v angličtine

**Abstract:** Preklad abstraktu do angličtiny (poriadny, nie Google Translate). Je dobré ho robiť až keď je autor (a školiteľ) spokojný s tým slovenským, aby ste ho zbytočne nemuseli prekladať niekoľkokrát.

**Keywords:** 3 až 5 kľúčových slov/slovných spojení oddelených čiarkou



# Predhovor

(Nepovinná, ale užitočná časť práce, obvykle maximálne 1 strana)

V predhovore sa môže stručne uviesť, čím sa práca zaoberá, prípadne aké metódy, softvéry, či programovacie jazyky sa budú používať. Ak je práca súčasťou nejakého projektu, môžete načrtnúť, o čom je projekt a ako je práca zaradená do kontextu projektu. Popíše sa, aké sú ciele práce, môžu uviesť okolnosti vzniku témy, motivácia témy práce, prípadne dôvody, prečo ste si prácu vybrali. Je dobré stručne povedať o členení práce – o čom sú jednotlivé kapitoly, prípadne sekcie práce. Posledný odsek predhovoru môže byť poďakovanie (ak nebolo uvedené na osobitnej strane).





# Obsah

<b>1</b>	<b>Úvod</b>	<b>11</b>
<b>2</b>	<b>V akom editore písať? Aký kompilátor používať?</b>	<b>13</b>
<b>3</b>	<b>Členenie práce</b>	<b>15</b>
3.1	Podkapitola (sekcia)	15
3.1.1	Subsekcia	15
<b>4</b>	<b>Tipy na písanie v L<sup>A</sup>T<sub>E</sub>X-u</b>	<b>17</b>
4.1	Vymenovanie, číslovanie	17
4.2	Rovnice	18
4.2.1	Fyzikálne jednotky	20
4.3	Definície, vety, dôkazy	20
4.4	Referencie a citácie	21
4.5	Obrázky	22
4.5.1	Vkladanie obrázkov	22
4.5.2	Vytváranie obrázkov	24
4.6	Tabuľky	28
4.7	Vkladanie pseudokódu a kódu	29
<b>5</b>	<b>Záver</b>	<b>31</b>
	<b>Bibliografia</b>	<b>33</b>



# Kapitola 1

## Úvod

Úlohou úvodu je uviesť čitateľa do problematiky práce. Ak ste nepísali Predhovor, tak na tomto mieste môžete popísať ciele práce. Obvykle sa tu nezachádza do detailov teórie a nebýva tu veľa vzorcov. Môžete tu prípadne vysvetliť základné pojmy, s ktorými budete narábať.

Je tu priestor na charakterizáciu stavu poznania v oblasti, ktorá je predmetom záverečnej práce, citovanie literatúry (knihy, vedecké články, iné záverečné práce) ktorá rieši podobnú problematiku, uviesť, z akých zdrojov vychádzate alebo na ne priamo nadväzujete. (Toto je veľmi dôležité v dizertačnej práci.)

Ak ste nepísali Predhovor, tak je dobré na konci Úvodu stručne povedať o členení práce – o čom sú jednotlivé kapitoly, prípadne sekcie práce.



## Kapitola 2

# V akom editore písať? Aký kompilátor používať?

### Editor

Existuje niekoľko populárnych LaTeX editorov: TeXstudio, TeXmaker, TeXnicCenter, LyX, Overleaf (ten je online),... Dobrý prehľad nájdete [napríklad tu](#). **Odporúčam TeXstudio**, dá sa v ňom pracovať veľmi efektívne. V akomkoľvek editore sa naučte **používať klávesové skratky** (v TeXstudiu ich nájdete v Options/Configure TeXstudio/Shortcuts), rýchlo sa orientovať a preklikávať v TeX súbore pomocou referencií, preklikávať sa medzi TeX súborom a pdf, neuveriteľne to zrýchli prácu.

Ak sa vám páči mať vo veciach systém a poriadok, tak dobrý nápad je priebežne prácu commitovať do repozitára na [GitHub-e](#). Mimochodom, GitHub je veľmi užitočný pri programovaní – hlavne ak s niekým spolupracujete (napríklad so školiteľom) alebo píšete komplexnejší program (v rámci záverečnej práce to tak často býva). Ak GitHub nepoužívate, odporúčam začať.

### Kompilátor

Akýkoľvek editor si zvolíte, tak na kompilovanie odporúčam **kompilátor XeLaTeX**. Výsledné PDF bude mať správne kódovanie slovenských znakov<sup>1</sup> (v texte sa bude dať dobre vyhľadávať). Pre väčšinu editorov (aj TeXstudio) sa dá kompilátor nastaviť aj priamo v tex súbore pomocou tzv „magic commentu“, v súbore `_Zaverecna_praca.tex` je to riadok `% !TeX program = xelatex`, ktorý editoru hovorí, že tento dokument má kompilovať pomocou kompilátora XeLaTeX. Samozrejme, v TeXstudiu je možné aj prenastaviť defaultný kompilátor pre všetky dokumenty: Options/Configure TeXstudio/Build/Default compiler/XeLaTeX.

Ak ale chcete kompilovať pomocou **kompilátora PdfLaTeX**, treba správne kódovanie v PDF zabezpečiť pomocou pridania balíkov do preamble. Funguje napríklad táto kombinácia (ale sú aj iné možnosti):

```
\usepackage[utf8]{inputenc} % input encoding
\usepackage[T1]{fontenc} % spravne makcene
\usepackage{lmodern} % Latin modern fonts
```

V súbore `_Zaverecna_praca.tex` treba ešte vymazať magic comment `% !TeX program = xelatex` (ak defaultne používate PdfLaTeX) resp. ho prepísať na `% !TeX program = pdflatex`.

---

<sup>1</sup>Vďaka tomu, že XeLaTeX používa Unicode.

## Porovnávanie verzií (diff)

Keď školiťelovi posielate novú verziu, je veľmi užitočné mu v nej vyznačiť zmeny oproti poslednej verzii, ktorú čítal. Odporúčam na to výborný skript **latexdiff** napísaný v jazyku Perl. Je to analógia funkcie Track changes vo Worde, ktorú možno poznáte. Skript sa inštaluje napríklad takto:

1. Nainštalujte latexdiff. Napríklad cez MiKTeX Console (Run as administrator), Packages.
2. Nainštalujte [Strawberry Perl](#).

