

Jihočeská univerzita  
v Českých Budějovicích  
Fakulta přírodovědecká

Ústav fyziky a biofyziky

# ZPRACOVÁNÍ LABORATORNÍCH MĚŘENÍ V SAGEMATH CLOUD

Bakalářská práce

**Autor:** Milan Somora

**Vedoucí práce:** Ing. Michal Šerý, Ph.D.

České Budějovice, 2017

## **Abstrakt**

Somora, M., 2017: Zpracování laboratorních měření v SageMath Cloud. [Processing of laboratory measurements in SageMath Cloud. Bc. Thesis, in Czech.] - 115 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

---

Cílem práce je představit potenciál webové služby CoCalc (dříve SageMath Cloud) a naučit v ní pracovat především studenty vysokých škol, aby dosáhli lepších výsledků při zpracování laboratorních měření. Uvedené informace a postupy se mohou aplikovat rovněž na dokumenty jiného zaměření.

## **Klíčová slova**

CoCalc, SageMath Cloud, Laboratorní protokol, Protokol o měření, Typografická pravidla, Tabulka, Graf, Zpracování naměřených dat, Jupyter Notebook, GNU Octave, Markdown, LaTeX

## Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejich internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce.

Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 7. prosince 2017

Milan Somora

## **Poděkování**

Rád bych poděkoval vedoucímu práce panu Ing. Michalovi Šerému, Ph.D za jeho cenné rady a připomínky a mé rodině za velkou trpělivost.

# Obsah

<b>1. Úvod</b>	<b>1</b>
<b>2. Co je CoCalc</b>	<b>3</b>
<b>3. Proč právě CoCalc?</b>	<b>7</b>
<b>4. Založení projektu</b>	<b>10</b>
4.1. Postup při založení projektu a popis prostředí . . . . .	10
4.2. Vytvoření souboru . . . . .	13
4.3. Orientace v projektu . . . . .	15
4.4. Práce se soubory a složkami . . . . .	16
4.5. Vyhledávání v projektu . . . . .	19
4.6. Zobrazení historie úprav . . . . .	20
4.7. Dodatečné nastavení projektu . . . . .	20
4.7.1. Název a popis projektu . . . . .	21
4.7.2. Spolupracovníci . . . . .	22
4.7.3. Dostupné kvóty a jejich využití . . . . .	23
4.7.4. Skrytí nebo vymazání projektu . . . . .	23
4.7.5. Restartování nebo zastavení projektu . . . . .	24
<b>5. Požadavky na laboratorní protokol</b>	<b>25</b>
5.1. Protokol o měření . . . . .	25
5.2. Všeobecné požadavky na protokol . . . . .	26
5.3. Osnova protokolu . . . . .	28
5.3.1. Titulní strana . . . . .	28
5.3.2. Zadání . . . . .	30
5.3.3. Teoretický úvod . . . . .	30
5.3.4. Použité přístroje a pomůcky . . . . .	31
5.3.5. Postup měření . . . . .	31

5.3.6. Vypracování . . . . .	32
5.3.7. Diskuse . . . . .	34
5.3.8. Závěr . . . . .	35
5.3.9. Použitá literatura . . . . .	35
5.3.10. Přílohy . . . . .	36
5.4. Pravidla pro tabulky . . . . .	37
5.5. Pravidla pro grafy . . . . .	39
<b>6. Typografická pravidla</b>	<b>43</b>
6.1. Všeobecná pravidla . . . . .	43
6.2. Pravidla pro zápis matematických výrazů . . . . .	47
6.3. Pravidla pro zápis fyzikálních veličin a jejich jednotek . . . . .	49
<b>7. Sázení matematických rovnic v jazyku Latex</b>	<b>52</b>
7.1. Matematické operace . . . . .	53
7.2. Matematické symboly . . . . .	54
7.3. Symboly řecké abecedy . . . . .	55
7.4. Ostatní příkazy . . . . .	55
7.5. Příklad . . . . .	55
<b>8. Zpracování naměřených hodnot v Jupyter Notebooku</b>	<b>57</b>
<b>9. Základy jazyka Octave</b>	<b>62</b>
9.1. Základní příkazy . . . . .	63
9.2. Skalární operace . . . . .	66
9.3. Operace s řádkovými vektory . . . . .	66
9.4. Zápisy matematických rovnic . . . . .	67
9.5. Příklad zpracování naměřených hodnot . . . . .	68
9.6. Vytvoření 2D grafu . . . . .	70
9.6.1. Zobrazení hodnot pomocí hodnot z vektorů . . . . .	70
9.6.2. Zobrazení pomocí matematické funkce . . . . .	74
9.6.3. Zobrazení aproximace naměřených dat . . . . .	76
<b>10. Vytvoření dokumentu v editoru CoCalc</b>	<b>79</b>
10.1. Markdown . . . . .	81
10.2. L <sup>A</sup> T <sub>E</sub> X . . . . .	85

<b>11. Další možnosti v CoCalc</b>	<b>87</b>
11.1. Týmová práce . . . . .	87
11.2. Možnosti výuky . . . . .	88
11.3. Dokumentace a podpora . . . . .	91
<b>12. Shrnutí a závěr</b>	<b>92</b>
<b>Použitá literatura</b>	<b>92</b>
<b>Přílohy</b>	<b>95</b>
<b>A. Dodatečné značky a příkazy pro Markdown a HTML</b>	<b>96</b>
<b>B. Graf 1: Millikanův experiment</b>	<b>98</b>
<b>C. Graf 2: Franckova - Hertzova křivka</b>	<b>100</b>
<b>D. Graf 3: Aktivita Barya-137</b>	<b>103</b>
<b>E. Aproximace metodou nejmenších čtverců</b>	<b>106</b>

# 1. Úvod

Hlavním cílem této práce je představit potenciál webové služby CoCalc (dříve SageMath Cloud<sup>1</sup>) a naučit v ní pracovat především studenty vysokých škol, aby dosáhli lepších výsledků při zpracování laboratorních měření.

Důvodem byla osobní zkušenost, že studenti při zpracování laboratorních měření využívají obvykle více nástrojů najednou a jak se v praxi ukázalo, tento způsob celkovou situaci spíše komplikuje a současně generuje značné množství chyb, což přispívá k časové tísně a někdy také k horším studijním výsledkům. Z tohoto pohledu se domníváme, že by služba CoCalc mohla být pro studenty přínosem, neboť poskytuje profesionální vědecké nástroje, které zajišťují vzájemnou kompatibilitu, jsou zdarma a přístupné téměř z každého webového prohlížeče na světě.

Jelikož informace a postupy uvedené v této práci se mohou aplikovat také na dokumenty jiného zaměření (technické zprávy, manuály, vědecké, diplomové a seminární práce, zpracování různých dat atd.), není tato práce vyhraněna pouze pro studenty<sup>2</sup>, ale může z ní vycházet každý, komu přinese užitek. Např. učitelé, vědci nebo zaměstnanci firem z jakéhokoli odvětví (elektrotechnika, chemie, biologie, strojírenství ad.).

Kapitoly byly vypracovány s ohledem na využití v praxi a pokud možno bez zbytečného balastu. Práce by se dala rozdělit na čtyři části.

Nejprve je čtenář seznámen se službou CoCalc a jejími možnostmi. Naučí se v ní založit projekt, umět se v něm orientovat a pracovat se soubory a složkami.

Ve druhé části jsou mu vysvětleny pojmy Laboratorní protokol, Protokol o měření a typografická pravidla, která úzce souvisí s estetikou laboratorního protokolu. Rovněž je mu vysvětleno, jak v protokolu správně zapisovat matematické

---

<sup>1</sup>k přejmenování služby došlo v průběhu psaní této práce

<sup>2</sup>a proto bude dále používáno slovo čtenář



rovnice, fyzikální veličiny a jejich jednotky s využitím jazyka  $\text{\LaTeX}$ .

Třetí část pojednává o tom, jak zpracovávat naměřená data a vytvářet grafy pomocí nástroje Jupyter Notebook s využitím základů skriptovacího jazyka Octave. Poslední část popisuje, jak vypracovat laboratorní protokol v editoru CoCalc s využitím značkovacího jazyka Markdown a již zmíněného jazyka  $\text{\LaTeX}$ .

Součástí této práce jsou tři vypracované laboratorní protokoly z jaderné a částicové fyziky a další příklady, které lze v případě potřeby využít jako návod. Tento obsah byl však kvůli velkému rozsahu uložen na disk CD, který je součástí tohoto výtisku.

Při zpracování byla věnována pozornost pouze těm funkcím CoCalc, které byly nutné pro zpracování laboratorního měření. Přesto práce nabyla značného rozsahu, neboť se jedná o poměrně obsáhlou a relativně obtížnou problematiku. Současně je nutné upozornit, že je služba CoCalc vyvíjena rychlým tempem, tudíž se mohou některé prvky na panelech lišit nebo další funkce přibývat.

Bude-li tato práce přínosem, a to nejen studentům či učitelům, ale také např. formou ušetření nákladů za drahý software ve firmách, efektivnějšího postupu při zpracování technických dokumentů, použití CoCalc ve školách apod., bude její úkol splněn.

## 2. Co je CoCalc

CoCalc je webová služba, ve které lze on-line:

- provádět *matematické výpočty*, které pokrývají velkou oblast matematiky - např. matematickou analýzu, statistiku, algebru, kombinatoriku, kryptografii, numerickou matematiku, teorii čísel, algoritmy pro strojové učení, symbolické a statistické zpracování přirozeného jazyka a další
- vytvářet *odborné dokumenty* v jazyku  $\text{\LaTeX}$ , Markdown a HTML - např. laboratorní protokoly, technické zprávy, vědecké, diplomové či seminární práce atd.
- vytvářet *soukromé projekty*
- *spolupracovat* s dalšími lidmi na projektu v reálném čase
- *vyučovat* - umožňuje vytvořit kurz, přidat do něj studenty, vytvořit pro ně úkoly, kontrolovat a klasifikovat jejich práci.



Hlavní ideou služby CoCalc je poskytnout *profesionální vědecké nástroje*, které jsou zdarma a přístupné on-line z jakéhokoli webového prohlížeče na světě. Čili

není nutné kupovat drahý profesionální software ani dodatečně něco instalovat. Všechny nástroje jsou dostupné na jednom místě a díky *cloudu* je možné práci uložit a kdykoli v ní pokračovat. CoCalc lze využít rovněž k výukovým účelům, což může být výhodné na některých školách. Za vývojem stojí společnost SageMath, Inc., jejíž zakladatelem je americký profesor matematiky William A. Stein, který v roce 2005 vytvořil matematický software SageMath (dříve Sage).

CoCalc je služba typu *open-source*<sup>1</sup>, čili je poskytována *zdarma* a vydělává pouze na souvisejících službách. Snaží se konkurovat placeným alternativám typu Matlab, Mathematica, Magma a Maple, ve skutečnosti je však mocnější, neboť využívá více nástrojů než všichni uvedení konkurenti dohromady.



Obrázek 2.1.: Nástroje podporované službou CoCalc

Po registraci získáme *bezplatný účet*, který poskytuje 1 GB paměti, 1 sdílené jádro CPU a 3 GB prostoru v *cloudu*, což je pro běžného uživatele dostačující. Náročnější uživatelé mohou za menší příplatek získat větší výkon, větší rychlost serveru, více úložného prostoru v *cloudu* ad. Při psaní této práce vystačil bezplatný účet.

CoCalc je sestavena ze stovky nástrojů, které byly vytvořeny rovněž z open-

---

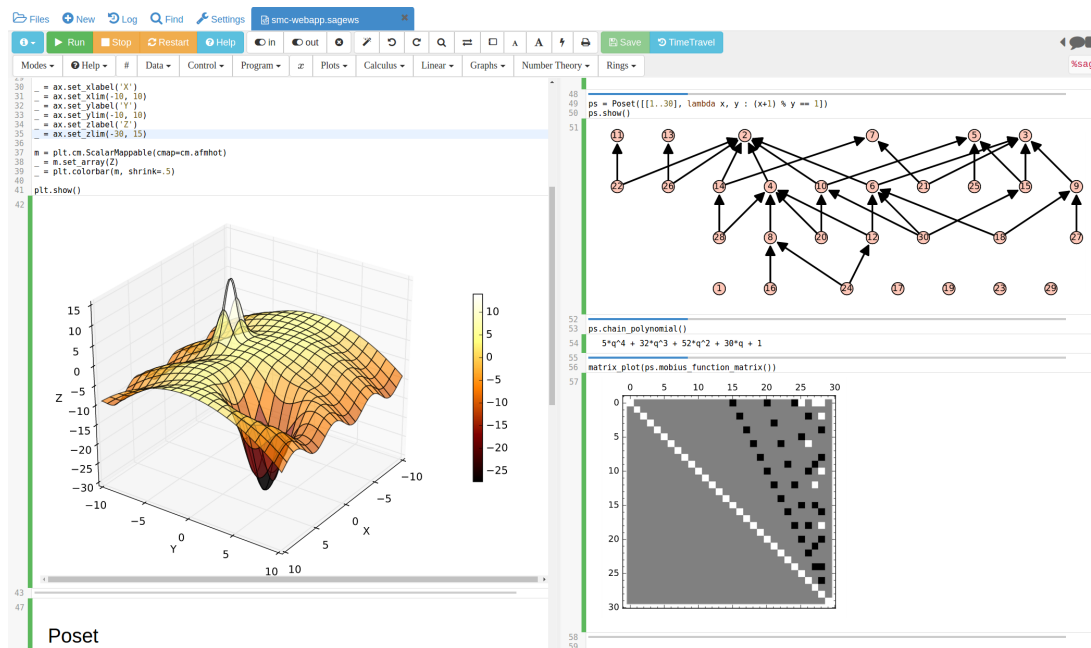
<sup>1</sup>Open-source software je označován za svobodný software poskytovaný zdarma, který vyvíjí komunita složená z dobrovolníků a jehož kód je volně přístupný dalším vývojářům, aby jej mohli studovat, případně vylepšovat. Open-source software může vydělávat na souvisejících službách, nikoliv na prodeji samotného softwaru.

source balíčků. Jak bylo uvedeno, tyto nástroje umožňují nejrůznější výpočty, modelování a mnoho dalších možností. Zde jsou jedny z nejpoužívanějších:

- **SageMath** - silný matematický software, který tvoří jednu z nejsilnějších stránek CoCalc. Využívá jazyk Python a balíčky jako jsou **Statsmodels** pro vytváření statistických modelů, **Pandas** pro analýzu dat, **SymPy** pro symbolickou matematiku, **Scikit Learn** k vytváření algoritmů pro strojové učení ad.

SageMath lze rovněž nainstalovat na počítač. Na internetu má poměrně velkou uživatelskou základnu, tudíž není těžké dohledat požadovanou informaci. Pro zájemce o tuto problematiku jsou v sekci 11.3. uvedeny odkazy na literaturu k dodatečnému nastudování.

- **R** - nejlepší open-source software pro statistiku.
- **Vědecký Python** - podobně jako konkurenční program MATLAB využívá pro matematické výpočty svůj skriptovací jazyk, CoCalc využívá jazyk Python přizpůsobený pro vědecké účely.
- **Julia** - programovací jazyk pro numerické výpočty.
- **GNU Octave** - skriptovací jazyk do velké míry kompatibilní s komerčním jazykem MATLAB.
- **Tensorflow** - softwarová knihovna pro umělou inteligenci.
- **LaTeX** - program, který umožňuje sázet text a tisknout svá díla ve velmi vysoké typografické kvalitě s předdefinovanými vzhledy dokumentu.
- **GNU/Linux** - CoCalc běží na systému Linux, jehož terminál lze plně využívat.
- **Jupyter Notebook** - interaktivní výpočetní prostředí, ve kterém lze kombinovat spustitelný programový kód, formátovaný text, matematiku a grafy. Lze v něm např. provádět výpočty a vytvářet grafy pomocí jazyka Octave. Ve skutečnosti podporuje až 40 programovacích jazyků.



Obrázek 2.2.: Prostředí CoCalc

## Další přednosti CoCalc

- Každou hodinu *zálohuje* soubory pro případ jejich poškození nebo náhodnému smazání.
- Detailně *zaznamenává* změny - umožňuje procházet historii ve stovkách krocích směrem dozadu a dopředu, tzv. Time Travel.
- Umožňuje integraci jazyka C/C++ a Fortranu.

## 3. Proč právě CoCalc?

Tato kapitola pojednává o tom, proč není dobré zůstat u starých způsobů. Jak bylo naznačeno v úvodu, studenti vysokých škol při zpracování laboratorního měření *kombinují* více nástrojů najednou. Obvykle se jedná o nástroje typu MS Word a Excel, Octave, Matlab, SciLab, Gnuplot, L<sup>A</sup>T<sub>E</sub>X, cloudová úložiště typu DropBox, Google Drive, OneDrive ad. Z vlastních zkušeností<sup>1</sup> můžeme potvrdit, že studenti při zpracování laboratorních protokolů byli vystaveni extrémní *časové tísní*, kterou způsobovala především nedostatečná funkcionalita programů a jejich vzájemná nekompatibilita. Zde je výčet některých problémů:

### Vytváření protokolu v MS Word

- MS Word je z našeho pohledu pro zpracování laboratorních protokolů neprofesionální editor, protože se v něm obtížně udržuje číslování obrázků, tabulek, vzorců a křížových odkazů.
- Tabulky často nešly nastavit podle požadavků a někdy bylo náročné udržet jednotný vzhled obsáhlejšího protokolu<sup>2</sup>.
- Studenti, kteří neměli vypnuté automatické opravy ve Wordu často přehlédli slova nebo znaky, které Word automaticky nahradil bez předchozího upozornění - např. znaménko mínus za delší pomlčku.
- Grafy importované z Excelu někdy ztratily svůj původní formát.
- Celkový vzhled dokumentu působil amatérským dojmem.