Keď už máte skript nainštalovaný, používa sa veľmi jednoducho a rýchlo (keď už sa to naučíte):

1. V príkazovom riadku sa nastavíte do správneho priečinka (rýchly spôsob v TeXstuidu je Tools/Open External Terminal, a tam prípadne príkaz `cd ..` na posun o priečinkov vyššie).
2. V príkazovom riadku spustíte príkaz

```
latexdiff --flatten stara_verzia.tex nova_verzia.tex > diff.tex
```

Vstupom je stará verzia dokumentu `stara_verzia.tex` nová verzia `nova_verzia.tex` a výstupom je súbor `diff.tex`, v ktorom sú vyznačené zmeny. Nastavenie `--flatten` je potrebné, keď sa dokument skladá z viacerých súborov (ako táto šablóna). Použitie skriptu na túto šablónu môže vyzeráť takto

```
latexdiff --flatten SK_verzia_stara\_Zaverecna_praca.tex SK_verzia\_Zaverecna_praca.tex > Porovnanie_verzii\diff.tex
```

V príkazovom riadku teda musíte byť v priečinku `ZaverecnaPracaMPM`, v ktorom sa nachádza priečinkov so starou verziou<sup>2</sup> `SK_verzia_stara` aj s novou verziou `SK_verzia`, ak to máte inde, cesty si upravte, ale pozor na medzery v názvoch priečinkov.

3. Vytvorte priečinkov `Porovnanie_verzii` a dajte doňho všetky obrázky a dokumenty (podpriečinky `figures` a `documents`), ktoré sa do práce vkladajú a aj bibliografický `.bib` súbor.
4. Súbor `diff.tex` skompilujete, čím vytvoríte výsledné `diff.pdf`, ktoré môžete spolu s aktuálnou verziou poslať školiťelovi.

---

<sup>2</sup>Na uchovávanie starých verzií aj tu výborne poslúži GitHub, ale ak ho nepoužívate, tak si priebežne zálohujte celý priečinkov s prácou.

## Kapitola 3

# Členenie práce

Keď začínate prácu písať, je dobré urobiť si základnú kostru – štruktúru kapitol, podkapitol,... Premyslieť si a napísať krátku poznámku, o čom sa v časti bude hovoriť a až potom začať jednotlivé časti postupne naplňať. Samozrejme, členenie sa časom bude upravovať, ale je dôležité mať na začiatku nejakú predstavu.

Okrem základného členenia (`\chapter`, `\section`, `\subsection`,...) sa v špecifických prípadoch môže hodiť rozdelenie na časti pomocou `\part` (o úroveň nad `\chapter`). Občas môže byť vhodné niektoré časti práce<sup>1</sup> dať až za Záver do dodatkov čiast s dodatkami sa štartuje príkazom `\appendix`.

### 3.1 Podkapitola (sekcia)

Blabla

#### 3.1.1 Subsekcia

Blabla

#### Subsubsekcia

Blabla

---

<sup>1</sup>Napríklad také časti, ktoré nie sú nevyhnutne potrebné pre pochopenie práce alebo idú do prílišných detailov, ktoré by čitateľa mohli zahltiť.





## Kapitola 4

# Tipy na písanie v L<sup>A</sup>T<sub>E</sub>X-u

Úlohou tejto kapitoly nie je byť príručkou k L<sup>A</sup>T<sub>E</sub>X-u. Na internete ich nájdete kopec, výborná je napríklad [táto príručka na stránke Overleaf-u](#) alebo [táto na wikibooks](#). Pri písaní určite pomôžu aj rôzne fóra, napríklad <https://tex.stackexchange.com/>. Uvádzam len niekoľko základných tipov pre písanie matematickej záverečnej práce v L<sup>A</sup>T<sub>E</sub>X-u. Keď budete pri písaní niečo potrebovať, môžete sa inšpirovať, resp. si to odtiaľto skopírovať a upraviť.

Určite existuje veľa užitočných príkazov a balíkov, o ktorých neviem/nepoužívam ich. Človek sa v L<sup>A</sup>T<sub>E</sub>X-u má čo učiť celý život.

### 4.1 Vymenovanie, číslovanie

#### Vymenovávanie cez odrážky

- Prvá odrážka
- Druhá odrážka
- Poďme o level hlbšie
  - Odrážka o level hlbšie
  - Ešte jedna
- Posledná odrážka

#### Číslovanie

1. Ovocie
2. Zelenina
3. Pečivo
  - (a) Chlieb
  - (b) Rožky

#### Abecedné číslovanie

- a) Parabolická rovnica
- b) Hyperbolická rovnica
- c) Eliptická rovnica

## Číslovanie rímskymi číslicami

- i) Jablko
- ii) Banán
- iii) Mrkva

## Číslovanie bez medzier medzi položkami

- 1. Kameň
- 2. Papier
- 3. Nožnice

## 4.2 Rovnice

Písanie rovníc je veľká téma a je to jedna z nasilnejších stránok LaTeX-u. V podstate všetko, čo vás napadne napísať, sa dá napísať, len stačí vygoogliť ako. Veľmi dobre je písanie rovníc rozobraté [tu na stránke Overleaf-u](#), určite si to prebehnite (aj podstránky uvedené dole). Nebudem rozoberať, ako sa píšú matematické symboly, zlomky, sumy, integrály, matice,... (to si ľahko nájdete). Budem sa skôr venovať prostrediam pre tvorbu rovníc a referencovaniu rovníc, to býva väčší problém.

Na písanie rovníc sú dva režimy:

- **Inline:** Vzorec je súčasťou textu. Napríklad  $a^2 + b^2 = c^2$  alebo  $x \in \mathbb{R}$ . Je viacero ekvivalentných spôsobov, ako ich písať, najčastejšie sa používa syntax  $\$...\$$  alebo  $\backslash(\dots\backslash)$ .
- **Display:** Vzorec je na samostatnom riadku. Napríklad

$$f(x) = ax^2 + bx + c$$

alebo

$$\sin^2 x = \frac{1 - \cos 2x}{2}, \quad \cos^2 x = \frac{1 + \cos 2x}{2}.$$

Keď sa na rovnicu chceme v texte odvolať, použijeme automatické číslovanie

$$\int_V \nabla \cdot \vec{F} dx = \int_{\partial V} \vec{F} \cdot \vec{n} dS. \quad (4.1)$$

Keď potrebujete napísať dlhú rovnicu na viac riadkov alebo viac rovníc pod seba a zarovnať ich, je na to v balíku `amsmath` veľa prostriedí. Neurobte tú chybu (čo som kedysi robil aj ja), že sa naučíte iba jedno – dve a s nimi budete robiť. Pri písaní vzorcov sa vyskytnú rôzne situácie a vhod môže prísť raz jedno prostredie, raz iné. Niekedy je dobré skúsiť viac možností a zistiť, čo vyzerá lepšie. Často existuje aj viacero riešení pre dosiahnutie rovnakého výsledku. Uvediem niekoľko príkladov prostriedí, ktoré by sa mohli hodiť (niektoré sú zo stránky Overleaf-u).

Prostredie `align` sa používa takto

$$x + y = 1, \quad (4.2)$$

$$2x - y = 5. \quad (4.3)$$

Komplikovanejší príklad použitia **align** je tu

$$\begin{array}{lll} x = y, & w = z, & a = b + c, \\ 2x = -y, & 3w = \frac{1}{2}z, & a = b, \\ -4 + 5x = 2 + y, & w + 2 = -1 + w, & ab = cb. \end{array}$$

Ak máme viacero stĺpcov rovníc (ako v predchádzajúcom príklade), vhodné je aj prostredie **alignat**. V ňom sa dá dobre ovládať vzdialenosť stĺpcov rovníc.

$$\begin{array}{lll} x = y, & w = z, & a = b + c, \\ 2x = -y, & 3w = \frac{1}{2}z, & a = b, \\ -4 + 5x = 2 + y, & w + 2 = -1 + w, & ab = cb. \end{array}$$

Ak chcete dve rovnice, ale len jeden label, môžete použiť napríklad **split** vnútri **equation**

$$\begin{aligned} A &= \frac{\pi r^2}{2}, \\ &= \frac{1}{2}\pi r^2. \end{aligned} \tag{4.4}$$

Ak máme sústavu rovníc (4.5), ktorú chceme zarovnať do viacero stĺpcov a zároveň ju chceme očíslovať jedným labelom, dá sa to urobiť pomocou prostredia **aligned** vnútri **equation** takto

$$\begin{array}{ll} 0. \text{ rovnica} & u_0 = 0, \\ 1. \text{ rovnica} & -u_0 + 2u_1 - u_2 = f(x_1)h^2, \\ 2. \text{ rovnica} & -u_1 + 2u_2 - u_3 = f(x_2)h^2, \\ & \vdots \\ i. \text{ rovnica} & -u_{i-1} + 2u_i - u_{i+1} = f(x_i)h^2, \\ & \vdots \\ (n-1). \text{ rovnica} & -u_{n-2} + 2u_{n-1} - u_n = f(x_{n-1})h^2, \\ n. \text{ rovnica} & u_n = 1. \end{array} \tag{4.5}$$

Ak sa vám rovnica nezmestí na riadok, môžete použiť prostredie **multline**

$$\begin{aligned} p(x) &= c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6 + c_7x^7 + c_8x^8 \\ &\quad + c_9x^9 + c_{10}x^{10} + c_{11}x^{11} + c_{12}x^{12} + c_{13}x^{13} + c_{14}x^{14}, \end{aligned} \tag{4.6}$$

ale dá sa to (inak) vyriešiť aj pomocou **align**

$$\begin{aligned} p(x) &= c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6 + c_7x^7 + c_8x^8 \\ &\quad + c_9x^9 + c_{10}x^{10} + c_{11}x^{11} + c_{12}x^{12} + c_{13}x^{13} + c_{14}x^{14}. \end{aligned} \tag{4.7}$$

Záleží od situácie, čo sa viac hodí. Po častiach definovaná funkcia sa robí pomocou prostredia **cases** takto

$$w(x) = \begin{cases} (x - x_1)^2(x - x_2)^2, & x \in [x_1, x_2], \\ 0, & \text{inak.} \end{cases}$$

Prostredie **aligned** vnútri **equation** sa hodí napríklad v tejto situácii

$$\begin{array}{ll} a + bx_1^e = u_1^e \\ a + bx_2^e = u_2^e \end{array} \quad \text{resp. maticovo} \quad \begin{bmatrix} 1 & x_1^e \\ 1 & x_2^e \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}. \tag{4.8}$$

Niekedy pri zarovnávaní môže byť vhodné použiť aj prostredie `array`

$$\begin{array}{lcl} \Omega^{e-1} : & \begin{array}{l} K_{11}^{e-1}u_1^{e-1} + K_{12}^{e-1}u_2^{e-1} = f_1^{e-1} + Q_1^{e-1}, \\ K_{21}^{e-1}u_1^{e-1} + K_{22}^{e-1}u_2^{e-1} = f_2^{e-1} + Q_2^{e-1}, \end{array} \\ \Omega^e : & \begin{array}{l} K_{11}^e u_1^e + K_{12}^e u_2^e = f_1^e + Q_1^e, \\ K_{21}^e u_1^e + K_{22}^e u_2^e = f_2^e + Q_2^e. \end{array} \end{array}$$

Kvôli ľahšiemu referencovaniu sa občas môže hodiť aj prostredie `subequations`. Maxwellove rovnice majú tvar

$$\nabla \cdot \mathbf{D} = \rho, \quad (4.9a)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (4.9b)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (4.9c)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}. \quad (4.9d)$$

Môžeme sa potom na ne odvolávať naraz (4.9) alebo aj jednotlivo (4.9b).

**Poznámka 4.1.** Existuje ešte aj prostredie `eqnarray`, ale [nepoužívajte ho, môžu s ním byť problémy](#).

**Poznámka 4.2** (Vzorcová interpunkcia). V slovenčine by sa mala používať vzorcová interpunkcia – aj vzorec v režime display je súčasťou vety, a preto by sa mala používať interpunkcia (čiarky, resp. bodky) podobne, akoby rovnica bola v režime inline, pozri napríklad rovnice (4.2) a (4.3). Sú prípady, kedy interpunkcia môže byť vizuálne rušivá,<sup>1</sup> vtedy jej použitie treba zvážiť.

### 4.2.1 Fyzikálne jednotky

Na písanie fyzikálnych jednotiek je vhodný balík `siunitx`. Základné použitie vyzerá takto

- Jednotka zrýchlenia je  $\text{kg m s}^{-2}$ .
- Gravitačné zrýchlenie má hodnotu  $g = 9,8 \text{ kg m/s}^2$ .
- Youngov modul pružnosti ocele je  $E = 200 \text{ GPa} = 2 \cdot 10^{11} \text{ Pa}$ .
- Pravý uhol je  $\pi/2 = 90^\circ$ , štvrtina z neho je  $\pi/8 = 22,5^\circ = 22^\circ 30'$ .