---

<sup>1</sup>Autor této práce absolvoval fyzikální praktika na Přírodovědecké fakultě Jihočeské univerzity v Č. Budějovicích a vychází z vlastních zkušeností.

<sup>2</sup>jednotný vzhled tabulek, umístění obrázků, jejich číslování, stejné fonty, jednotný formát vzorců, dodržování mezer, stejné odsazení odstavců atp.

## Zpracování naměřených dat v MS Excelu

- Při výpočtech docházelo k různým chybám, které byly způsobeny především automatikou Excelu. Student musel nastavit správné formátování buněk, aby nedocházelo třeba k automatickému zaokrouhlování čísel.
- Opakovaně docházelo k situaci, kdy Excel automaticky nastavil řád exponentu a podle něho posouval desetinnou čárku. Např. když bylo nutné z čísla  $0,00000581964 \times 10^{-3}$  udělat číslo  $5,81964 \times 10^{-9}$ , muselo se to provést manuálně. Za povšimnutí také stojí, že některé automatické funkce v Excelu ovlivňovaly celkový výsledek měření, zatímco výsledek na studentské kalkulačce byl správný.
- Když bylo nutné vrátit se k výpočtům o několik dní později, stálo hodně času zorientovat se v propojení mezi buňkami, a někdy bylo nutné výpočet nastavit od začátku.

## Vytváření grafů

- V Excelu občas nešly nastavit rozsahy a popisy os grafu na základě požadavků. Vzhled grafů působil rovněž neprofesionálně.
- Program GNU Octave, který měli studenti nainstalovaný v počítači, někdy nebyl stabilní (docházelo k pádům) - student tedy musel vyřešit problém v jiném open-source programu (např. v Gnuplotu nebo SciLabu). Z velké části záviselo na tom, jakou verzi programu student nainstaloval, jak ji nainstaloval a jaký operační systém v počítači používal.

Komunikace mezi studenty, která probíhala pomocí služby DropBox, byla často komplikovaná - dokumenty se musely stáhnout, upravit v počítači, potom zpět nahrát na DropBox. Když se upload nezdařil, museli ostatní čekat, než dotyčný akci zopakuje.

Z výše uvedeného vyplývá, že studenti poměrně často museli:

- *opravovat chyby* způsobené programem
- *řešit problémy* vzniklé při exportu a importu z jednoho programu do druhého
- *dodělávat práci v jiném programu*, jelikož ten předchozí to neumožňoval.

A proto se domníváme, že by služba CoCalc mohla být pro studenty přínosem z následujících důvodů:

- nabízí *profesionální nástroje* určené k vědeckým účelům
- sdružuje veškerou práci na *jednom místě*
- zaručuje *propojenost a kompatibilitu* mezi jednotlivými nástroji
- díky jazyku  $\text{\LaTeX}$  a Markdown umožňuje získat *jednotný formát* a *profesionální vzhled* dokumentu.



## 4. Založení projektu

CoCalc poskytuje uživatelsky přívětivé rozhraní zaměřené na jednoduchost. Vše se ovládá z webového prohlížeče. Jeden z nejdůležitějších pojmů v CoCalc je *projekt*. Postup při zpracování laboratorních měření by se dal popsat ve třech krocích:

1. Založit projekt
2. Zpracovat naměřená data a vytvořit grafy v Jupyter Notebooku pomocí jazyka Octave.
3. Vytvořit laboratorní protokol v editoru CoCalc pomocí jazyka Markdown a  $\text{\LaTeX}$ .

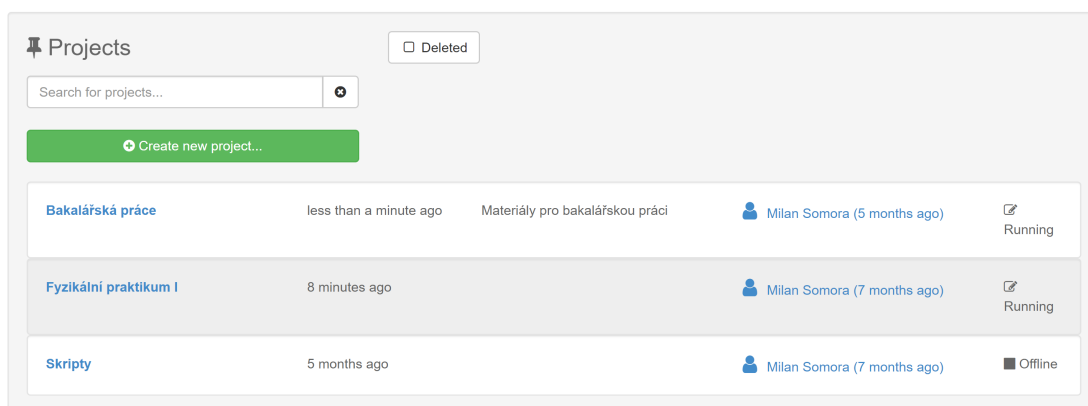
Tato kapitola seznamuje čtenáře s prvním krokem, tedy jak založit projekt a jak v něm pracovat. Dalším krokům bude věnována pozornost v jiných kapitolách.

### 4.1. Postup při založení projektu a popis prostředí

Po přihlášení<sup>1</sup> se zobrazí *úvodní stránka*, na které se budou později zobrazovat chronologicky seřazené vytvořené projekty (jako na obrázku 4.1). Po založení projektu (kliknutím na zelené tlačítko `Create new project`), např. pod názvem Franckův - Hertzův experiment, dojde k automatickému přesměrování na *záložku projektu* (obr. 4.2), ve které se budou vytvářet a upravovat soubory týkající se laboratorního protokolu.

---

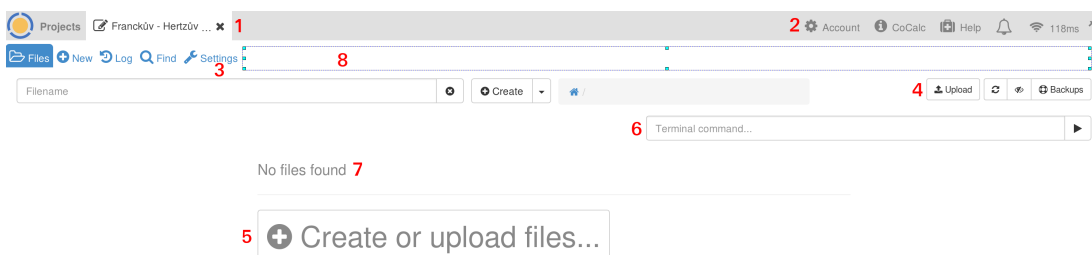
<sup>1</sup>je nutná předchozí registrace



Obrázek 4.1.: Úvodní stránka po přihlášení do CoCalc




## Panely a ikony


Viz čísla na obrázku 4.2.



Obrázek 4.2.: Hlavní složka projektu






- Po celou dobu činnosti v CoCalc je v záhlaví stránky umístěn *Panel záložek* (1), na kterém se zobrazují záložky otevřených projektů a rovněž se na něm nachází *systemový panel* (2), který obsahuje tato tlačítka:
  - Account - zobrazí nastavení účtu (profil uživatele, předplatné, nastavení služby CoCalc, odhlášení ze služby atd.)
  - CoCalc - zobrazí souhrnné informace o CoCalc
  - Help - uživatelská podpora a dokumentace pro CoCalc

-  - zobrazí historii všech vytvořených souborů v CoCalc (ze všech projektů). Seznam je automaticky doplňován dokud není smazán. Vyhledávací pole umožňuje v seznamu vyhledávat.
-  - zobrazí časovou odezvu vzdáleného serveru (tzv. ping)
-  - skryje panely (1), (2) a (3)





Na panelu záložek se nachází ještě tlačítko , které vrací zpět na úvodní stránku založených projektů.

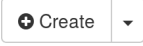



Každá *aktivní* záložka projektu obsahuje tato tlačítka:

- **Funkční panel (3)**

-  **Files** - zobrazí soubory a složky projektu
-  **New** - vytvoří soubor volitelného typu
-  **Log** - zobrazí historii úprav v projektu (včetně jmen spolupracovníků, jsou-li do projektu zapojeni)
-  **Find** - vyhledá soubory, složky nebo řetězec v souboru
-  **Settings** - umožňuje dodatečné nastavení projektu (změna názvu, přidání spolupracovníků, smazání projektu ad.)

- **Funkční panel (4):**

-  **Upload** - nahraje libovolný externí soubor do projektu
-  - aktualizuje *hlavní složku* projektu (kořenový adresář)
-  - zobrazí skryté soubory
-  **Backups** - zobrazí seznam záloh projektu, které CoCalc provádí automaticky zhruba každou hodinu

-  - vytvoří soubor volitelného typu (ekvivalent k tlačítku  **New**)
-  - vrací zpět do hlavní složky projektu
- Create or upload files... (5) - vytvoří soubor volitelného typu nebo nahraje libovolný soubor (ekvivalent k tlačítku  **New**). Toto tlačítko se objevuje, pouze když projekt neobsahuje žádné soubory ani složky.

- Terminal command... (6) - jak bylo uvedeno, CoCalc běží na systému Linux, jehož terminál lze plně využívat. Toto je „rychlá příkazová řádka“, do které je možné zadávat příkazy Linuxu. Uživatel se může např. dozvědět, jaká verze Octave je v CoCalc využívána (viz obrázek 4.3).
- Oblast, ve které se budou zobrazovat vytvořené soubory a složky je na obrázku 4.2 vyznačena číslem (7). Na obrázku je ještě oblast (8), resp. lišta, na které se budou zobrazovat záložky s otevřenými soubory (viz dále).

```

octave -v
octave: X11 DISPLAY environment variable not set
octave: disabling GUI features

GNU Octave, version 4.2.1
Copyright (C) 2017 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.

Octave was configured for "x86_64-pc-linux-gnu".

Additional information about Octave is available at http://www.octave.org

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.

```

Obrázek 4.3.: Využití vzdáleného terminálu - zjištění verze Octave

## 4.2. Vytvoření souboru

Vytvoření souboru v CoCalc znamená spuštění nějakého *nástroje*. K dispozici je mnoho typů a každý slouží k jinému účelu. V prvním nástroji je možné zpracovávat naměřená data, ve druhém vytvářet laboratorní protokol a ve třetím např. chatovat se spolupracovníky zapojenými do projektu. Nástroje (resp. soubory), které budou využívány v našem projektu, jsou následující:

## Jupyter Notebook

Tzv. zápisník Jupyter. Velmi užitečný nástroj, ve kterém lze např. zpracovávat naměřené hodnoty. Po vytvoření souboru s příponou (.ipynb) se otevře stránka s buňkami, do kterých lze psát spustitelný programový kód či prostý text, který lze dále formátovat pomocí jazyků Markdown a  $\text{\LaTeX}$ . V této práci je mu věnována zvláštní sekce 8 na straně 57.

## Soubory typu Markdown a $\text{\LaTeX}$

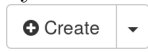
Oba soubory slouží pro vytváření dokumentu. Po vytvoření souboru s příponou (.md) se otevře editor se dvěma okny, ve kterém je možné vytvářet dokument v jazyku Markdown. Obdobně po vytvoření souboru s příponou (.tex) lze vytvářet dokumenty v jazyku  $\text{\LaTeX}$ . Více je o nich pojednáno v sekci 10 na straně 85.

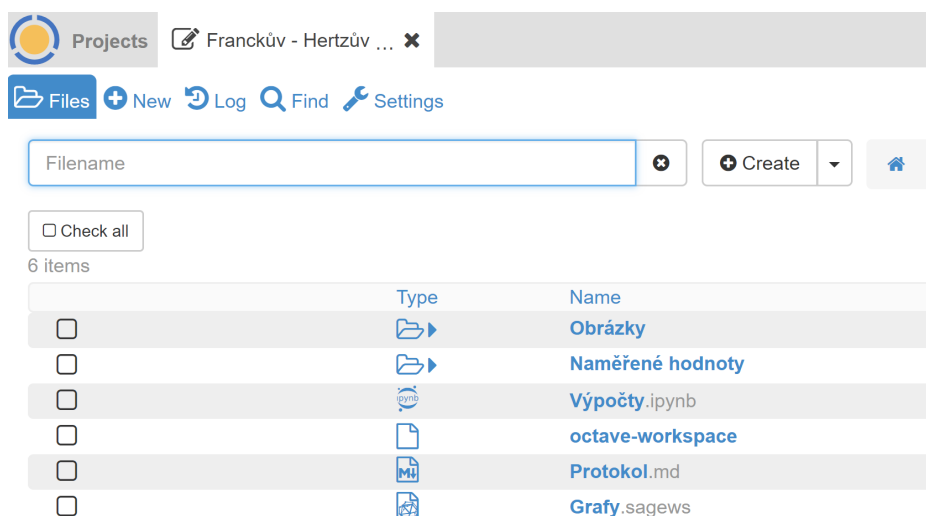
## Soubory Task list a Chat

Soubor typu Task list s příponou (.tasks) je vlastně TO-DO list, do kterého lze zadávat úkoly související s projektem. Soubor typu Chat s příponou (.sage-chat), jak bylo již naznačeno, otevře stránku, na které lze chatovat se všemi účastníky zapojenými do projektu. Více v 11.1 na straně 87.

## Folder

Vytvoří adresář nebo-li složku. Složky umožňují zpřehlednit projekt (obr. 4.4).

V předchozí sekci byla popsána tři ekvivalentní tlačítka, kterými lze vytvořit soubor požadovaného typu. Doporučujeme používat tlačítko , které zobrazuje nabídku nejčastěji používaných souborů. Do příslušného pole se napíše název souboru a ze seznamu se vybere požadovaný typ souboru (viz obrázek 4.5). Poté automaticky dojde ke spuštění nástroje a soubor se uloží do hlavní složky projektu (viz dále). Názvy souborů a složek se doporučuje psát bez diakritiky.



Obrázek 4.4.: Hlavní záložka projektu - soubory a složky

Více typů souborů lze získat kliknutím na tlačítko **+ New** (viz obr. 4.6). Např. soubor s příponou (.c++) umožňuje zápis kódu v jazyku C++ včetně zvýrazňování syntaxe.

### 4.3. Orientace v projektu

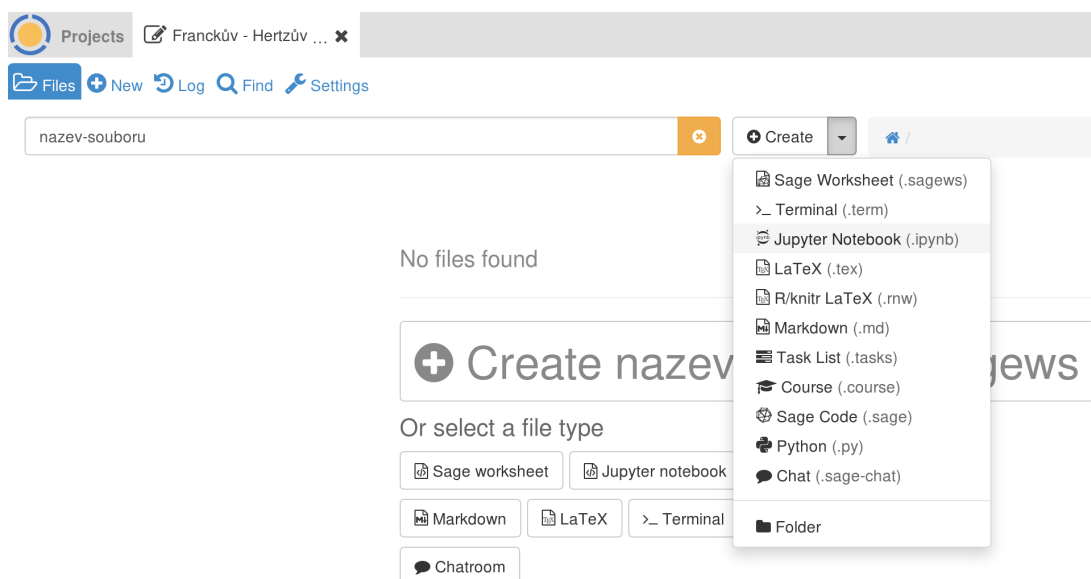
Předpokládejme, že již máme vytvořeny nějaké soubory a složky, které tvoří podobnou strukturu jako na obrázku 4.4. Kliknutím na jakýkoli soubor se otevře záložka a v ní náležitý obsah<sup>2</sup>. Pro lepší názornost je uveden obrázek 4.7, na kterém je otevřeno více záložek najednou.

Na *šedém* panelu (1) jsou otevřeny projekty a na *bílém* (2) jsou otevřeny soubory. Manipulace se záložkami je stejná jako v běžném internetovém prohlížeči<sup>3</sup>.

Je-li to možné, zkuste nyní sledovat ikonu domečku 🏠. Za touto ikonou je zobrazena *cesta* (3), ve které složce se právě nacházíme. Do adresářové cesty je možné klikat a tím se dostat do adresáře o úroveň níže. Kliknutím na ikonu

<sup>2</sup>takto lze otevřít i obrázky, PDF dokumenty ad.

<sup>3</sup>přepínání mezi záložkami, pracovat v každé zvlášť, přesouvání tahem myši, zavírání křížkem na pravé straně apod.




Obrázek 4.5.: Vytvoření souboru

domečku se vrátíme do hlavní složky projektu, resp. do kořenového adresáře. Obdobně kliknutím na ikonu [Files](#) budeme přesměrováni do složky, ve které jsme se nacházeli naposled.

## 4.4. Práce se soubory a složkami

### Přesouvání, kopírování, mazání ad.

Je-li soubor vybrán (označen), zobrazí se ovládací panel (obr. 4.8), pomocí kterého lze soubor *stáhnout*, *smazat*, *přejmenovat*, *duplikovat*, *přesunout* či *zkopírovat* (i do jiného projektu) a *sdílet*. Je-li vybráno více souborů nebo složek najednou, lze je *komprimovat* do souboru ZIP.