Ďalšie možnosti nájdete v dokumentácii balíka `siunitx` (v TeXstudiu Ctrl+klik na `siunitx` v preambule v riadku `\usepackage{siunitx}`).

## 4.3 Definície, vety, dôkazy

V niektorých prácach môže byť vhodné používanie štýlu „Definícia-Veta-Dôkaz“ (prípadne aj Lemma, Poznámka, Príklad,...). V preambule sú kvôli tomu riadky:

```
% Definície, Vety, Dôkazy...
\usepackage{amsthm}
\theoremstyle{definition}
\newtheorem{thm}{Veta}[chapter]
\newtheorem{defn}[thm]{Definícia}
```

<sup>1</sup>Čo je rušivé a čo nie, je otázka vkusu. Vkus sa aj časom mení, ja som napríklad kedysi vzorcovú interpunkciu nepoužíval vôbec.

```

\newtheorem{lem}[thm]{Lemma}
\newtheorem{cor}[thm]{Dôsledok}
\newtheorem{rem}[thm]{Poznámka}
\newtheorem{exp}[thm]{Príklad}

```

Nastavenie a spôsob číslovania jednotlivých prostredí si v týchto riadkoch môžete podľa potreby upraviť, viac sa dozviete napríklad [na tejto stránke](#).

Použitie vyzerá takto:

**Definícia 4.3** (Slabé riešenie). Nech funkcia  $f$  je z priestoru  $L_2(0, 1)$ . **Slabým riešením** okrajovej úlohy (??) nazývame funkciu  $u$  z priestoru  $V$  takú, že pre všetky váhové funkcie  $w \in V$  platí

$$(u, w)_A = (f, w).$$

**Veta 4.4** (Vlastnosť ortogonalít). Chyba  $\varepsilon = U - u$  je v zmysle energetického skalárneho súčinu kolmá (ortogonálna) na všetky funkcie  $w \in S \subset V$ , teda

$$(\varepsilon, w)_A = (U - u, w)_A = 0. \quad (4.10)$$

*Dôkaz.* Pre približné riešenie  $U$  platí rovnica

$$(U, w)_A = (f, w), \quad \text{pre všetky } w \in S \subset V. \quad (4.11)$$

Podobne pre slabé riešenie  $u$  podľa Definície 4.3 platí

$$(u, w)_A = (f, w), \quad \text{pre všetky } w \in V \text{ a teda aj pre všetky } w \in S \subset V. \quad (4.12)$$

Keď rovnice (4.11) a (4.12) od seba odčítame, dostaneme vlastnosť ortogonalít (4.10).  $\square$

## 4.4 Referencie a citácie

Na orientáciu v práci slúžia referencie. Môžu odkazovať na všeličo, len to musí byť označené pomocou `\label{name}`. Refazec `name` by mal byť výstižný (nie len nejaké číslo), v zdrojovom kóde je to potom prehľadnejšie a lepšie sa s labelami pracuje. Je vhodné si v labelovaní zaviesť nejakú konvenciu. Štandardne sa používa pre obrázky `\label{fig:name}`, pre rovnice `\label{eq:name}`, pre časti a kapitoly `\label{sec:name}` a podobne. Takáto konvencia je výhodná napríklad aj keď máte sekciu, obrázok a rovnicu, ktoré hovoria o jednej veci, tak to ľahko odlíšite. Refazec `name` môže obsahovať aj medzery, ale používajte radšej podčiarkovník. Uvediem niekoľko príkladov referencií.

**Kapitola, sekcia.** V kapitole 2 sme hovorili o tom, aký editor použiť. V časti 3.1 sme len bľabotali.

**Obrázok, tabuľka.** Pozri Obrázok 4.3. Tabuľka 4.1 je tiež pekná.

**Rovnica.** Vzťah (4.1) je Gaussova veta. Na referencovanie rovníc je efektívnejšie (rýchlejšie) používať `\eqref{label}`, nie `(\ref{label})`.

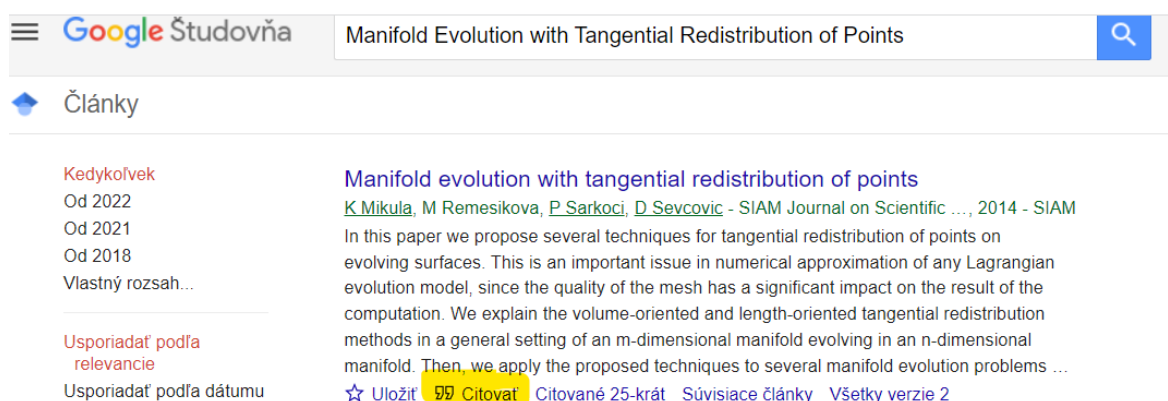
**Strana.** Občas sa zide referencia na stranu. Na strane 15 sme hovorili o členení práce.

V záverečnej práci je veľmi dôležité poriadne citovať zdroje. Najefektívnejšie je používať balík `biblatex`. Často sa stretávam s tým, že ľudia vypisujú bibliografiu „natvrdo“ manuálne. Nerobte túto chybu. Všetky zdroje treba dať do súboru `.bib`, ktorý je databázou vašich zdrojov. Balík `biblatex` sa postará o správne formátovanie a zoradenie položiek v bibliografii.

Ako do bibliografickej databázy v súbore `.bib` efektívne pridávať nové položky? Veľa z nich nájdete na internete v rôznych databázach publikácií a na stránkach vedeckých časopisov. Ručné vpisovanie bibliografických údajov by malo byť až poslednou voľbou.

- Odkazy na vedecké články a knihy väčšinou nájdete na pár klikov na <https://scholar.google.com/>. Ak chcete pridať odkaz napríklad na vedecký článok [3], stačí po vyhľadani kliknúť na „Citovať“ (Obr. 4.1) a zvolíte BiBTeX.
- Ak chcete pridať odkaz na vedecký článok, napríklad [3], položku do databázy s veľkou pravdepodobnosťou nájdete aj cez [www.researchgate.net](http://www.researchgate.net), klik na Download citation. Položku štandardne nájdete aj na [stránke časopisu](#), v ktorom článok vyšiel. Vždy treba hľadať niečo v zmysle „download citation“, „cite this paper“ a podobne, a následne zvoliť BiBTeX.

Na položky z bibliografickej databázy potom v texte odkazujete. Môžete citovať jednu položku [1], alebo aj viacero naraz [2, 3].



Obr. 4.1: Google scholar.

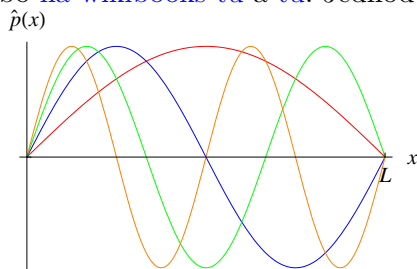
## 4.5 Obrázky

Obrázky sú veľmi dôležitý prvok záverečnej práce. Keď sú dobre urobené, môžu čitateľovi veľmi pomôcť prácu pochopiť. Je veľa možností, ako obrázky vkladať do TeX-u a ešte viac možností, ako ich vytvárať. Obrázky, ktoré chcete vložiť do práce je dobré mať pokope v priečinku/pričinkoch. V preambule je cesta k priečinku nastavená pomocou príkazu `\graphicspath{ {./figures/} }`.

### 4.5.1 Vkládanie obrázkov

#### Jeden obrázok

O tom, ako vkladať obrázky do TeX-u sa prehľadne píše napríklad [tu na stránke Overleaf-u](#) alebo [na wikibooks tu](#) a [tu](#). Jednoduché vloženie obrázka sa robí takto:

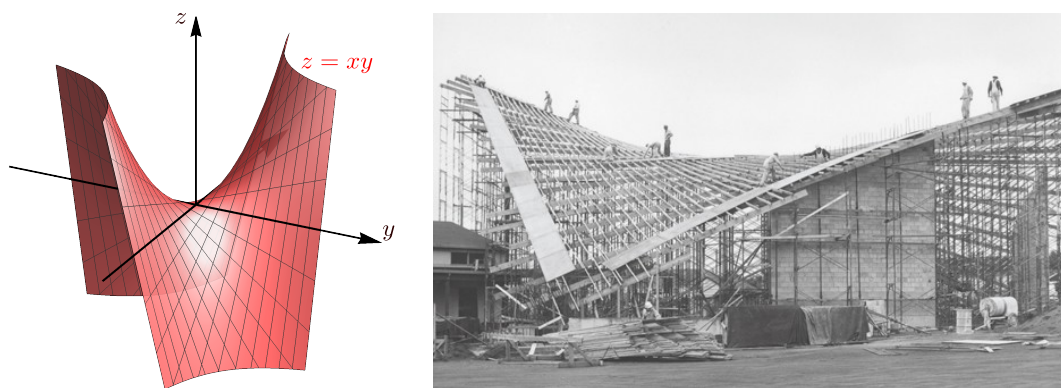


Takýto obrázok ale nie je vycentrovaný a nemá popis. V práci by mali mať všetky obrázky číslo a popis, potom sa na ne ľahko odvolávalo v texte. Takýmto príkladom je Obrázok 4.2.<sup>2</sup>

<sup>2</sup>Všimnite si, že v zdrojovom kóde je (zakomentovaný) `\vspace{...}`, pomocou ktorého môžete zväčšiť/zmenšiť biele miesto pod/nad obrázkom a pod popisom, to sa občas môže hodiť.







Obr. 4.5: Graf funkcie  $f(x, y) = xy$  a konštruovanie strechy v tvare hyperbolického paraboloidu (v menšom okolí sedlového bodu, než vykresľujeme v grafe).

Pomocou balíka `subcaption` sa dajú obrázky vložiť vedľa seba tak, ako vidíme na Obrázku 4.9, pričom môžu mať jeden hlavný popis a ďalšie podpopisy 4.9a resp. 4.9b.

#### 4.5.2 Vytváranie obrázkov

Z časti 4.5.1 už máme predstavu, ako vkladať existujúce obrázky (resp. výstupné obrázky zo softvéru, napríklad z ANSYS-u). V akom programe ale vytvárať nové grafy a obrázky? Toto je (prekvapujúco?) ťažká otázka.