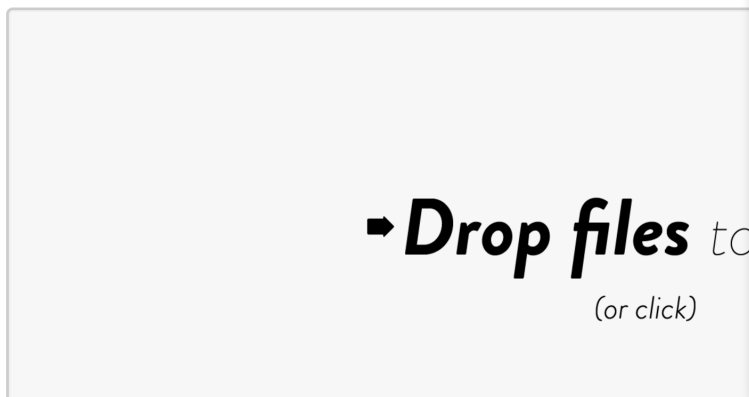
Použití je natolik snadné, že nemá smysl vytvářet podrobný návod. Pouze doplníme, že soubor je možné stáhnout také kliknutím na ikonu , která se nachází napravo.

Name your file, folder or paste in a link

Select the type



- Stex (.bbl)
- Makefile (.build)
- Archive (.bz2)
- C (.c)
- Csharp (.c#)
- C++ (.c++)
- Text/Cjsx (.cjsx)
- Clojure (.clj)
- Coffeescript (.coffee)
- Nginx (.conf)
- Course (.course)
- Sql (.cql)
- CSS (.css)
- Diff (.diff)
- Application/XML-Dtd (.dtd)
- Eiffel (.e)
- Ecl (.ecl)

Drag and drop files.

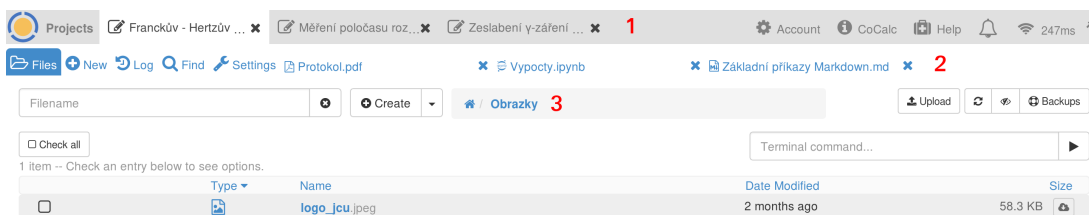


Obrázek 4.6.: Vytvoření jiných typů souborů

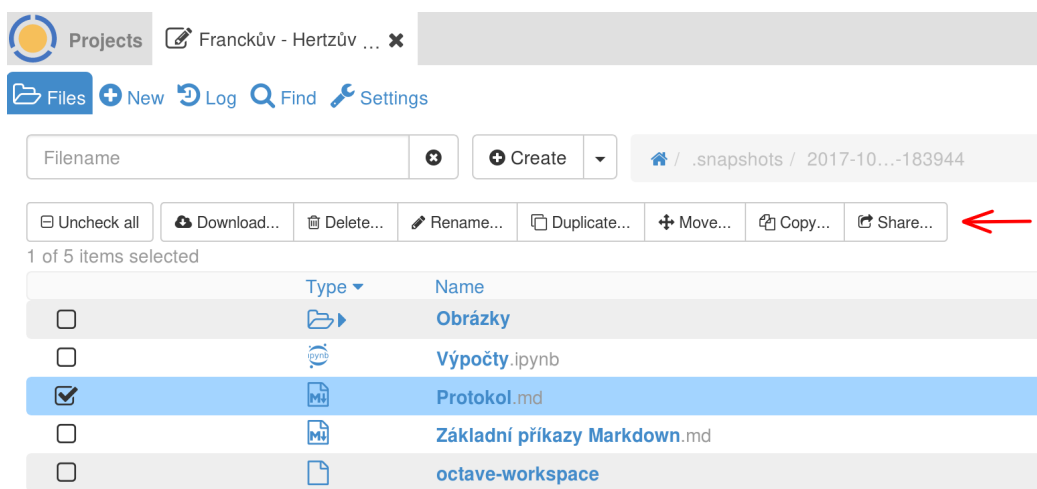
## Nahrání externího souboru

Do projektu je možné nahrát soubor *libovolného* typu. Nahrávání lze provést dvěma způsoby. Nejrychlejší způsob je použití tlačítka  Upload. Druhým způsobem je použití tlačítka  New. Po kliknutí na toto tlačítko se zobrazí stránka s *rámcem* (obr. 4.9), do kterého lze soubory jednoduše *přetahovat*.





Obrázek 4.7.: Otevřené záložky v projektu



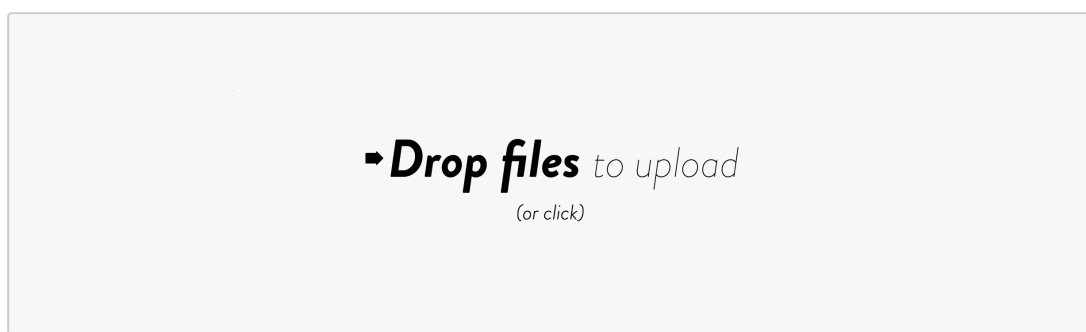
Obrázek 4.8.: Ovládací panel pro práci se soubory

## Sdílení

CoCalc umožňuje sdílet soubory a složky pomocí:


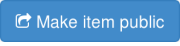


- webového odkazu
- emailu
- sociálních sítí

Uživatelé, kterým je sdílen obsah, nemusí mít u CoCalc účet. Sdílený obsah lze prohlížet, zobrazovat, stahovat, ale není možné jej upravovat. Pro úpravy je nutná registrace v CoCalc a role spolupracovníka (viz [11.1](#)).



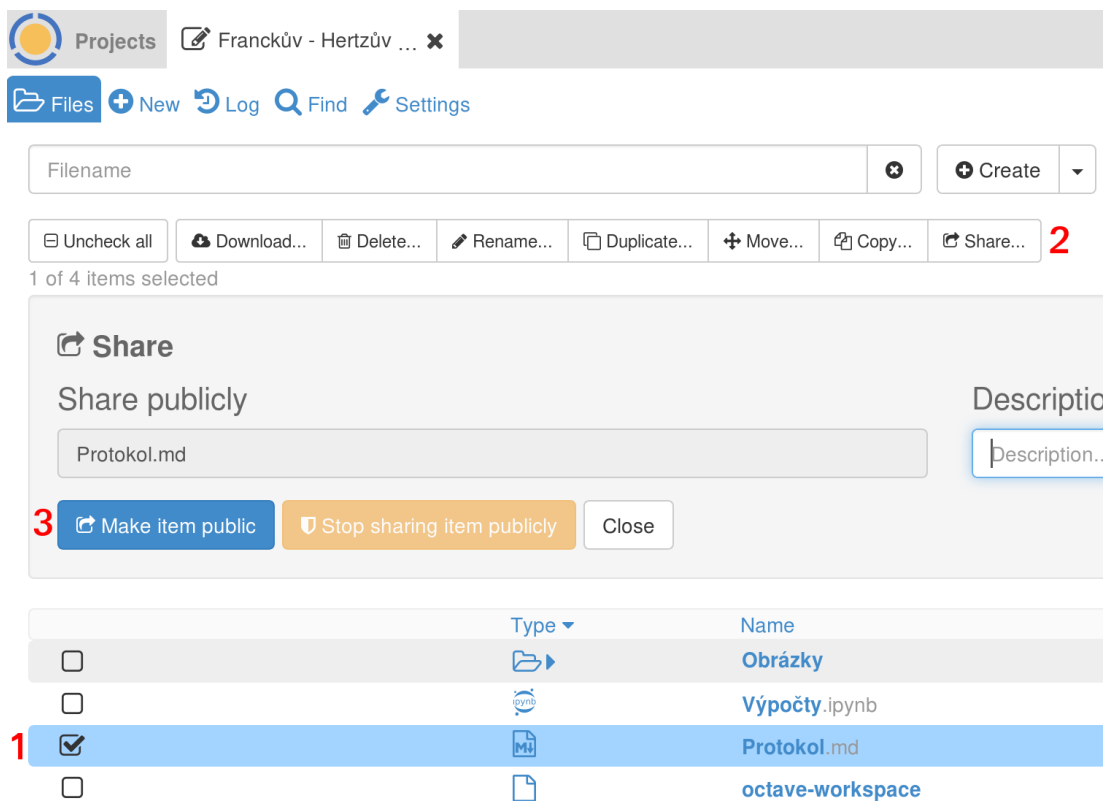
Obrázek 4.9.: Rámec pro nahrávání souborů způsobem Drag and drop

### Postup při sdílení (viz obr. 4.10):

1. Označit soubor nebo složku „fajfkou“.
2. Kliknout na tlačítko .
3. Soubor nebo složku zveřejníme kliknutím na .
4. CoCalc vygeneruje odkaz, kterým lze sdílet položku prostřednictvím mailu, sociálních sítí nebo jej můžeme zkopírovat a poslat např. Messengerem (obr. 4.11).
5. Sdílený soubor nebo složka získají v seznamu ikonu Public  (obr. 4.12), aby bylo možné rozlišit, které položky jsou sdíleny.
6. Sdílení lze zrušit opakováním bodů 1 a 2 a kliknutím na tlačítko .


## 4.5. Vyhledávání v projektu

CoCalc umožňuje vyhledávat zadaný řetězec v celém projektu, a to včetně obsahů souborů. Způsob vyhledávání nejlépe demonstruje obrázek 4.13. Vyhledávání umožňuje zahrnout další podmínky, např. vyhledání skrytých souborů.




Obrázek 4.10.: Postup při sdílení souboru

## 4.6. Zobrazení historie úprav

CoCalc podrobně zaznamenává kdo, co a kdy na projektu dělal. Tuto aktivitu lze prohlížet v tzv. Project Logu, který lze zobrazit kliknutím na ikonu  Log.





## 4.7. Dodatečné nastavení projektu

Kliknutím na ikonu  Settings se zobrazí stránka *Settings and configuration*, na které lze projekt dodatečně nastavit. Pro naše potřeby jsou důležité tyto funkce:

Public access link

<https://cocalc.com/projects/e2de7155-2d03-4f14-8702-f8eff1f84f27/files/Protokol.md>

Obrázek 4.11.: Vygenerovaný odkaz pro sdílení souboru

	Type ▾	Name
<input type="checkbox"/>		<b>Obrázky</b>
<input type="checkbox"/>		<b>Výpočty.ipynb</b>
<input checked="" type="checkbox"/> Public		<b>Protokol.md</b>
<input type="checkbox"/>		<b>octave-workspace</b>

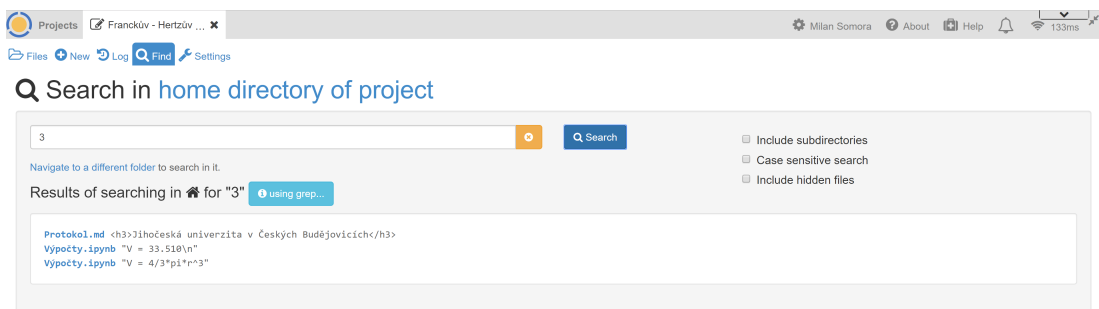
Obrázek 4.12.: Označení souboru ikonou Public

### 4.7.1. Název a popis projektu

**H** Title and description

Title

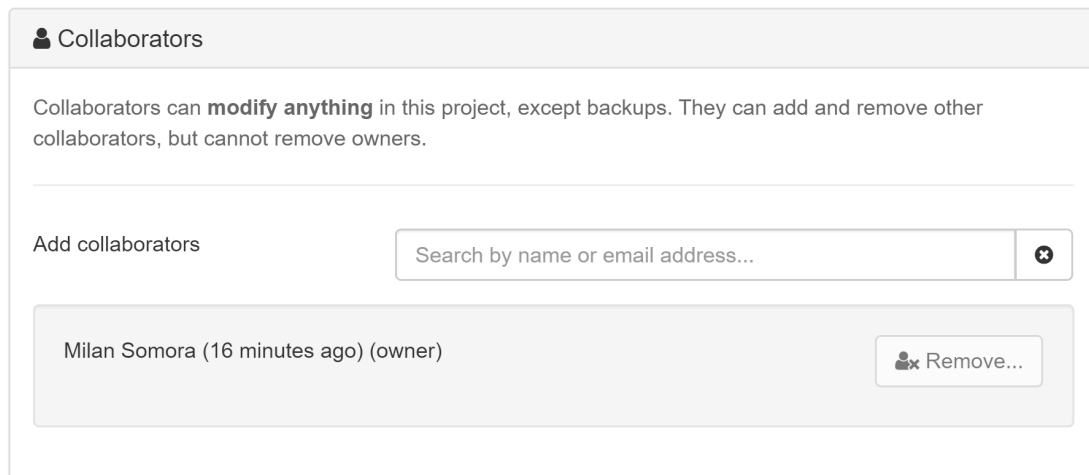
Description



Obrázek 4.13.: Vyhledávání znaku „3“ v celém projektu

## 4.7.2. Spolupracovníci

Umožňuje přidat další spolupracovníky do projektu.



### 4.7.3. Dostupné kvóty a jejich využití

Zobrazuje využití dostupného prostoru v Cloudu, využití paměti RAM, počet přidělených jader CPU atd. V případě zakoupení předplatného dojde k navýšení kvót.

Project usage and quotas	
	<a href="#">Adjust your quotas...</a>
Member hosting	<b>No</b>
Internet access	<b>Yes</b> <ul style="list-style-type: none"><li>1 project given by Milan Somora</li></ul>
Idle timeout	<b>0.5 hours</b> of non-interactive use before project stops <ul style="list-style-type: none"><li>0.5 hours given by free project</li></ul>
Disk space	<b>3000 MB</b> disk usage limit ( <b>1 MB</b> used) <ul style="list-style-type: none"><li>3000 MB given by free project</li></ul>
Shared RAM	<b>1000 MB</b> shared RAM memory limit ( <b>69 MB</b> used) <ul style="list-style-type: none"><li>1000 MB given by free project</li></ul>
Dedicated RAM	<b>0 MB</b> dedicated RAM
Shared CPU	<b>1 core</b> <ul style="list-style-type: none"><li>1 core given by free project</li></ul>
Dedicated CPU	<b>0 cores</b>

### 4.7.4. Skrytí nebo vymazání projektu

Skryje<sup>4</sup> projekt na úvodní stránce projektů nebo jej celý smaže.




Hide or delete project	
Hide this project, so it does not show up in your default project listing. This only impacts you, not your collaborators, and you can easily unhide it.	<a href="#">Hide Project</a>
Delete this project for everyone. You can undo this.	<a href="#">Delete Project...</a>

<sup>4</sup>tato funkce nemá vliv na spolupracovníky

## 4.7.5. Restartování nebo zastavení projektu

V případě, že nástroje v CoCalc nefungují podle očekávání, lze vzdálený server restartovat a obnovit tím činnost projektu. Anebo jej zastavit.

⚙️ Project control

State	 Running Project is running.
Actions	 
Project id	<input type="text" value="e2de7155-2d03-4f14-8702-f8eff1f84f27"/>

If your project is not working, please create a [support ticket](#).

## 5. Požadavky na laboratorní protokol

Následující kapitoly se věnují tématům, které úzce souvisí s estetikou laboratorního protokolu. To znamená, jaké jsou požadavky na protokol, jaká typografická pravidla by v něm měla být dodržována, jak správně vytvářet tabulky a grafy nebo jak správně zapisovat matematické rovnice, fyzikální veličiny a jejich jednotky.

Cílem této a následujících kapitol je zavést pravidla a postupy, jejichž dodržováním bude protokol působit profesionálním dojmem. Následující pravidla a postupy se pochopitelně dají použít také na dokumenty jiného zaměření (chemie, biologie, ekonomie, strojírenství ad.).

### 5.1. Protokol o měření

*Laboratorní protokol* je všeobecně považován za technický dokument (nebo technickou zprávu), který obsahuje postupy z mnoha technických oborů. Na školách technického zaměření se laboratorní protokoly vytvářejí např. v rámci fyzikálního praktika nebo elektrotechnického měření.

*Protokol o měření* se vytváří z důvodu, aby bylo jasné, jak měření probíhalo, jaké důsledky z něho vyplývají, a také proto, aby bylo možné měření zopakovat za stejných podmínek. Jelikož je považován za *oficiální dokument*, měl by se řídit jasnými pravidly, která všeobecně platí pro všechny technické zprávy<sup>1</sup>.