##### Aké formáty používať?

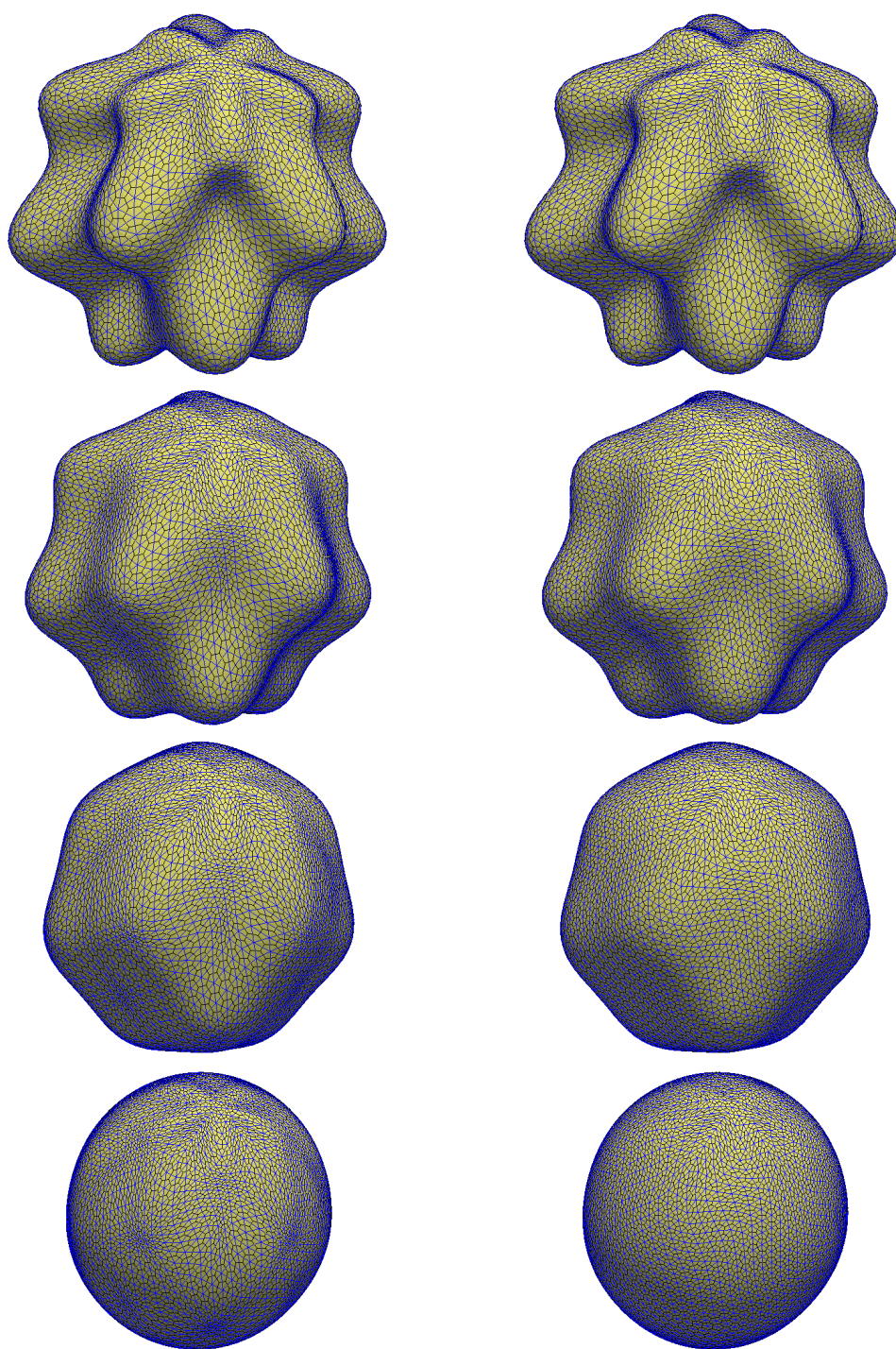
- **Vektorová grafika:** PDF, EPS (LaTeX ale aj tak EPS skonvertuje na PDF).
- **Rastrová grafika:** PNG, JPG (odporúčam radšej PNG, lebo pri ukladaní do JPG resp. JPEG sa kvôli kompresii znižuje kvalita a vzniká šum).

Kvôli kvalite obrázkov **preferujte vektorovú grafiku** (ak je to možné a ak vektorový obrázok vyzerá dobre). V elektronickej verzii práce sú vektorové obrázky aj po priblížení stále pekné, hladké. Pri grafike, ktorá je tieňovaná (napr. plochy v 3D) je lepšie obrázok v dostatočnej kvalite uložiť ako raster. Vo všeobecnosti je dobré obrázky ukladať tak, aby boli orezané, bez veľkého nadbytočného orámovania okolo toho, čo nás zaujíma.

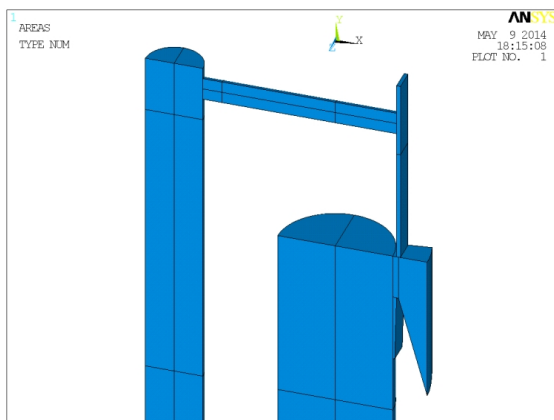
**Mathematica.** V softvéri Mathematica robiť viete, preto je dobrý nápad tvoriť grafy/obrázky v ňom. Pre export nepoužívajte screenshot, ale export:

- **2D grafika:** pravý klik na obrázok/Save Graphic As/pdf. Grafika bude automaticky vektorová.
- **3D grafika:** pravý klik na obrázok/Save Graphic As/png (obrázok treba mať pred uložením dostatočne veľký). Je dobré ešte pred uložením obrázka odstrániť Bounding Box: pravý klik na obrázok/ Trim Bounding Box, aby okolo obrázka nebolo zbytočne veľa bielej plochy.

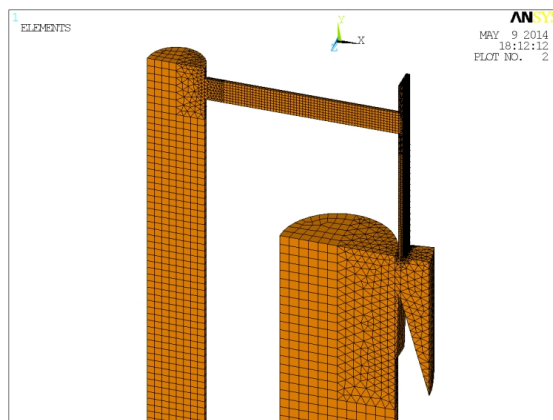
Niekedy potrebujeme obrázok uložiť aj spolu s legendou (prípadne inými grafickými objektami), vtedy treba **pravý klik na pravú lištu/Save Selection As/**, predtým je ešte vhodné vymazať `Out[1]=` na ľavej strane. Obrázok sa dá uložiť aj pomocou príkazu **Export** (zide sa napríklad vtedy, keď potrebujete export automatizovať počas behu kódu alebo si ukladanie zautomatizovať).



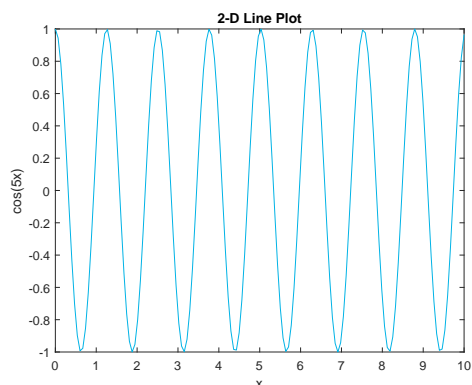
Obr. 4.6: Příklady obrázků z ParaView.



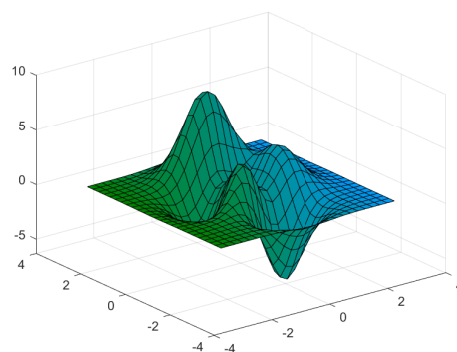
Obr. 4.7: Obrázok z ANSYS-u.



Obr. 4.8: Obrázok z ANSYS-u.



(a) Graf funkcie jednej premennej (2D).



(b) Graf funkcie v dvoch premenných (3D).

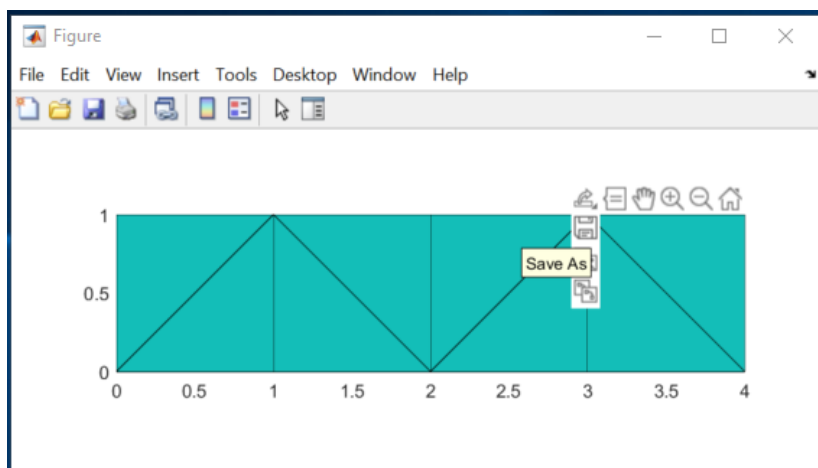
Obr. 4.9: Príklady obrázkov z MATLAB-u.

**MATLAB.** Ak viete robiť MATLAB-e, prípadne v ňom programujete program k záverečnej práci, je dobrý nápad robiť grafy/obrázky v ňom. Pre export nepoužívajte screenshot, ale **Save As** v okne Figure (aby sa ponuka **Save As** zobrazila, treba podržať kurzor myši priamo nad obrázkom a kliknúť na prvú možnosť, pozri Obr. 4.10) a vybrať formát (PDF resp. PNG). Ak z nejakého dôvodu potrebujete formát, ktorý nie je v ponuke (napr. EPS), sú aj ďalšie možnosti, napr. – v okne Figure klik na **File/Save As** a vybrať formát.<sup>4</sup> Je praktické si obrázky ukladať aj vo formáte **.fig**, aby ste ľahko mohli aj neskôr editovať rôzne nastavenia. Obrázok sa dá uložiť aj pomocou príkazu **exportgraphics**, v ktorom si nastavíte aj formát. Viac informácií o ukladaní obrázkov nájdete na [stránke MathWorks](#).

**ParaView.** Ak sú vaše dáta, resp. výsledky vhodné na vizualizáciu v ParaView, tak obrázky odtiaľ môžete ľahko exportovať. Platia podobné odporúčania, ako pri Mathematice a MATLAB-e. Ja tu pre všetky obrázky používam **File/Save Screenshot/PNG** (alebo aj cez ikonu fotoaparátu na lište nad oknom **RenderView**). V okne **Save Screenshot Options** je potom vhodné nastaviť si v **Coloring** pozadie na **Transparent Background** alebo **White Background**. Pred exportom je vhodné vypnúť **Show Orientation Axes** a nastaviť okno tak, aby nebolo príliš veľa pozadia. Viac sa o exportovaní obrázkov píše [tu na stránke ParaView](#). V ParaView boli vytvorené Obrázky 4.6.

**Inkscape.** [Inkscape](#) je výborný program na vytváranie vektorovej grafiky. Zíde sa, keď potrebujete kresliť schematické obrázky (nie grafy funkcií). Pracuje s formátom SVG, ale môžete

<sup>4</sup>PDF tu nefunguje dobre, vloží obrázok do stredu A4.

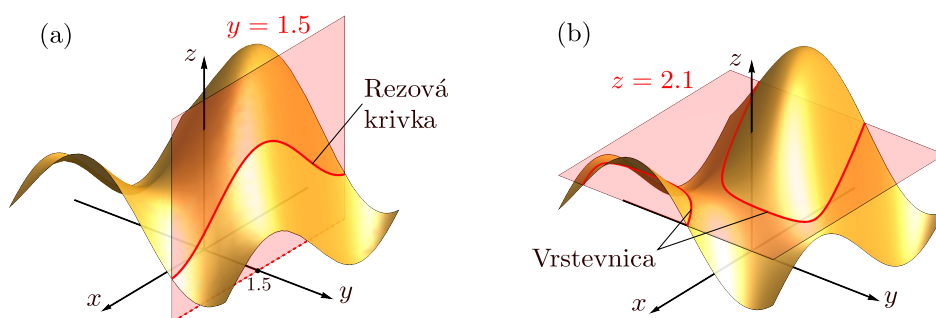


Obr. 4.10: Uloženie obrázka v MATLAB-e.

importovať/exportovať do mnohých formátov (aj PDF a EPS). Veľmi odporúčam doinštalovať [TeXText extension](#), aby ste do obrázkov mohli vpisovať LaTeX-ovské symboly a vzorce. Inkscape má [výborné tutorialy](#) (zrejme stačia Basic+Advanced), takže sa dá celkom rýchlo zvládnuť. V Inkscape bol vytvorený napríklad obrázok 4.3. Keď už máte niečo nakreslené, export sa urobí takto:

1. Označíte si, čo chcete exportovať.
2. `Ctrl+Shift+R` alebo `Edit/Resize Page to Selection`.
3. `Ctrl+Shift+S` alebo `File/Save As` a vybrať PDF.

**Mathematica/Matlab/ParaView + Inkscape** Ak do obrázka potrebujete doplniť popisky alebo vzorce, ktoré nedokážete dobre urobiť v Mathematice (resp. inom softvéri), často je praktickejšie na to použiť vhodný editor – napríklad Inkscape. Takouto kombinovanou technikou (Mathematica+Inkscape) bol vytvorený Obrázok 4.11 (po priblížení zistíte, čo bolo dorobené v Inkscape).