Především by měl mít danou osnovu a měl by být srozumitelný. Čtenáři protokolu by mělo být jasné, co se měřilo, čím a jak. Tabulky, obrázky, rovnice a další

---

<sup>1</sup>pokud není stanoveno jinak



entity by měly být řádně očíslovány a popsány, aby nedocházelo k nejednoznačnosti. Čtenář by neměl dohledávat význam tabulek, smysl grafů apod. Celý protokol by měl být pokud možno stručný a mělo by být jasné, co z měření vyplývá a jaké závěry lze z něho odvodit.

Jak bude zřejmé později, tvorba formálně správného protokolu není jednoduchou záležitostí, a proto doporučujeme této kapitole věnovat obzvlášť velkou pozornost. Uvědomujeme si však, že každý může mít jiné požadavky, a proto studentům doporučujeme, aby se řídili především pokyny svého vyučujícího.

## 5.2. Všeobecné požadavky na protokol

- Má danou osnovu, je srozumitelný, přehledný a jednoznačný
- Zpracovává se tak, aby bylo možné zopakovat měření za stejných podmínek.
- Formát listu a nastavení písma:
  - list formátu A4
  - v celém protokolu<sup>2</sup> by se mělo využívat především patkové písmo, protože se lépe čte, např. Times New Roman
  - velikost písma: 12 pt pro základní text, 9 - 11 pt pro poznámky
  - řádkování 1,5
  - text se zarovnává do bloku
- Číslování
  - Stránky protokolu se číslují uprostřed. Titulní strana se nečísluje (je-li to proveditelné). Přílohy se číslují římskými číslicemi. Stránky protokolu se číslují pro případ, kdyby došlo ke ztrátě listu nebo k jejich přeházení.
  - V protokolu se číslují rovněž obrázky, tabulky, grafy, matematické vzorce (chceme-li se na ně v textu odkazovat), křížové odkazy, po-

---

<sup>2</sup>také v tabulkách a popiskách obrázků

známky pod čarou a odkazy na citace. Mohou se číslovat také kapitoly, sekce a podseky<sup>3</sup>.

- Odkazy
  - Odkazy na citace se uvádí v hranatých závorkách, které jsou odděleny od slova mezerou.
  - Odkazy na vzorce nebo rovnice se uvádí v kulatých závorkách.
  - Vše ostatní je číslováno bez závorek.

Dosazením hodnot do rovnice (4) získáme...

Naměřené hodnoty se shodují s hodnotami v MFCHT [1]

Na obrázku 3.3 je zobrazen graf...

Graf má exponenciální křivku (obr. 7), a proto...

V části 7.1 na straně 12 je pojednáno o...

Tabulka 2.1 obsahuje hodnoty z...

- V protokolu se dodržují základní typografická pravidla (viz kap. 6).
- Popisy obrázků, tabulek a schémat včetně jejich číslování se uvádí pod nimi.
- Vyváření tabulek a grafů se řídí určitými pravidly (viz dále).
- Každý matematický výraz se řídí typografickými pravidly.
- Výpočty se uvádějí celé včetně správného zápisu fyzikálních jednotek.
- Dodatečné komentáře se píše jako poznámky pod čarou (zvyšují přehlednost protokolu).
- V diskusi se komentují poznatky z měření a do závěru se píše pouze výsledek měření.
- Materiály, které snižují přehlednost v protokolu nebo působí rušivě se přesouvají do příloh (dodatečné grafy, obsáhlé tabulky atp.) a uvádí se na ně odkaz.

<sup>3</sup>V takovém případě se čísluje víceúrovňově: 1. Kapitola, 1.1 sekce, 1.1.1 podseky. Kapitoly se začínají psát vždy na nové stránce.

- Výpis programového kódu se píše strojovým písmem.
- Protokol by měl obsahovat seznam použité literatury (byla-li použita) a citace (odkazy na literaturu).

### **5.3. Osnova protokolu**

Každá část protokolu má své nezastupitelné místo. Správně vytvořený protokol se skládá z těchto částí:

1. Titulní strana
2. Pracovní úkol
3. Teoretický úvod
4. Použité přístroje a pomůcky
5. Postup měření
6. Vypracování
7. Diskuse
8. Závěr
9. Použitá literatura
10. Přílohy

Každá část by měla být vypracována jako samostatný oddíl, který má určitou vypovídací schopnost. Protokol by měl být vytvořen tak, aby ho čtenář nemusel číst od začátku do konce. Nyní podrobněji:


#### **5.3.1. Titulní strana**

Každá škola nebo firma může mít jiné požadavky na úvodní stránku. Pokud není stanoveno jinak, formálně správná hlavička by měla obsahovat:

- Název školy, fakulty popř. jejího ústavu
- Název protokolu (např. Fyzikální praktikum)
- Název měřené úlohy
- Jméno a příjmení
- Jména a příjmení spolupracovníků
- Místo měření
- Datum měření

Nezapoménáme na tituly před ani za jménem. Studenti, kteří mají stejná jména, mohou uvést svá studentská čísla, aby nedošlo k záměně.

Některé vysoké školy mohou vyžadovat také položku „Klasifikace“ či „Hodnocení“ nebo místo úvodní stránky úvodní tabulku, která tvoří hlavičku protokolu (viz tabulka 5.4 nebo 5.2). V této práci byla použita úvodní stránka, jejíž příklad je na obrázku 10.2 na straně 83.

	Předmět	
	Jméno a příjmení	
	Osobní číslo	Spolupracovali
	Měřeno dne	Odevzdáno dne
Číslo úlohy	Název měřené úlohy	

Tabulka 5.2.: Tabulka jako hlavička protokolu - vzor 2

Název úlohy	
Jméno a příjmení	
Studijní obor	
Ročník	
Datum měření	

Tabulka 5.4.: Tabulka jako hlavička protokolu - vzor 1

### 5.3.2. Zadání

Zadání se píše heslovitě. Obsahuje-li více pracovních úkolů, musí se protokol rozdělit na části a každý úkol vypracovat zvlášť. Části se poté číslovají římskými číslicemi. Každá část by měla začínat na nové stránce.

<p><b>Část I</b></p> <p><b>Měrný náboj elektronu</b></p> <p>...</p> <p><b>Část II</b></p> <p><b>Difrakce elektronů</b></p> <p>...</p>
---

### 5.3.3. Teoretický úvod

Tato část seznamuje čtenáře s problematikou měření. Měla by obsahovat stručný teoretický rozbor. Vysvětluje základní spojitosti, které jsou nezbytné k pochopení významu vlastního měření. Obsahuje základní pojmy, fyzikální vztahy a veličiny, vysvětluje význam vlastního měření, princip metody měření, obsahuje schémata,

grafy, obrázky apod.

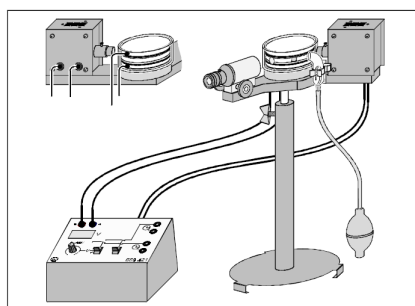
Teoretický rozbor by neměl přesáhnout délku jedné strany a měl by se věnovat jen pasážím, které se bezprostředně týkají vlastního měření <sup>4</sup>.

Teoretický rozbor není nutný, většina škol ho však vyžaduje.

### 5.3.4. Použité přístroje a pomůcky

Uvádí se pouze přístroje a pomůcky, které byly při měření použity, a to včetně jména výrobce, typu a výrobního čísla. Tyto údaje jsou nutné z důvodu, aby se měření dalo zopakovat za stejných podmínek. Rozsáhlejší seznam se uvádí do tabulky.

Používá-li se při měření sestava, která je složena z několika komponentů, doporučuje se vytvořit nákres, aby bylo možné sestavu znovu zapojit (obr. 5.1).



Obrázek 5.1.: Sestava využívaná při Millikanově experimentu

### 5.3.5. Postup měření

Postup měření by měl být stručný, ale také přesný. Je to opět z důvodu, aby se dalo měření zopakovat za stejných podmínek. Je to dobré i v situaci, když je potřeba zpětně dohledat chybu.

Obsahuje-li postup schéma zapojení, kreslí se pokud možno pomocí normalizovaných značek. Jeli to vyžadováno, zaznamenávají se rovněž okolní podmínky

<sup>4</sup>studenti např. uvádějí vzorce, které v protokolu nepoužijí nebo grafy, které nemají s vlastním měřením nic společného

v laboratoři (obvykle teplota, vlhkost a atmosférický tlak vzduchu - viz tabulka 5.5).

Rovněž je nutné poznamenat, že správné měření by mělo obsahovat dostatečný počet naměřených hodnot. Proč je to nutné?

Při měření nějaké fyzikální veličiny vznikají chyby, které mají různé příčiny (lidský faktor, přesnost měřících přístrojů ad.). Abychom chyby co nejvíce eliminovali, musíme fyzikální veličinu měřit vícekrát. Čím více naměřených dat budeme mít, tím přesnější měření bude<sup>5</sup>.

Teplota vzduchu $T_C$	24,40 °C
Teplota vzduchu $T_K$	297,55 K
Tlak vzduchu $p_L$	963,20 hPa
Relativní vlhkost vzduchu $\varphi$	30 %

Tabulka 5.5.: Naměřené hodnoty v laboratoři meteorologickou stanicí

### 5.3.6. Vypracování

Tato část protokolu by měla obsahovat naměřené hodnoty, provedené výpočty, grafy a výpočet celkové chyby měření.

#### Naměřené hodnoty

Naměřené hodnoty se uvádí do tabulky - zvyšují přehlednost. Obsahují-li výsledky výpočtů větší množství hodnot, zpravidla se rovněž uvádějí do tabulky. Příliš obsáhlé tabulky se přesouvají do přílohy.

#### Výpočty

Musí být jasné, jak se k výsledku došlo, proto se vždy uvádí celý výpočet. Respektování této zásady potvrzuje věrohodnost výsledku a lépe se dohledávají chyby.

<sup>5</sup>Minimální počet naměřených hodnot je pět. Při menším počtu se musí hodnoty vynásobit příslušným koeficientem, jinak klesá věrohodnost celkového měření.

Ve výpočtech se uvádí fyzikální jednotky, jejichž zápis se řídí pravidly uvedenými v 6.3.

Matematický vztah, který byl aplikován na více naměřených hodnot (např. aritmetický průměr) stačí napsat jenom jednou, je však nutné konkrétní proměnné přiřadit odpovídající index (zpravidla index s písmenem  $i$ ), který bude odkazovat na hodnoty v nějaké tabulce. Např. vztah 5.1 pro výpočet práce působící síly a vztah 5.2 pro výpočet chyby aritmetického průměru rozdílu napětí:

$$W = \int_{s_1}^{s_2} F_i \, ds + c = \sum_0^7 F_i \cdot 0,1 \, \text{m} + 2 \, \text{J} = 10 \, \text{J} \quad (5.1)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\Delta U_i - \overline{\Delta U})^2}{n(n-1)}} = \sqrt{\frac{\sum_1^6 (\Delta U_i - 4.88 \, \text{V})^2}{6(6-1)}} \doteq 0,03 \, \text{V} \quad (5.2)$$

Z výše uvedených rovnic je zřejmé, že za proměnné  $F_i$  a  $\Delta U_i$  byly dosazeny hodnoty z nějaké tabulky. Tuto informaci je pochopitelně nutné k příslušné rovnici uvést včetně čísla tabulky, ve které jsou hodnoty uvedeny, aby čtenář nemusel tápat a dohledávat hodnoty v protokolu.

## Matematické a fyzikální výrazy

Při zápisu je nutné dodržovat jejich správnou typografii a nezapomínat ani na správný zápis fyzikálních jednotek. Rovněž by se měly dodržovat pravidla při vytváření grafů a tabulek (viz dále).

## Zaokrouhlování

Vypočítané hodnoty se zaokrouhlují pokud možno až na konci celého výpočtu a maximálně na čtyři nenulové číslice za desetinnou čárkou, např. 1,06258 kg. Čísla, která mají delší desetinný rozvoj zapisujeme pomocí mantisy a exponentu (viz dále).



**Chybný zápis:**

0,0054671551684 kg

**Správný zápis:**

0,005467 kg

 $5,467 \times 10^{-3}$  kg (zápis pomocí mantisy a exponentu)**Zápis celkového výsledku včetně chyby měření**

Tabulka 5.7 zobrazuje příklady nesprávných a správných zápisů. Pouze poznamenejme, že obě čísla v kulatých závorkách by měla být celá nebo mít stejný počet desetinných míst.

Chybný zápis	Co je špatně	Správný zápis
$b = (10,14378 \pm 2,1197112) \text{ N.m}^{-1}$	není zaokrouhleno	$b = (10 \pm 2) \text{ N.m}^{-1}$
$l = (0,00653 \pm 0,0000347) \text{ m}$	není zaokrouhleno, mnoho nul	$l = (6,53 \pm 0,03) \text{ mm}$ $l = (6,53 \pm 0,03) \cdot 10^{-3} \text{ m}$
$p = (2,412 \cdot 10^3 \pm 5) \text{ MPa}$	různé řády	$p = (2412 \pm 5) \text{ MPa}$
$\rho = (1,5 \cdot 10^3 \pm 6 \cdot 10^2) \text{ J.K}^{-1}$	různé řády	$\rho = (1,5 \pm 0,6) \cdot 10^3 \text{ kg.m}^{-3}$
$R = 250 \pm 2 \Omega$	chybí závorky	$R = (250 \pm 2) \Omega$
$d = 9,35 \text{ cm} \pm 0,5 \text{ mm}$	různé jednotky	$d = (93,5 \pm 0,5) \text{ mm}$
$f = (16000 \pm 1000) \text{ Hz}$	mnoho nul	$f = (16 \pm 1) \text{ kHz}$ $f = (16 \pm 1) \cdot 10^3 \text{ Hz}$

Tabulka 5.7.: Chybné a správné zápisy celkového výsledku měření

**5.3.7. Diskuse**

Tato část je nejdůležitější z celého protokolu, neboť obsahuje shrnutí celého měření. Popisuje např. průběh měření, rozbor získaných dat, příčiny vzniklých problémů a jejich řešení, co bylo zjištěno, jaký byl výsledek celého snažení atp.

Rovněž můžeme porovnat naměřené výsledky s teoretickými hodnotami např. v MFCHT<sup>6</sup>. Nebo diskutovat, proč není naměřená hodnota přesná dle očekávání,

<sup>6</sup>Matematické, fyzikální a chemické tabulky

identifikovat (případně odhadnout) její příčinu a nabídnout řešení, které povede k nápravě v případě opakovaného měření.

### 5.3.8. Závěr

V této části se stručně uvádí pouze výsledky měření. Např.

Měrný náboj elektronu je  $(1.665 \pm 0.130) \times 10^{11} \text{ C} \cdot \text{kg}^{-1}$ .

Nebo

Energie urychleného elektronu, který uvede atom do excitovaného stavu je stejná, jako energie fotonu, kterou vyzáří atom rtuti po srážce s urychleným elektronem, tedy  $(4.88 \pm 0.09) \text{ eV}$ .

### 5.3.9. Použitá literatura

Laboratorní protokol by měl obsahovat také seznam použité literatury, neboť se předpokládá, že autor protokolu se bez odborné literatury neobešel. Seznam použité literatury se uvádí ve formátu, který je dán normou ČSN ISO 690. Mělo by být dodrženo následující pořadí:

1. autor
2. název knihy nebo článku
3. díl
4. vydání
5. vydavatelství
6. místo
7. rok vydání
8. Celý zápis ukončujeme tečkou.

Následující příklady byly převzaty z [5] a mírně upraveny:

## Publikace

[1] PAVEL, Pavel. *Rapa Nui: jak chodily sochy moai na Velikonočním ostrově*. 1. vyd. Praha: Olympia, 2000. 205 s. ISBN 80-7033-208-5.

## Část publikace – např. kapitola

[2] KOVANICOVÁ, Dana. *Abeceda účetních znalostí pro každého*. Praha: Polygon, 1998. Kapitola 2, Majetek podniku, s. 7–20.

## Časopis

[3] *Forum: časopis Univerzity Karlovy*. Vydává rektorát Univerzity Karlovy. 1995–, roč. 1, č. 1–. Praha: T-Studio, 1995–. 1× za 14 dnů. ISSN 1211-1724.

## Článek v časopise

[4] JAKEŠ, Petr. *Meziplanetární a mezihvězdné částice: prach ze sluneční soustavy a odjinud*. *Vesmír*. 2004, roč. 83, č. 9, s. 521–524.

## WWW stránka

[5] *Český chmel: atlas odrůd* [online]. [cit. 2004-10-15]. Dostupný z WWW: <http://www.beer.cz/humulus/>.

## 5.3.10. Přílohy

Přílohy se číslují velkými písmeny latinské abecedy (A, B, C...) a jejich části arabskými číslicemi. Strany příloh se číslují římskými číslicemi.

### Příloha A

...

### Příloha B

B.1 Název přílohy 1

...

B.2 Název přílohy 2

...

## 5.4. Pravidla pro tabulky

Tabulky slouží k přehlednému publikování naměřených a vypočtených hodnot, k jejich organizaci, a také ke znázornění jejich vzájemných vztahů [5]. Při jejich vytváření by měla být dodržována tato pravidla:

- Tabulka musí mít svoje číslo, název a stručný popis, který se umísťuje pod ní. Popis začíná zkratkou Tab. nebo slovem Tabulka a měl by vystihovat účel a smysl tabulky.
- V tabulce se rovněž dodržují typografická pravidla, zejména při zápisu čísel, fyzikálních veličin a jejich jednotek.
- Sloupce nebo řádky musí obsahovat nadpisy, které začínají velkým písmenem a jsou zarovnány nejlépe na střed buňky (horizontálně i vertikálně). Nepopsaná tabulka má nulovou vypovídající hodnotu. Malým písmenem mohou začínat pouze podnázvy.

Barva	$U_p$ [V]	$f_{\max}$ [THz]	$\sigma$ [THz]	$\lambda_{\max}$ [nm]	$1/\lambda_{\max}$ [ $\times 10^6$ m]
červená	1,73	477,30	8,27	628,10	159,21
zelená	1,92	532,80	7,57	562,67	177,73
modrá	2,63	644,30	13,01	465,30	214,92

Tabulka 5.8.: Naměřené hodnoty LED diod

- Fyzikální veličiny
  - Uvádí se v základních jednotkách případně v řádových jednotkách (mikro-, mili-, kilo-, mega- apod.). Je-li hodnota veličiny příliš velká nebo příliš malá, využívá se zápisu pomocí *mantis*y a *exponentu*, např.  $R_H [\times 10^6 \text{ m}^{-1}]$  (více v 6.1).
  - Veličina se píše kurzívou a jednotka se uvádí do hranatých závorek, např.  $v [\text{m} \cdot \text{s}^{-1}]$  (více v 6.3). Bezrozměrné veličiny mají jednotku:  $[-]$ .

- Veličiny uvedené v tabulce by měly nést stejné označení jako veličiny uvedené v protokolu.
- Celý sloupec (nebo řádek) musí obsahovat jednotky patřící jedné veličině.
- Nejdříve se uvádí sloupce s naměřenými hodnotami, poté sloupce s vypočtenými hodnotami.
- Do prázdných buněk se píše pomlčka.
- Desetinná čísla
  - Naměřené hodnoty se zapisují s takovým počtem desetinných míst s jakým byly změřeny.
  - Vypočtené hodnoty se zaokrouhlují maximálně na čtyři desetinná nenulová čísla.
  - Pokud některé z čísel nemá dostatečný počet desetinných míst, chybějící desetinná místa se doplní nulami.
  - Hodnoty v celém sloupci by měly mít stejný počet desetinných míst. Pokud některé z čísel desetinné není, ale ostatní jsou, udělá se u něj desetinná čárka a za ní se doplní odpovídající počet nul, např. 2,00.
  - Čísla se zarovnávají podle desetinné čárky.
- Odchytky od aritmetického průměru se uvádí včetně znamének + a – (viz tab. 5.9).
- Tabulka, která přesahuje okraje stránky se dává do přílohy na šířku. Je-li tabulka rozdělena na více stránek, musí na každé stránce začínat opět hlavičkou.
- Do tabulky se mohou uvádět aritmetické průměry. Ty se obvykle nachází na posledním řádku, který se odděluje linkou. Jsou-li již řádky odděleny linkou, oddělí se poslední řádek dvojitou linkou.
- Šířka sloupců se obvykle volí podle nejširšího údaje v tabulce, ale není to nutné.

Příklady některých tabulek viz 5.9, 5.8 a 5.11.

$U_A$ kV	$e$ [m]	$r$ [m]	$B$ [ $\times 10^{-3}$ T]	$\frac{q_e}{m_e}$ [ $\times 10^{11}$ C $\cdot$ kg $^{-1}$ ]	$\Delta\left(\frac{q_e}{m_e}\right)$ [ $\times 10^{11}$ C $\cdot$ kg $^{-1}$ ]	$\left[\Delta\left(\frac{q_e}{m_e}\right)\right]^2$ [ $\times 10^{21}$ C $\cdot$ kg $^{-1}$ ]
1	0,050	0,094	1,263	2,819	+1,154	13,313
	0,045	0,092	1,643	1,754	+0,089	0,004
	0,040	0,089	2,066	1,171	-0,494	2,440
	0,035	0,087	2,449	0,875	-0,790	6,247
2	0,050	0,094	1,416	3,364	+1,699	28,849
	0,045	0,092	1,951	1,870	+0,205	0,420
	0,040	0,089	2,487	1,213	-0,453	2,049
	0,035	0,087	2,946	0,907	-0,759	5,756
3	0,050	0,094	1,913	2,456	+0,791	6,253
	0,045	0,092	2,334	1,743	+0,078	0,006
	0,040	0,089	2,908	1,183	-0,483	2,329
	0,035	0,087	3,367	0,925	-0,740	5,471
4	0,050	0,094	2,066	2,632	+0,967	9,346
	0,045	0,092	2,678	1,655	-0,011	3,311
	0,040	0,089	3,252	1,182	-0,483	2,337
	0,035	0,087	3,826	0,896	-0,769	5,920

Tabulka 5.9.: Vypočtené hodnoty pro stanovení měrného náboje elektronu

## 5.5. Pravidla pro grafy

Graf vyjadřuje závislost mezi dvěma veličinami. Představuje jednu z nedůležitějších forem výsledků měření, proto se objevuje téměř v každém protokolu. Při jeho vytváření by měla být dodržována následující pravidla:

- Graf musí mít svoje číslo, název a stručný popis, který umísťujeme pod něj. Popis začíná zkratkou Obr. nebo slovem Obrázek a měl by vystihovat účel a smysl grafu.
- V grafu se rovněž dodržují typografická pravidla, zejména při popisu os a zobrazování legendy.

Č. měření	$U$ [V]	$I$ [mA]
1	2,31	69,40
2	2,21	58,80
3	2,13	48,80
4	2,03	35,30
5	1,91	21,20
6	1,80	9,02
7	1,70	2,46
8	1,65	1,02
9	1,61	0,38
10	1,55	0,10

Tabulka 5.11.: Naměřené hodnoty napětí a proudu

- Osy grafu
  - Všechny osy musí mít svůj popisek, který je umístěn pokud možno na jejich konci<sup>7</sup>.
  - Rozsah os se volí tak, aby byla využita co největší plocha grafu. Osy nemusí začínat v počátku souřadnicové soustavy.
  - Stupnice na osách grafu mohou být lineární nebo nelineární<sup>8</sup> - volí se s ohledem na přehlednost grafu.
- Popisy os
  - Fyzikální veličiny se uvádí v základních jednotkách případně v řádových jednotkách (mikro-, mili-, kilo-, mega- apod.). Je-li hodnota veličiny příliš velká nebo příliš malá, využívá se zápisu pomocí *mantisy* a *exponentu*, např.  $R_H [\times 10^6 \text{ m}^{-1}]$  (více v 6.1).
  - Veličina se píše kurzívou a jednotka se uvádí do hranatých závorek, např.  $v [\text{m} \cdot \text{s}^{-1}]$  (více v 6.3). Bezrozměrné veličiny mají jednotku:  $[-]$ .

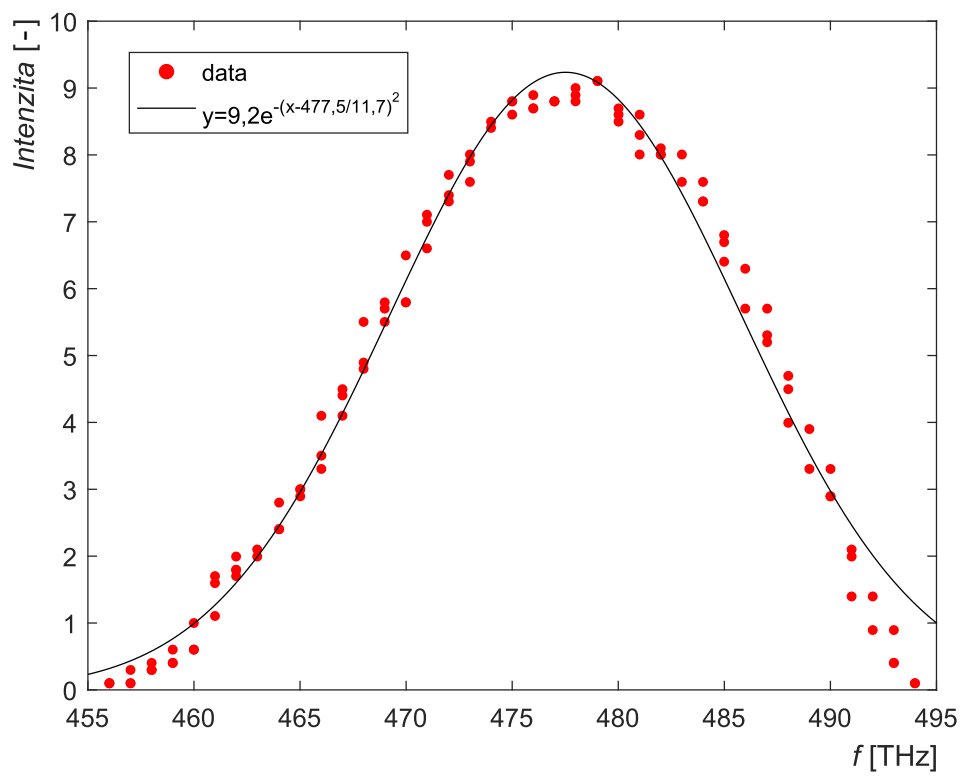
<sup>7</sup>Dodnes není stanoveno, zda má být popis uprostřed osy nebo na jejím konci. Ze zkušenosti víme, že popisy na konci os se vyskytují ve většině odborných pracích a v mnoha vysokškolských publikacích, proto doporučujeme držet se těchto zvyklostí.

<sup>8</sup>logaritmická, kvadratická atd.

- Veličiny uvedené v grafu by měly nést stejné označení jako veličiny uvedené v protokolu.
- Mřížky slouží k lepšímu odečítání hodnot z grafu. Je nutné zvážít, zda mřížku dávat nebo ne. Např. pokud je cílem zobrazít jenom průběh fyzikální veličiny, tak se mřížka nedává, abych byla zachována přehlednost grafu. Naopak v případě, kdy je nutné odečíst hodnoty z grafu, dáva se taková hustota mřížky, která odpovídá nejmenšímu dílku na ose (viz graf v příloze C).
- Body v grafu
  - Je-li v grafu zobrazena závislost tvořená body, tyto body se nespojují, ale aproximují se příslušnou přímkou nebo křivkou a uvádí se její rovnice (více v 9.6.3).
  - Jsou-li v grafu zobrazeny pouze body, používá se k jejich zobrazení vhodných grafických symbolů (tečka, křížek, kolečko, čtvereček, ...).
- Barevná výplň grafu se nepoužívá, protože zvyšuje nepřehlednost.
- Legenda grafu
  - Zobrazuje se v případě, když graf obsahuje více jak jednu křivku, aby bylo jasné, co která křivka zobrazuje.
  - Umísťuje do volného rohu uvnitř grafu nebo vně grafu nejčastěji pod ním nebo napravo.
  - Do legendy se umísťuje taktéž rovnice proložené přímkou (nebo křivky). Veličiny uvedené v rovnici by měly souhlasit s veličinami na osách.
- Graf se vytváří s ohledem na černobílý tisk. Např. u většího počtu křivek se volí odlišné typy čar.
- Každý graf by měl být pokud možno uzavřený (ohraňčený ze všech stran).
- Graf by měl být dostatečně velký - alespoň formát A5.

Na obrázku 5.2 je graf, který respektuje výše uvedená pravidla. Další grafy viz příloha B.





Obrázek 5.2.: Hledání střední frekvence  $f_{max}$  červené LED diody na základě naměřených hodnot spektrometrem

## 6. Typografická pravidla

Typografie je obor, který se zabývá úpravou tiskovin a současně je to druh grafického umění. Z hlediska grafického se zabývá tvarem písma a jeho dalšími vlastnostmi. Snaží se o co nejlepší čitelnost textu, aby se v něm čtenář dobře orientoval, a také aby působil esteticky. Z hlediska úpravy textu zavádí různá pravidla, mezi která patří např. správné psaní uvozovek, dodržování mezer, pravidla pro formátování odstavců atd. Do dalšího odvětví typografie by se dala zařadit pravidla pro zápis matematických rovnic, fyzikálních veličin a jejich jednotek.

V našem případě se *typografická kvalita písma* projeví pouze při sázení matematických rovnic, protože tuto výhodu poskytuje pouze jazyk L<sup>A</sup>T<sub>E</sub>X. Zbytek dokumentu bude vytvářen v jazyku Markdown. Nicméně *typografická pravidla* musíme dodržovat sami, proto jim byla věnována tato kapitola. Je nutné upozornit, že pravidla z této kapitoly by se měla dodržovat nejen v samotném textu, ale také v tabulkách a grafech.

### 6.1. Všeobecná pravidla

Pravidla, která jsou notoricky běžná, byla vynechána. Na internetu je k dispozici dostatek literatury, který se typografických pravidlům věnuje podrobněji. Mezi základní typografická pravidla při tvorbě dokumentů patří:

- **Vhodné písmo** - nejčastěji se doporučuje volit patkové písmo, protože se dobře čte, nejpoužívanější je Times New Roman.
- **Jednoznakové předložky** v, s, z, k, o, u, i by neměly zůstat na konci řádku. V L<sup>A</sup>T<sub>E</sub>Xu se to řeší vložením pevné nezlomitelné mezery mezi písmeno a slovo. Díky tomu je předložka na stejném řádku jako následující slovo.

- Za „**viz**“ se nedělá tečka, protože to není zkratka, ale slovo.

- **Spojovník**, tj. krátká vodorovná čárka „-“ se používá bez mezer, např.

matematicko-fyzikální tabulky  
anglicko-český slovník  
n-tice  
Praha-západ  
bude-li, není-li

- **Pomlčka** „ - “ má mezeru z obou stran a používá se pro nahrazení závorky, může nahrazovat čárku v souvětí, anebo se používá ve formě rozsahu „až-do“ (bez mezer), např.

Do mlžné komory byl umístěn suchý led - asi 5 kg.  
Částice - elektrony a protony - byly detekovány.  
váha 10-20 kg  
8-16 hodin  
10.–20. 3. 2017  
směr Praha-Brno

- Pokud věta končí **zkratkou** (která má tečku), další tečka se již nepíše, např.

Firma Technika s.r.o.  
Barva grafu, průhlednost, rozsah grafu atd.

- **Dvojtečka** použitá ve významu dělení a poměru se píše s mezerami.

Dělení 10 : 5 = 2  
Poměr 3 : 1

- **Uvozovky** - existuje několik typů uvozovek. V českém jazyce se používají výhradně uvozovky 9966 („“).
- **Závorky** - při psaní textu se používají výhradně kulaté závorky (). Hra-  
naté [ ] a složené {} závorky se používají především v matematice a pro-  
gramování.
- **Procento, stupeň, fyzikální jednotka** - rozlišujeme zápis bez mezery  
a s mezerou. Zápis s bez mezery znamená jedno slovo, zatímco zápis s me-

zerou (nezlomitelnou) znamená dvě slova. Zápis fyzikálních jednotek je lépe popsán v 6.3.

5%	čteme jako „pětiprocentní“
5°	čteme jako „pětistupňový“
5m	čteme jako „pětimetrový“

5 %	čteme jako „pět procent“
5 °	čteme jako „pět stupňů“
5 m	čteme jako „pět metrů“

- **Datum a čas** - datum musí zůstat v celku, proto by měl obsahovat nezlomitelné mezery. Ve formě rozsahu „až-do“, resp. „od-do“ se používá pomlčka bez mezer. Pokud je však datum zapsán pomocí více slov, doporučuje se u pomlčky udělat nezlomitelnou mezeru.

Mezi hodiny a minuty se podle normy ČSN 01 6910 píše dvojtečka, zatímco podle pravidel českého pravopisu tečka. Dvojtečka bez mezer se dělá také mezi minutami a sekundami, ale desetiny a setiny sekundy se již oddělují čárkou.

3. 6. 2017
3. července 2017
10.-17. 8. 2017
9. ledna - 7. září
7:00 hodin
15.20 hodin
32:10,46 (32 minut, 10 sekund, 46 setin sekundy)

- **Řádkový zlom** - u některých názvů, jmen, čísel nebo zkratk by neměl nastat řádkový zlom<sup>1</sup>, např.

Ing. Jan Novák, PhD.
10. listopadu 2006

<sup>1</sup>Tj. přechod na nový řádek. Kdyby se tak stalo, na prvním řádku by se nacházel titul Ing. a na druhém jméno Jan Novák, což je z typografického hlediska nepřipustné.

## Čísła

Věta by neměla začínat číslem. Číslo a slovo tvořící jednoslovný výraz se píše vždy dohromady. Číslo v souvislém textu nahrazujeme slovem (netýká se technických textů).

### **Chybný zápis:**

7 kusů bylo vyexportováno do zahraničí.

Ve 2 hodiny bylo vyrobeno 1000 kusů.

3x, 8-bitový, 2-denní, xtý

Váha je pět tisíc čtyřista šedesát dva kilogramů.

### **Správný zápis:**

Sedm kusů bylo vyexportováno do zahraničí.

Ve dvě hodiny bylo vyrobeno tisíc kusů.

3krát, 8bitový, 2denní (lépe dvoudenní), ale v matematice se píše x-tý, n-násobek atp.

Ve skladu je 236 náhradních dílů. Kotoučová pila má 125 zubů. Váha je 5 462 kg. (platí pro technické texty)

Při zápisu čísel se vždy používá desetinná čárka<sup>2</sup>. Čísła s více než třemi číslicemi dělíme po trojicích pomocí nezlomitelné mezery<sup>3</sup>. V  $\text{\LaTeX}$ u se čísla oddělují pomocí nezlomitelné mezery o velikosti 1/4 čtverčíku příkazem `\;` ; a desetinná čárka se zapisuje příkazem `{,}` (viz dále).

Velmi dlouhá čísla se zapisují ve zkrácené podobě pomocí *mantis*y a *exponentu* [8], např.

$$-1,602 \times 10^{-19},$$

$$(3,5 \pm +0,1) \times 10^{-19},$$

případně pomocí řádových jednotek (mikro-, mili-, kilo-, mega- apod.) - viz tabulky 5.9, 5.8 a 5.11 v 5.4. Někdy bývá symbol ( $\times$ ) nahrazen středovou tečkou

<sup>2</sup>desetinná tečka se používá v jiných zemích

<sup>3</sup>neplatí pro sériová a inventární čísla

(·), nicméně symbol ( $\times$ ) byl určen Mezinárodním úřadem pro míry a váhy.