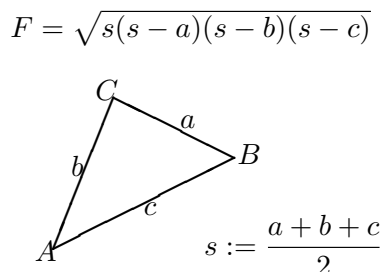
Obr. 4.11: Mathematica + dorobenie v Inkscape.

**Programovanie obrázkov v LaTeX-u.** Kvalitné obrázky sa dajú naprogramovať aj priamo v LaTeX-u. Nepoužívam to, ale sú ľudia, čo na tom frčia. Tu je niekoľko odkazov na túto možnosť:

- [Balík TikZ](#) je asi najlepší. [Na tejto stránke](#) nájdete veľké množstvo príkladov – pre predstavu, čo všetko a ako sa dá nakresliť.
- [Prostredie picture](#), príkladom je Obrázok 4.12.

- [Balík PSTricks](#)

Z niektorých programov na kreslenie obrázkov sa dá TeX-ovský kód generujúci obrázok exportovať (napríklad Inkscape → PSTricks).



Obr. 4.12: Obrázok vytvorený pomocou prostredia picture.

## 4.6 Tabuľky

Jednoduchá tabuľka bez popisu sa robí takto:

2	4	4.5
3	1	5
18	4.5	5

Tabuľky v záverečnej práci by ale mali mať popis, tu je príklad s popisom:

Tabuľka 4.1: The  $L^2$  EOC for the case with no redistribution.

$n_V$	$\tau$	$N$	$L^2$ error	EOC
122	0.04	2	1.43e-02	
482	0.01	8	3.35e-03	2.10
1922	0.0025	32	8.07e-04	2.05
7682	0.000625	128	1.95e-04	2.05
30722	0.00015625	512	4.64e-05	2.07

Ešte jeden príklad s použitím farby a definovaním šírky stĺpcov je v Tabuľke 4.2. Viac o tabuľkách nájdete aj s peknými príkladmi a zdrojovými kódmi [napríklad tu na wikibooks](#).

Tabuľka 4.2: Prehľad veličín.

Veličina	Jednotka	Názov	Popis
$u(x, t)$	K	Teplota v mieste $x$ a čase $t$	
$c(x)$	J/kg K	Merná tepelná kapacita	Množstvo tepla (J) potrebné na zohriatie 1 kg látky o 1 K
$\rho(x)$	kg/m <sup>3</sup>	Hustota	
$f(x, t)$	J/m <sup>3</sup> s	Intenzita (hustota) zdrojov tepla	Množstvo tepla (J) vyprodukované v 1 m <sup>3</sup> za 1 s
$q(x, t)$	J/m <sup>2</sup> s	Hustota tepelného toku	Množstvo tepla, ktoré prejde prierezom 1 m <sup>2</sup> za 1 s

**Dôležitý tip.** Vytvárať tabuľku priamo v zdrojovom kóde v LaTeX editore môže byť pri väčších tabuľkách dosť neprehľadné a neefektívne, odporúčam [tento výborný online nástroj](#). Môžete tam ľahko vytvoriť novú tabuľku, vložiť (copy-paste) existujúcu tabuľku (napríklad



z Excelu), či importovať z .csv, následne efektívne upravovať (perpisovať, meniť orámovanie, merge-ovať bunky, meniť farby,...), a nakoniec jedným klikom vygenerovať zodpovedajúci kód pre TeX-ovskú tabuľku, ktorý si vložíte do práce.

## 4.7 Vkládanie pseudokódu a kódu

V záverečných prácach na MPM je niekedy vhodné vysvetliť váš algoritmus alebo kód. Na písanie pseudokódov existuje veľa balíkov, prehľad nájdete [na stránke Overleaf-u](#) alebo [na wikibooks](#). Algoritmus 1 je príkladom použitia balíka `algorithm2e`.

---

**Algoritmus 1:** An algorithm with caption

---

```
Data:  $n \geq 0$ 
Result:  $y = x^n$ 
 $y \leftarrow 1$ ;
 $X \leftarrow x$ ;
 $N \leftarrow n$ ;
while  $N \neq 0$  do
  if  $N$  is even then
     $X \leftarrow X \times X$ ;
     $N \leftarrow \frac{N}{2}$ 
  else
    if  $N$  is odd then
       $y \leftarrow y \times X$ ;
       $N \leftarrow N - 1$ ;
    end
  end
end
```

---

Ak potrebujete do práce vložiť nejakú časť kódu, vodný je napríklad balík `listings`. Jednoduché použitie je tu:

```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Hello , World!");
    return 0;
}
```

Dá sa pohrať s formátovaním, nastaviť si farby, font a podobne. Nastavenie som vložil sem do kódu. Ak budete nejaké používať, dajte si ho do preambuly. Viac o vkladaní kódu sa dočítate v dokumentácii balíka `listings`.

---

```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

---



## Kapitola 5

### Záver

V závere zhrniete, čomu sa venovala vaša práca, ako sa vám podarilo naplniť stanovené ciele a k akým výsledkom ste prišli. Môžete zhodnotiť váš postup riešenia problému, jeho výhody/nevýhody. V závere je vhodné aj naznačiť, akými smermi by sa dalo v práci pokračovať, aké zostali nevyriešené otázky, kde vidíte možnosti vylepšenia a podobne.





# Bibliografia

1. EYMARD, R.; GALLOUËT, T.; HERBIN, R. Finite volume methods. In: *Solution of Equation in  $R^n$  (Part 3), Techniques of Scientific Computing (Part 3)*. Elsevier, 2000, zv. 7, s. 713–1018. Handbook of Numerical Analysis. ISSN 1570-8659. Dostupné z DOI: [https://doi.org/10.1016/S1570-8659\(00\)07005-8](https://doi.org/10.1016/S1570-8659(00)07005-8).
2. HANDLOVIČOVÁ, A.; TIBENSKÝ, M. *Základy funkcionálnej analýzy a variačného počtu*. Slovenská technická univerzita v Bratislave, 2016.
3. MIKULA, K.; REMEŠÍKOVÁ, M.; SARKOCI, P.; ŠEVČOVIČ, D. Manifold Evolution with Tangential Redistribution of Points. *SIAM Journal on Scientific Computing*. 2014, roč. 36, č. 4, A1384–A1414. Dostupné z DOI: [10.1137/130927668](https://doi.org/10.1137/130927668).