V některých případech se můžeme setkat se zápisem čísel v takzvané „*E formě*“, např.  $6,10e - 8$  nebo  $-3.81E - 6$ . Zápis se používá zejména tam, kde je problematické vysázet exponent (např. kalkulačka).

Tyto zápisy čísel by měly být dodržovány nejen při výpočtech, ale také v tabulkách a grafech. Viz tabulky v 5.4.

0,2
8 848
0,141 592 6
16 246,4
732 111 856 (telefonní číslo)
Zápis v $\LaTeX$ u:
$0\{, \}2$
$8\backslash;848$
$0\{, \}141\backslash;592\backslash;6$
$16\backslash;246\{, \}4$
$732\backslash;111\backslash;856$

## 6.2. Pravidla pro zápis matematických výrazů

Aby výrazy byly čitelné a jednoznačné, je nutné u nich oddělovat řezy písma. Zjednodušeně řečeno: *kurzívou*<sup>4</sup> se píše jenom proměnné, vektory se píše **tučným** písmem (výjimečně se šipkou) a vše ostatní antikvou<sup>5</sup>.

### Výrazy, které se píše antikvou

- **Konstanty a matematické symboly**

Eulerovo číslo $e$ , rychlost světla $c$ , suma $\sum$ , integrál $\int$ , diferenciál $d$
--

- **Funkce**

$\sin$ , $\arccos$ , $\exp$ , $\log$ , $\min$ , $\max$
--

<sup>4</sup>šikmým řezem písma neboli italkou

<sup>5</sup>základním (rovným) řezem písma

- **Prvky a částice**

He, e<sup>-</sup>, p, n

- **Rovnice** - neočíslované rovnice se mohou psát na řádek. Očíslované vzorce se píší na nový řádek a zarovnávají se na střed. Pořadové číslo se uvádí v kulatých závorkách.

$$V = \frac{4}{3}\pi r^3 \quad (6.1)$$

Rozdíl napětí  $\Delta U = U_2 - U_1$  je dán...

## Ostatní výrazy

- **Proměnné** se píší kurzívou, a to i proměnné označené řeckými písmeny nebo proměnné umístěné v horním či dolním indexu.

proměnná  $x$ , poloměr  $r$ , úhlová rychlost  $\omega$ , vlnová délka  $\lambda$ , odpor  $R_i$

- **Vektory** se píší antikvou a tučným řezem písma, výjimečně se šipkou nad symbolem.

$$\mathbf{c} = \mathbf{a} \times \mathbf{b}, W = \int \mathbf{F} d\mathbf{r}$$

$$\vec{c} = \vec{a} \times \vec{b}, W = \int \vec{F} d\vec{r}$$

- **Exponenty a dolní indexy**

Platí stejná pravidla jako v ostatní případech. Tj. pokud je v dolním indexu číslo, písmeno nebo konstanta, píší se antikvou. Pokud dolní index obsahuje proměnnou, píše se kurzívou.

$$\mathbf{F}_3, x_1, U_{\text{out}}, y_k, u_2(t), l_{n-1}$$

Exponenty se zapisují obdobným způsobem:

$$m^2, v^{e+x}, x^n, 10^{-3}$$

- **Řecká písmena** - jako proměnné kurzívou, jako fyzikální jednotky antikvou, jako přírůstek delta  $\Delta$  antikvou.

úhlová rychlost  $\omega$ , vlnová délka  $\lambda$ ,  $10\ \Omega$ ,  $5\ \mu\text{F}$ ,  $\Delta U$ ,  $\Delta x$

## Pozor na záměny

- Součin nebo skalární součin by se neměl zapisovat běžnou tečkou na konci věty (.) nebo písmenem (x). Správně se má zapisovat středovou tečkou (·).

**Chybný zápis:**  $c = a.b$  nebo  $c = a x b$

**Správný zápis:**  $c = a \cdot b$

- Rovněž vektorový součin by se neměl nahrazovat písmenem (x). Používá se znak vyhrazený pro vektorový součin ( $\times$ ).

**Správný zápis:**  $c = a \times b$

- Úhlové minuty by se neměly zaměňovat s apostrofem a úhlové vteřiny s uvozovkami.

**Chybný zápis:**  $7'10''$  nebo  $7'10''$

**Správný zápis:**  $7'10''$

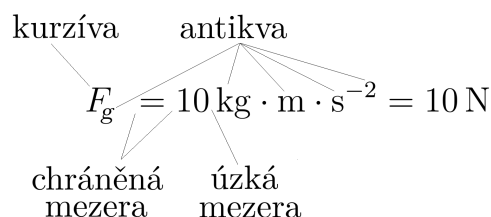
- Znak mínus by měl být stejně dlouhý a stejně vysoko jako znak plus (tj.  $+ -$ ).

## 6.3. Pravidla pro zápis fyzikálních veličin a jejich jednotek

Fyzikální veličiny se píše kurzívou, vše ostatní antikvou. U rovnítka (=) se píše nezlomitelná *chráněná mezera* a mezi hodnotou a jednotkou fyzikální veličiny se píše nezlomitelná *úzká mezera*.

Má-li fyzikální veličina více jednotek, dělá se mezi nimi *středová tečka* (·) - viz obrázek 6.1.





Obrázek 6.1.: Zápis fyzikálních veličin a jejich jednotek

## Zápis v tabulkách a grafech

Dodnes nebylo jasně stanoveno, zda se mají fyzikální jednotky psát do hranatých nebo do kulatých závorek. Některé odborné práce dokonce závorky neuvádějí vůbec a oddělují fyzikální veličinu od její jednotky čárkou, např.  $I$ ,  $A$  nebo  $v$ ,  $\text{m} \cdot \text{s}^{-1}$ .

Ze zkušenosti víme, že většina odborných prací a mnoho vysokoškolských publikací fyzikální jednotky uvádí v hranatých závorkách, proto doporučujeme držet se těchto zvyklostí. Rozměry<sup>6</sup> fyzikálních jednotek se rovněž píšou do hranatých závorek.

Zápis fyzikální veličiny a její jednotky:	$I [A]$ , $U [\text{kV}]$ , $v [\text{m} \cdot \text{s}^{-1}]$ , $R_H [\times 10^6 \text{ m}^{-1}]$
Rozměr fyzikální veličiny:	$[N] = [\text{kg} \cdot \text{m} \cdot \text{s}^{-2}]$

## Zápis pomocí mantisy a exponentu

Jak bylo uvedeno v 6.1, je-li hodnota veličiny příliš velká nebo příliš malá, využívá se zápisu pomocí *mantisy* a *exponentu*, např.

$$m_e = 9.622 \times 10^{-31} \text{ kg}, \quad (6.2)$$

$$R_H = (1.109 \pm 0.013) \times 10^7 \text{ m}^{-1}, \quad (6.3)$$

$$\frac{0,000\ 015 \times 10^{-15}}{8,617\ 343 \times 10^{-15}} \doteq 1.74 \times 10^{-6}. \quad (6.4)$$

<sup>6</sup>může se někdy objevit při výpočtu v protokolu

Správného zápisu lze dosáhnout využitím jazyka  $\text{\LaTeX}$ , kterému bude věnována pozornost v následující kapitole.

Příklad viz tabulka 5.9, 5.8 a 5.11 na straně 39. Příklady grafů jsou v přílohách B, C a D.

## 7. Sázení matematických rovnic v jazyku Latex

L<sup>A</sup>T<sub>E</sub>X je považován za nejlepší nástroj pro sazbu matematických rovnic. Především pro svou vysokou typografickou kvalitu a způsob, kterým jsou matematické symboly vysázeny. Stačí zadat sadu příkazů a o vše ostatní se postará sám L<sup>A</sup>T<sub>E</sub>X.

Níže uvedené příkazy lze využít také pro sazbu jiných výrazů než jsou matematické rovnice. Např. je možné uprostřed textu napsat <sup>137</sup>Ba nebo 10 kg (antikvou a s úzkou mezerou).

V předchozí kapitole bylo pojednáno o chráněné mezeře. Tato mezeře se používá v případě, když se rovnice (slovo či slovní spojení) nevhodně rozdělí na dva řádky, jinak nutná není.

### Při sázení je nutné dodržovat tato pravidla:

- Některé příkazy začínají zpětným lomítkem `\` a některé jsou se složenými závorkami `{}`.
- Za každým příkazem se dělá mezeře.
- Pokud za příkazem následuje delší matematická formule, uzavírá se do složených závorek, např. `\sqrt{a+b}`.
- Příkazy pro jednu rovnici se vždy uzavírají mezi dva symboly dolaru `$` (tzv. *matematický zápis*), např. `$c=a+b$`.
- Rovnice uzavřená mezi čtyři symboly dolaru se umístí na střed nového řádku, např. `$$c=a+b$$`.

- Fyzikální jednotky se od čísla oddělují úzkou mezerou a čísla se dělí na trojice chráněnou mezerou.
- Chráněná mezera se používá pouze v případě, je-li matematický zápis nevhodně zalomen.
- Dáváme pozor, abychom nenapsali o závorku nebo o znak dolaru navíc.
- V Markdown editoru se matematické rovnice číslují pomocí mezery `qqad`, např. `$y=2x\qqad(1)$`.

## 7.1. Matematické operace

Operace	Příkaz	Příklad zápisu	Grafický výstup
součet	<code>+</code>	<code>\$a+b\$</code>	$a + b$
rozdíl	<code>-</code>	<code>\$a-b\$</code>	$a - b$
součin	<code>\cdot</code>	<code>\$a\cdot b\$</code>	$a \cdot b$
podíl	<code>/</code>	<code>\$a/b\$</code>	$a/b$
rovnost	<code>=</code>	<code>\$c=a+b\$</code>	$c = a + b$
zápis zlomku	<code>\frac{}{}</code>	<code>\$\frac{a}{b}\$</code>	$\frac{a}{b}$
mocnina	<code>^{} </code>	<code>\$x^{-19}\$</code>	$x^{-19}$
druhá odmocnina	<code>\sqrt</code>	<code>\$\sqrt{a+b}\$</code>	$\sqrt{a + b}$
$n$ -tá odmocnina	<code>\sqrt[n]{}</code>	<code>\$\sqrt[3]{a+b}\$</code>	$\sqrt[3]{a + b}$
logaritmus	<code>\log</code>	<code>\$\log 5\$</code>	$\log 5$
přirozený logaritmus	<code>\ln</code>	<code>\$\ln 5\$</code>	$\ln 5$
funkce sinus <sup>1</sup>	<code>\sin</code>	<code>\$\sin x\$</code>	$\sin x$
suma	<code>\sum</code>	<code>\$\sum i\$</code>	$\sum i$

<sup>1</sup>obdobně `\cos`, `\tan`, `\cot`, `\arcsin`, `\arccos` atp.

## 7.2. Matematické symboly

Symbol	Příkaz	Příklad zápisu	Grafický výstup
znaménko $\pm$	<code>\pm</code>	<code>\$30\pm 2\$</code>	$30 \pm 2$
vektorový součin $\times$	<code>\times</code>	<code>\$V1 \times V2 \$</code>	$V1 \times V2$
zápis čísla pomocí mantisy a exp.	<code>\times</code>	<code>\$5\times 10^{-9}\$</code>	$5 \times 10^{-9}$
symbol nekonečno $\infty$	<code>\infty</code>	<code>\$+\infty\$</code>	$+\infty$
symbol zaokrouhlení $\doteq$	<code>\doteq</code>	<code>\$1{,}75\doteq 2\$</code>	$1,75 \doteq 2$

Symbol	Příkaz	Příklad zápisu	Grafický výstup
absolutní hodnota $\bar{a}$	<code>\overline</code>	<code>\overline{x+y}</code>	$\overline{x + y}$
vektor $\vec{a}$	<code>\vec</code>	<code>\$\vec{a}\$</code>	$\vec{a}$
první derivace podle času $\dot{a}$	<code>\dot</code>	<code>\$\dot a\$</code>	$\dot{a}$
druhá derivace podle času $\ddot{a}$	<code>\ddot</code>	<code>\$\ddot a\$</code>	$\ddot{a}$
integrál $\int$	<code>\int</code>	<code>\$\int\{f(x)dx\}\$</code>	$\int f(x) dx$
procento %	<code>\%</code>	<code>55\,\%</code>	55 %
kulaté závorky $()$	<code>()</code>	<code>\$6c(2a+b)\$</code>	$6c(2a + b)$
hranaté závorky $[\ ]$	<code>[ ]</code>	<code>\$x[6c(2a+b)]\$</code>	$x [6c(2a + b)]$
složené závorky	<code>\{ \}</code>	<code>\$2\{x[6c(2a+b)]\}\$</code>	$2 \{x [6c(2a + b)]\}$

## 7.3. Symboly řecké abecedy

Pozn. Tabulka obsahuje jen nejpoužívanější symboly.

Sym.	Příkaz	Sym.	Příkaz	Sym.	Příkaz	Sym.	Příkaz
$\alpha$	<code>\alpha</code>	$\vartheta$	<code>\vartheta</code>	$\varrho$	<code>\varrho</code>	$\omega$	<code>\omega</code>
$\beta$	<code>\beta</code>	$\lambda$	<code>\lambda</code>	$\sigma$	<code>\sigma</code>	$\Delta$	<code>\Delta</code>
$\gamma$	<code>\gamma</code>	$\mu$	<code>\mu</code>	$\tau$	<code>\tau</code>	$\Sigma$	<code>\Sigma</code>
$\delta$	<code>\delta</code>	$\nu$	<code>\nu</code>	$\upsilon$	<code>\upsilon</code>	$\Phi$	<code>\Phi</code>
$\varepsilon$	<code>\varepsilon</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>	$\Omega$	<code>\Omega</code>
$\theta$	<code>\theta</code>	$\rho$	<code>\rho</code>	$\varphi$	<code>\varphi</code>	-	-

## 7.4. Ostatní příkazy

Popis	Příkaz	Příklad zápisu	Grafický výstup
rovný řez písma (antikva)	<code>\mathrm</code>	<code>\mathrm{písmo}</code>	písmo
tučný řez písma	<code>\mathbf</code>	<code>\mathbf{písmo}</code>	<b>písmo</b>
úzká mezera (fyzikální jednotky)	<code>\,</code>	<code>\,cm</code>	1 cm
dlouhá mezera (pro číslování rovnic)	<code>\quad</code>	<code>y=2x\quad(1)</code>	$y = 2x \quad (1)$
chráněná mezera <sup>2</sup>	<code>\;</code>	<code>2+4\;=\;6</code>	$2 + 4 = 6$
mezislovní nezalomitelná mezera	<code>\sim</code>	<code>t\sim[\mathrm{min}]</code>	$t \text{ [min]}$
desetinná čárka	<code>{,}</code>	<code>1{,}325</code>	1,325
dolní index	<code>_</code>	<code>R_2</code>	$R_2$

## 7.5. Příklad

Níže jsou vzorce vysázené v jazyku  $\text{\LaTeX}$  s ohledem na správnou typografii. Vedle každého vzorce je uveden příslušný blok kódu, který lze skopírovat do Markdown editoru.

---

<sup>2</sup>neboli široká mezera

$$d = \frac{m\lambda}{\sin \overline{\alpha}_i}$$

$$d = \frac{m\lambda}{\sin \overline{\alpha}_i}$$

$$N(t) = N_0 e^{-\lambda t}$$

$$N(t) = N_0 e^{-\lambda t}$$

$$E = m_s g_e \mu_B B_0$$

$$E = m_s g_e \mu_B B_0$$

$$B = \mu_0 \frac{NR^2}{\left[R^2 + \left(\frac{1}{2}\right)^2\right]^{\frac{3}{2}}} I$$

$$B = \mu_0 \frac{NR^2}{\left[R^2 + \left(\frac{1}{2}\right)^2\right]^{\frac{3}{2}}} I$$

$$\overline{R_\infty} = \frac{1}{4} \sum_{i=1}^4 R_{\infty i}$$

$$\overline{R_\infty} = \frac{1}{4} \sum_{i=1}^4 R_{\infty i}$$

$$\overline{\delta h} = \sqrt{\frac{\sum_{i=1}^4 (\Delta h_i)^2}{n(n-1)}}$$

$$\overline{\delta h} = \sqrt{\frac{\sum_{i=1}^4 (\Delta h_i)^2}{n(n-1)}}$$

$$\frac{1}{\lambda} = \frac{\mu e^4}{8\epsilon_0^2 h^3 c} \left( \frac{1}{2^2} - \frac{1}{n^2} \right)$$

$$\frac{1}{\lambda} = \frac{\mu e^4}{8\epsilon_0^2 h^3 c} \left( \frac{1}{2^2} - \frac{1}{n^2} \right)$$

## 8. Zpracování naměřených hodnot v Jupyter Notebooku

Z 2. kapitoly víme, že Jupyter Notebook je výpočetní prostředí, ve kterém je možné zpracovávat naměřená data. Jedná se o velmi silný nástroj, který využívá řada významných firem (obr. 8.1).

Data v Jupyteru lze zpracovávat pomocí mnoha programovacích jazyků. Jupyter nainstalovaný na počítači<sup>1</sup> jich podporuje až čtyřicet. Pro naše potřeby byl zvolen jazyk Octave, neboť je snadný a do velké míry kompatibilní s jazykem MATLAB, který je populární na většině vysokých škol.

Tato kapitola nepopisuje všechny funkce Jupyteru, ale pouze ty důležité pro zpracování naměřených dat.



---

<sup>1</sup><https://jupyter.org/>












## Práce v Jupiter Notebook

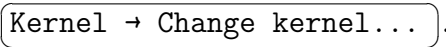


Po vytvoření Jupyter zápisníku s příponou \*.ipynb (vysvětleno v 4.2) se otevře prostředí jako na obrázku 8.2. Základem Jupyteru je buňka, do které lze:


- Napsat nějaký blok kódu a spustit ho (kód je zkompileován patřičným kompilátorem a výstup je zobrazen pod buňkou). Díky tomu lze provádět různé výpočty, zobrazovat grafy a další.
- Psát prostý text a formátovat jej pomocí jazyka Markdown a  $\text{\LaTeX}$ . Díky tomu lze napsat nějaký úvod nebo jiné poznámky týkající se výpočtu.


Tak např. v první buňce je možné napsat stručné poznámky v Markdownu, ve druhé vypočítat aritmetický průměr z naměřených hodnot, ve třetí nějakou nepřímou fyzikální veličinu a ve čtvrté zobrazit graf s nějakou závislostí. Implementace záleží pouze na našich požadavcích a schopnostech.

Každá buňka může vykonávat něco jiného. Buňky se tváří se jako samostatné objekty. Nejsou však od sebe izolovány. Jinými slovy, je-li v první buňce deklarována nebo definována proměnná, můžeme ji zavolat v jiné buňce.

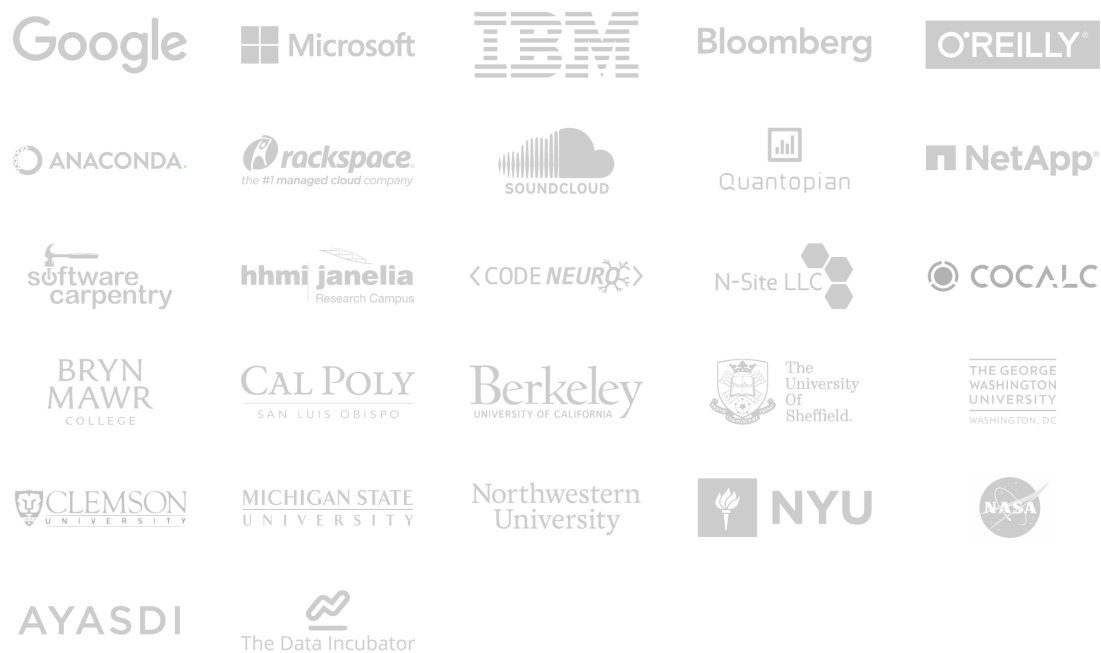
Buňky lze přidávat tlačítkem puls  nebo mazat nůžkami . Lze je také přesouvat směrem nahoru nebo dolů šipkami  . Historii úprav je možné procházet pomocí šipek   nebo lze zobrazit celou její historii tlačítkem . Je-li v buňce spustitelný kód, lze jej přeložit a spustit tlačítkem . Tlačítko  pochopitelně slouží k uložení sešitu.

V každé buňce lze používat jenom pouze jeden typ programovacího jazyka, který lze zvolit z nabídky . V pravém horním rohu se nachází indikátor kernelu  , který je zrovna využíván. Občas se stává, že kernel nefunguje jak má (indikátor se změní) a musí se restartovat v nabídce

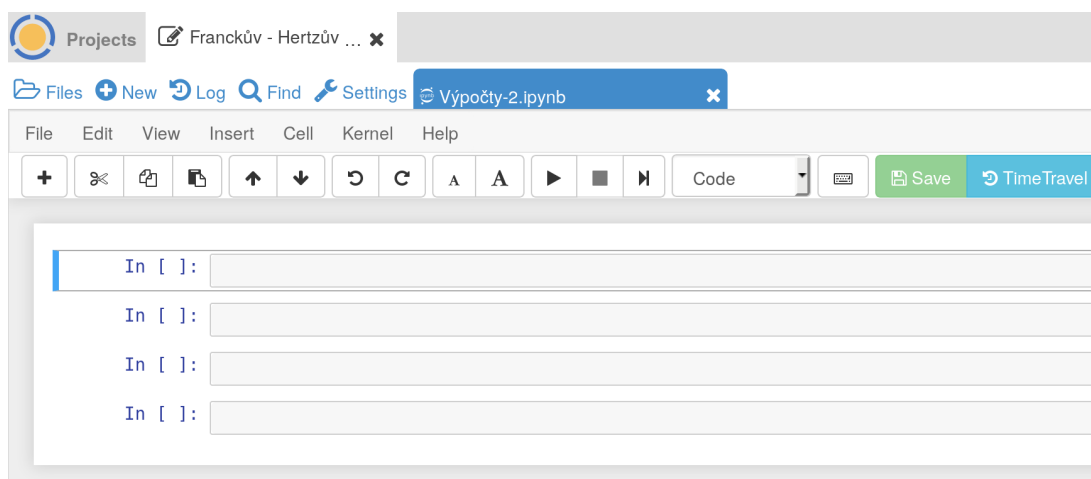
. Nejedná se však o nic mimořádného a nedochází při tom ke ztrátě dat. Tato situace může být způsobena např. aktualizací systému nebo výpadkem signálu.

Pomocí rozevírací nabídky  je možné buňku nastavit tak, aby šlo do ní místo kódu psát prostý text, který lze následně formátovat pomocí jazyka Markdown a  $\text{\LaTeX}$ , jak bylo uvedeno výše.

Za zmínku také stojí, že Jupyter Notebook je samostatný open-source projekt, který lze s výhodou nainstalovat na počítač. Nativní verze totiž nabízí mnohem více možností.



Obrázek 8.1.: Společnosti, které využívají Jupyter Notebook [1]




Obrázek 8.2.: Notebook Jupyter - prostředí

## Příklad

Na obr. 8.3 je uveden příklad pro výpočet energie vyzářeného fotonu. Buňky mají následující obsah:

- První buňka obsahuje informace týkající se výpočtu. Pro formátování textu byl využit jazyk Markdown a  $\LaTeX$ .
- Druhá buňka obsahuje definice konstant, které jsou nutné pro výpočet.
- Ve třetí buňce byl proveden výpočet energie fotonu.

Obsah každé buňky se spouští zvlášť tlačítkem . Buňky se po tomto kroku změny jako na obrázku 8.4. Výsledek výpočtu se zobrazí pod třetí buňkou.

Jak sami vidíte, nejedná se o složitý příklad (zápis výpočtu je podobný zápisu na studentské kalkulačce) a získali jsme elegantní řešení, ke kterému se lze kdykoli vrátet a libovolně jej modifikovat. Nicméně kompletní zpracování naměřených hodnot a vytvoření grafu vyžaduje minimálně základní znalosti jazyka Octave, proto tato problematika pokračuje v následující kapitole.

```

## Výpočet energie vyzářeného fotonu
Výpočet je dán vztahem


$$E_f = \frac{hc}{\lambda}$$
,

kde:
h - planckova konstanta
c - rychlost světla
 $\lambda$  - vlnová délka
E - energie fotonu
[E] = J

```

```

In [20]: # Definice konstant

h = 6.626176*(10^(-34));
c = 2.99792458*(10^8);
lambda = 254*(10^(-9));

```

```

In [21]: # Výpočet energie

E=(h*c)/lambda
E = 7.8208e-19

```

Obrázek 8.3.: Jupyter Notebook - výpočet energie vyzářeného fotonu

## Výpočet energie vyzářeného fotonu

Výpočet je dán vztahem

$$E_f = \frac{hc}{\lambda}$$

kde:

h - planckova konstanta  
c - rychlost světla  
 $\lambda$  - vlnová délka  
E - energie fotonu  
[E] = J

```

In [20]: # Definice konstant

h = 6.626176*(10^(-34));
c = 2.99792458*(10^8);
lambda = 254*(10^(-9));

In [21]: # Výpočet energie

E=(h*c)/lambda
E = 7.8208e-19

```

Obrázek 8.4.: Jupyter Notebook - výpočet energie vyzářeného fotonu (výstup)

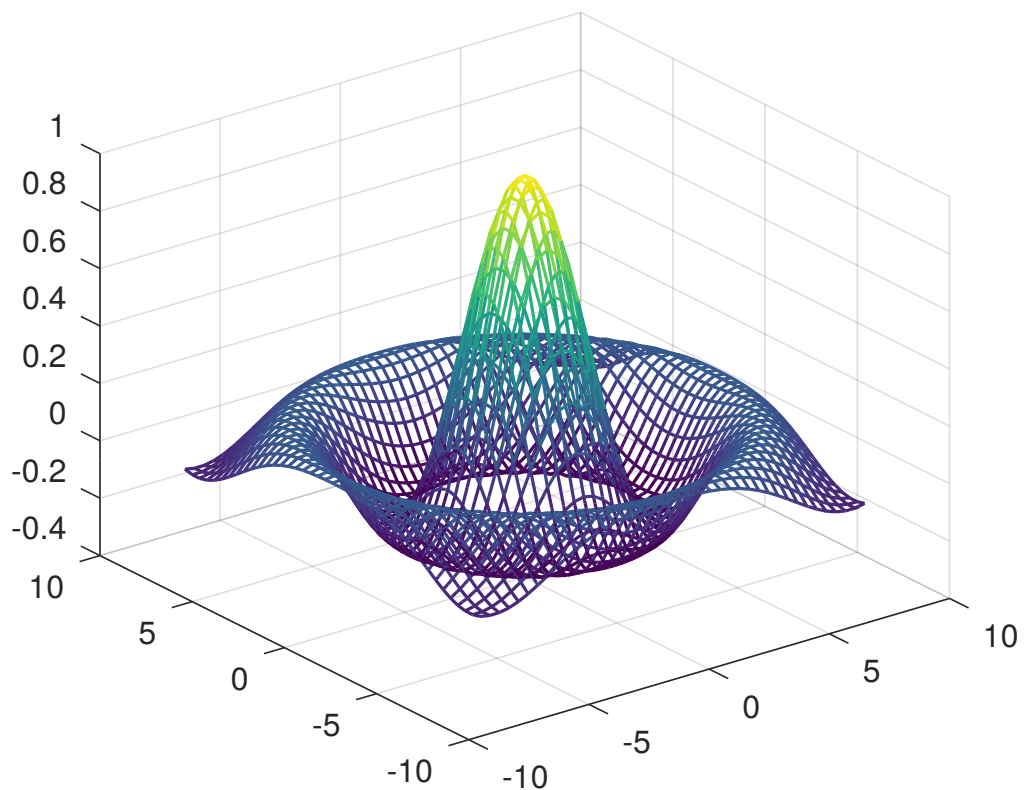
## 9. Základy jazyka Octave

Jazyk Octave, resp. GNU Octave, je v této práci využíván ke zpracování naměřených hodnot v Jupyter Notebooku. Byl zvolen z důvodu, protože je snadný a do velké míry kompatibilní s jazykem MATLAB, který je populární na většině vysokých škol. S oběma jazyky se rovněž můžeme setkat v některých firmách a na internetu mají velkou uživatelskou základnu, neboť jsou využívány mnoha vědci a studenty po celém světě. Čili je poměrně snadné dohledat požadovanou informaci.

Octave je open-source skriptovací jazyk, který umožňuje nejrůznější matematické výpočty, pracovat s maticemi, zobrazovat 2D a 3D grafy, provádět počítačovou simulaci, analýzu dat atd. Skriptovací se mu říká proto, že umožňuje napsat určitou posloupnost příkazů zvláště do souboru (skript), který lze pak samostatně spustit, což je výhodné, opakuje-li se nějaký kód nebo když se výpočet skládá z většího množství příkazů.

Tato kapitola obsahuje základní příkazy, které jsou nezbytné pro zpracování naměřených dat. Další funkce je možné dohledat v různých diskusních fórech na internetu nebo si je nastudovat z oficiální dokumentace GNU Octave [7].

Pozorný čtenář si asi položil otázku, proč nebyla věnována pozornost Jupyter Notebooku až po této kapitole. Je to z důvodu, aby bylo možné příklady z této kapitoly v Jupyteru procvičovat.



Obrázek 9.1.: Graf vygenerovaný v jazyku Octave [7]

## 9.1. Základní příkazy

- **Komentáře** se píše pomocí symbolu `#`<sup>1</sup>

```
# Toto je komentář
```

- Vytvoření skalární **proměnné** je jednoduché. Stačí napsat písmeno nebo nějaké slovo a rovnou jim můžeme přiřadit nějakou číselnou hodnotu, např.

```
x= 4;
prumer = 20;
c;
```

<sup>1</sup>lze také použít symbol `%`

- **Konstanty** - některé jsou integrovány přímo v Octave (např. Ludolfovo a Eulerovo číslo), ale většinu z nich je možné definovat, např.

```
# Planckova konstanta
h = 6.626176*(10^(-34));

# Rychlost světla
c = 2.99792458*(10^8);
```

Konstanty integrované v Octave:

```
# Ludolfovo číslo pi = 3,1416...
pi;

# Eulerovo číslo e = 2.7183...
e;
```


- **Zaokrouhlení čísla** - je-li v proměnné uložené desetinné číslo, lze jej zaokrouhlit funkcí `round`. Je možné zaokrouhlit také čísla obsažená ve vektoru. Tak např. mějme proměnnou `x` a vektor `V`, pak můžeme jednoduše psát

```
round(x)
round(V)

# Lze také napsat:
A = round(x*10)

# tím se zaokrouhlený výsledek přiřadí do nové proměnné A,
aniž by se musel předtím deklarovat
```

## Příklad

Následuje krátký příklad, který vysvětluje posloupnost příkazů a princip, kterým se v Octave používají. Po každém kroku je nutné stisknout tlačítko . Čtenář, který má alespoň základní znalosti z programování, může tyto body přeskočit.

1. Vytvoření proměnných a přiřazení hodnot:

```
x= 4;  
y = 20;
```

2. Součet proměnných (výsledek se přiřadí do proměnné c):

```
c = x + y;
```

3. Napsáním proměnné bez středníku dojde k vypsatí jejího obsahu (výsledek součtu) pod buňkou, ve které je deklarována:

```
c
```

4. Napsáním `c = 10` dojde k přepsání původní hodnoty v proměnné c novou hodnotou 10:

```
c = 10;
```

5. Výstup každého příkazu, který není ukončen středníkem, se vždy zobrazí pod příslušnou buňkou (je nutné vyzkoušet), např.:

```
c  
x + y  
c = x + y
```



## 9.2. Skalární operace

Operace	Název funkce/operátor	Příklad
součet	+	$c=a+b$
součin	*	$c=a*b$
podíl	/	$c=a/b$
mocnina	.^	$c=a.^2$
druhá odmocnina	sqrt()	sqrt(x)
$n$ -tá odmocnina	nthroot (číslo, exponent)	nthroot (10, 3)
přirozený logaritmus	log()	log(x)
exponenciální funkce	exp()	exp(x)
suma	sum()	sum(x)
funkce sinus <sup>2</sup>	sin()	sin(x)
absolutní hodnota	abs()	abs(x)

## 9.3. Operace s řádkovými vektory

Vektor je pro nás velmi důležitý prvek. Je to tzv. kontejner, do kterého lze ukládat naměřená data a potom s nimi pracovat jako s množinou (celkem). Např. je možné všechna čísla ve vektoru umocnit na druhou, nebo je zaokrouhlit a nebo z nich vypočítat aritmetický průměr. Hodnoty ve vektoru se neoddělují čárkou a desetinná čárka se v jazyku Octave píše jako tečka.

Tak např. chceme-li vytvořit vektory  $V1$  a  $V2$ , které budou obsahovat nějaké naměřené hodnoty, stačí napsat do libovolné buňky v Jupyteru tento řádek:

```
V1 = [5 8 16 0 2]
V2 = [1.2 8.5 0.2 6.8 9.4]
```

Následující tabulka obsahuje nejpoužívanější přehled operací s vektory:

Operace	Název funkce/operátor	Příklad
vytvoření vektoru		$V1 = [1 \ 2 \ 3]$
vektorový součet	+	$V2 = V1+V2$

<sup>2</sup>obdobně také cos, tan, cot, asin, acos...

Operace	Název funkce/operátor	Příklad
podíl vektoru skalárem	/	V3 = V1/2
podíl dvou vektorů	./	V4 = V1./V2
vynásobení vektoru skalárem	*	V5 = V1*10

Operace	Název funkce/operátor	Příklad
umocnění vektoru na druhou	.^	V5 =V1.^2
druhá odmocnina vektoru	sqrt	V5 =sqrt(V1)
$n$ -tá odmocnina	nthroot (vektor, exponent)	nthroot (V5, 3)
zobrazení $i$ -tého prvku ve vektoru		V5 =V1(5)
suma hodnot vektoru	sum	s = sum(V)
aritmetický průměr vektoru	mean	p = mean(V)
maximální hodnota vektoru	max	h = max(V)
minimální hodnota vektoru	min	d = min(V)
počet prvků ve vektoru	length	length(V)

## 9.4. Zápisy matematických rovnic

Matematické operace lze v Octave pochopitelně kombinovat a vytvářet tak různě složité rovnice, které pak slouží k různým výpočtům. Doporučuje se používat kulaté závorky, které určují prioritu operací. Tabulka níže obsahuje příklady některých zápisů.

Matematický vztah	Zápis v Octave
$y = 3x^2 - 16$	y=(3*x.^2)-16
$c = a^2 + b^2$	c=(a.^2)+(b.^2)
$y = \sum x$	y=sum(x)
$y = e^{x-2}$	y=exp(x-2)
$V = \frac{4}{3}\pi r^3$	V=4/3*pi*r.^3
$p = h\rho g$	p=h*r*g

Matematický vztah	Zápis v Octave
$c = \sqrt{a^2 + b^2}$	<code>c=sqrt((a.^2)+(b.^2))</code>
$y = \frac{\sqrt{3}}{2} + \left(\frac{x}{2}\right)^2$	<code>y=(sqrt(3)/2) + (x/2).^2</code>
$y = \sin x^2$	<code>y=(sin(x)).^2</code>

## 9.5. Příklad zpracování naměřených hodnot

Mějme vektor  $V3$  s naměřenými hodnotami proudu  $I_i$  a potřebujeme zjistit průměrnou hodnotu naměřeného proudu a chybu měření  $\sigma$ . Postup bude následující:

1. Výpočet aritmetického průměru naměřeného proudu:  $\bar{I} = \frac{1}{n} \sum_{i=1}^n I_i$ .

2. Výpočet odchylky od aritmetického průměru:  $\Delta = \bar{I} - I_i$ .

3. Výpočet chyby aritmetického průměru:  $\sigma = \sqrt{\frac{\sum_{i=1}^n (\Delta)^2}{n(n-1)}}$ .

4. Hodnota naměřeného proudu bude  $I = (\bar{I} \pm \sigma)$ .

## Výpis kódu

```
V3 = [1.2 8.5 0.2 6.8 9.4 3.2 8.9 1.7 6.8 8.8 2.3 1.7 9.1 0.5 0.1  
3.9 1.8 4.2 5.7 3.0];  
  
# Výpočet aritmetického průměru proudu  
prumerV3 = mean(V3)  
  
# Výpočet odchylek od střední hodnoty (vytvoříme nový vektor V4)  
V4 = prumerV3-V3;  
  
# Umocnění vektoru V4 na druhou  
V4ctverec = V4.^2;  
  
# Suma odchylek vektoru V4  
sumaV4 = sum(V4ctverec);  
  
# Zjištění počtu čísel vektoru V4  
n = length(V4)  
  
# Výpočet chyby aritmetického průměru (rozptyl od skutečné  
hodnoty)  
chybaMereni = sqrt(sumaV4/(n*(n-1)))
```

## Výpis pod buňkou:

```
prumerV3 = 4.3900  
n = 20  
chybaMereni = 0.73938
```

Naměřený proud tedy je  $I = (4.39 \pm 0.74) \text{ A}$

Z výše uvedeného výpočtu je patrné, že se jedná o jednodušší způsob než na kalkulačce. Pro někoho může být tento postup přehlednější než v MS Excelu, protože v něm jsou výpočty „skryty“ za buňkami. A v neposlední řadě se jedná o výsledek, kterému můžeme důvěřovat, neboť byl vypočítán pomocí profesionálního matematického nástroje.

Celý výpočet se dá pochopitelně zkrátit a Jupyter vypíše stejný výsledek, např.

```
V3 = [1.2 8.5 0.2 6.8 9.4 3.2 8.9 1.7 6.8 8.8 2.3 1.7 9.1 0.5 0.1
3.9 1.8 4.2 5.7 3.0];

# Výpočet chyby aritmetického průměru (rozptyl od skutečné
hodnoty)
chybaMereni = sqrt((sum(((mean(V3))-V3).^2))/((length(V4))*
((length(V4))-1)))
```

Nicméně tento postup je nepřehledný a může generovat chyby (snadno se přehlédneme). Proto se doporučuje používat raději předchozí sekvenční zápis.

## 9.6. Vytvoření 2D grafu

Octave umí zobrazit dvojrozměrné i třírozměrné grafy. K dispozici je několik způsobů jak to udělat. Zobrazení grafu se provádí pomocí funkce `plot`. Níže jsou uvedeny tři nejpoužívanější postupy.

### 9.6.1. Zobrazení hodnot pomocí hodnot z vektorů

Tato metoda je nejčastější a používá se tehdy, máme-li naměřené hodnoty obou fyzikálních veličin.

Tak např. mějme řádkové vektory `A` a `B`, které obsahují nějaké naměřené hodnoty. Vektor `A` představuje hodnoty na ose `x` a vektor `B` hodnoty na ose `y`:

```
A = [0 2 3 4 5 6 7 8 9 10];
B = [0 25 30 31 55 43 71 81 63 100];
```

Graf se potom zobrazí jednoduše tímto příkazem:

```
plot(A,B)
```

Aby měla zobrazená přímka silnější linku, je možné použít příkaz `LineWidth` a za ním celé číslo, které udává sílu linky:

```
plot(A,B,'LineWidth',2)
```

Nyní následují estetické úpravy grafu. Např. osy grafu se popisují pomocí příkazů `xlabel` (resp. `ylabel`):

```
xlabel('U [V]')
ylabel('I [A]')
```

Jelikož se fyzikální veličiny píší *kurzívou*, je nutné ještě připsat parametr `\it`. To co má být *kurzívou* se uzavírá do složených závorek `{}`:

```
xlabel('{\itU} [V]')
ylabel('{\itI} [mA]')
```

Dolní nebo horní indexy veličinám přidáme např. tímto zápisem:

```
xlabel('{\itU}_2 [103 V]')
ylabel('{\itI}_2 [mA]')
```

Je-li vyžadováno, aby popisy os byly umístěny na jejich konci, je nutné připsat ještě pár řádků kódu. Není-li to vyžadováno, může se tento bod přeskočit. Čísla v hranatých závorkách představují pozici na osách x-y-z. Tento parametr je nutné odzkoušet, neboť každý graf má jiný rozsah os:

```
# Pro osu x
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [9 0 0])
# Pro osu y
ylabh = get(gca,'yLabel');
set(ylabh,'Position',get(ylabh,'Position') + [0 95 0])
```

Pro zobrazení mřížky se používá příkaz `grid`:

```
# Hrubší mřížka
grid on

# Jemnější mřížka
grid minor on
```

Legenda se umísťuje do grafu príkazom `legend`:

```
# Standardní umístění do pravého horního rohu
legend ('y=f(x)');

# Nebo umístění vpravo dole
legend ('y=f(x)', 'location', 'southeast');

# A nebo umístění vně pod graf
legend ('y=f(x)', 'location', 'southoutside');

# Je-li zobrazováno více křivek, tak např.
legend ('y1=f(x)', 'y2=f(x)', 'y3=f(x)', 'y4=f(x)');
```

Potřebuje-li graf omezit rozsah os<sup>3</sup>, použijte se funkce `xlim` (resp. `ylim`):

```
# Pro osu x - budou zobrazena data pouze v intervalu [0;8]
xlim([0 8])

# Obdobně pro osu y
ylim([0 80])
```

Nakonec se graf vyexportuje jako *obrázek do souboru* příkazem `print`, aby ho bylo možné vložit do protokolu. Upozorňujeme, že kvalita vyexportovaného obrázku je výrazně vyšší než kvalita zobrazovaného obrázku v Jupyteru (zřejmě kvůli výkonu). Exportovat lze do formátu SVG, PNG, JPG a dalších. Doporučujeme vektorový formát **SVG**, který má nejvyšší možnou kvalitu.


```
print graf1.svg
```

V případě exportu do formátu PNG doporučujeme rozlišení obrázku 600 dpi, jinak mohou být grafy v protokolu rozostřené. Export se provádí tímto příkazem:

```
print graf1.png -r600
```

---

<sup>3</sup>např. je-li pravá strana grafu nevyužitá nebo když jsou v grafu data, která v něm nemusejí být

Následující rám obsahuje shrnutí výše uvedeného postupu. Kód lze vložit do libovolné buňky v Jupyteru a stisknutím tlačítka  dojde k zobrazení grafu, který je na obrázku [9.2](#).

```
# Vektory s naměřenými daty
A = [0 2 3 4 5 6 7 8 9 10];
B = [0 25 30 31 55 43 71 81 63 100];

# Zobrazení grafu
plot(A,B)

# Omezení rozsahu os
xlim([0 8])
ylim([0 80])

# Popisy os
xlabel('\itU}_2 [V]')
ylabel('\itI}_2 [mA]')

# Posunutí popisu na konec osy
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [7.5 0 0])

ylabh = get(gca,'yLabel');
set(ylabh,'Position',get(ylabh,'Position') + [0 75 0])

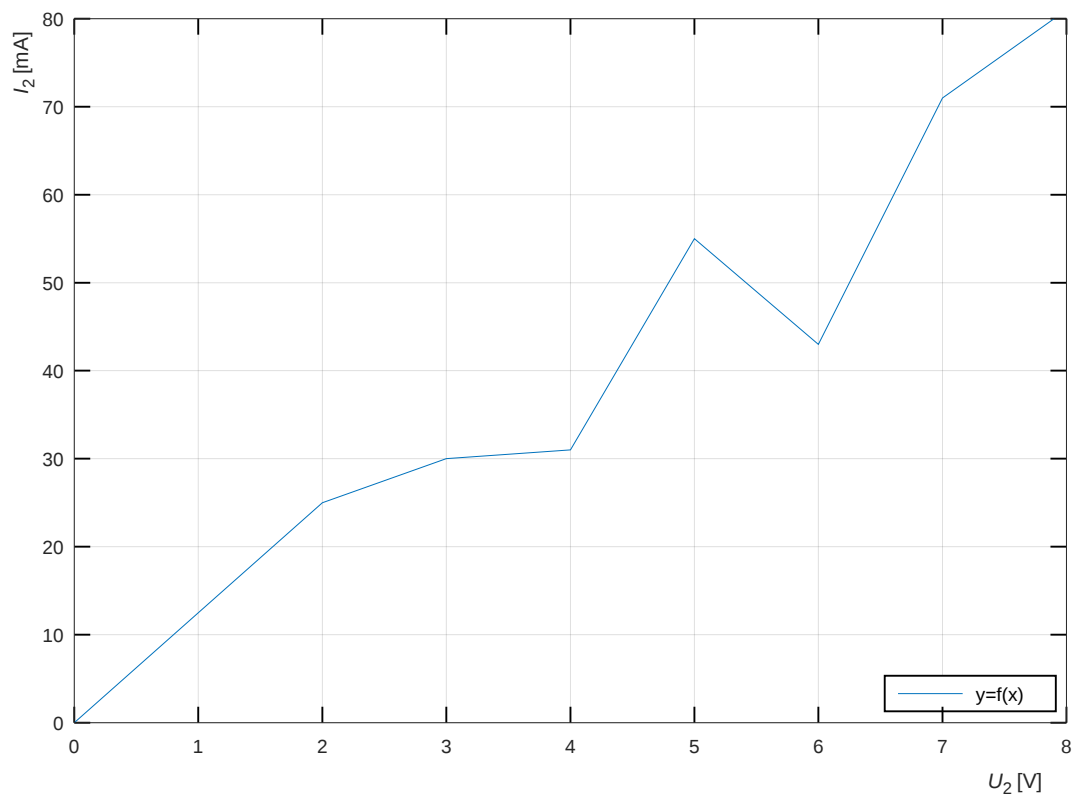
# Uzavření grafu do rámečku
box on

# Zapnutí mřížky
grid on
```



```
# Zobrazení legendy
legend ('y=f(x)', 'location', 'southeast');

# Export grafu do souboru
print graf1.svg
```



Obrázek 9.2.: Zobrazení grafu pomocí hodnot z vektorů

### 9.6.2. Zobrazení pomocí matematické funkce

Postup je obdobný jako v předchozím případě, ale tentokrát místo vektoru  $B$  budou na ose  $y$  zobrazeny funkční hodnoty  $f(x)$ . Mějme exponenciální funkci

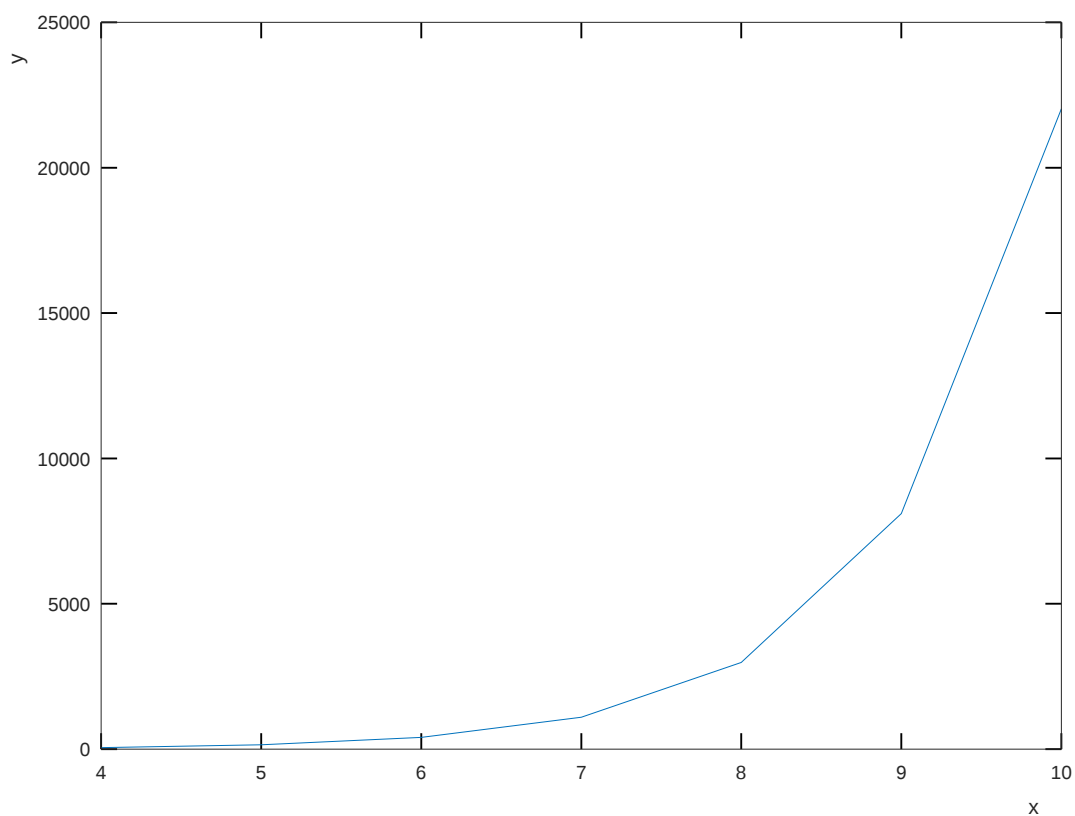
$$f : y = e^x,$$

potom zobrazení grafu bude realizováno tímto blokem příkazů (graf je na obrázku 9.3.):

```
# Naměřená data
x = [4 5 6 7 8 9 10];

# Funkční předpis
y = exp(x);

# Zobrazení grafu
plot(x,y,'LineWidth',2)
```



Obrázek 9.3.: Graf funkce  $y = e^x$

### 9.6.3. Zobrazení aproximace naměřených dat

Vysvětlení výpočtu aproximace je nad rámec této bakalářské práce, proto bude vysvětlen jenom její princip a stručně popsáno, jak proloženou přímkou (nebo křivku<sup>4</sup>) v grafu zobrazit.

Předpokládejme, že máme dostatečné množství naměřených hodnot hodnot a potřebujeme zjistit matematický předpis (funkci), podle kterého se hodnoty měnily, resp. jakou mají závislost (lineární, exponenciální atp.). Tato závislost se většinou zjišťuje aproximací metodou nejmenších čtverců. Aproximace je výpočet, díky kterému se získá nějaký funkční předpis, v tomto případě rovnice přímky. Zahrnutím této rovnice do grafu se poté zobrazí přímka, která bude procházet naměřenými body (některými přímo a některým se bude jenom přibližovat kvůli nepřesnosti měření) - viz obr. 9.4.

Z této přímky a její rovnice je potom možné zjistit nebo odhadnout řadu skutečností. Tak např. že rozpad Barya-137 za čas exponenciálně klesá. Nebo je možné odhadnout chování nějakého systému či dokázat průběh nějakého fyzikálního děje. Čím více naměřených hodnot aproximujeme, tím věrněji bude přímka popisovat nějakou závislost.

Rovnici přímky umí vypočítat některé programy, např. MATLAB, ten ale něco stojí. Nemáme-li takový program, je možné rovnici vypočítat v Jupyter Notebooku pomocí open-source jazyka Octave, který je s MATLABem téměř totožný.

Jeden takový příklad (skript) je v příloze E. Obsahuje výpočet koeficientů pro aproximaci dat, která byla naměřena při rozpadu radioaktivního Barya-137 (aktivita v závislosti na čase) v laboratoři na Jihočeské Univerzitě v Českých Budějovicích.

Z klasické fyziky víme, že vztah pro aktivitu radioaktivního nuklidu je dán vztahem

$$A(t) = A_0 e^{-\lambda t}. \quad (9.1)$$

Očividně se jedná o exponenciální funkci. Pro získání rovnice (resp. funkce, předpisu), podle které se Baryum rozpadalo, bylo zapotřebí do rovnice (9.1) vypočítat koeficienty  $A_0$  a  $\lambda$  metodou nejmenších čtverců a využitím maticového počtu. Do-

---

<sup>4</sup>dále bude uváděno jen „přímka“

sazením hodnot koeficientů do rovnice (9.1) byl poté získán vztah

$$A(t) = 16e^{-0,27t}, \quad (9.2)$$

což je rovnice exponenciální křivky, která byla hledána pro náš graf.

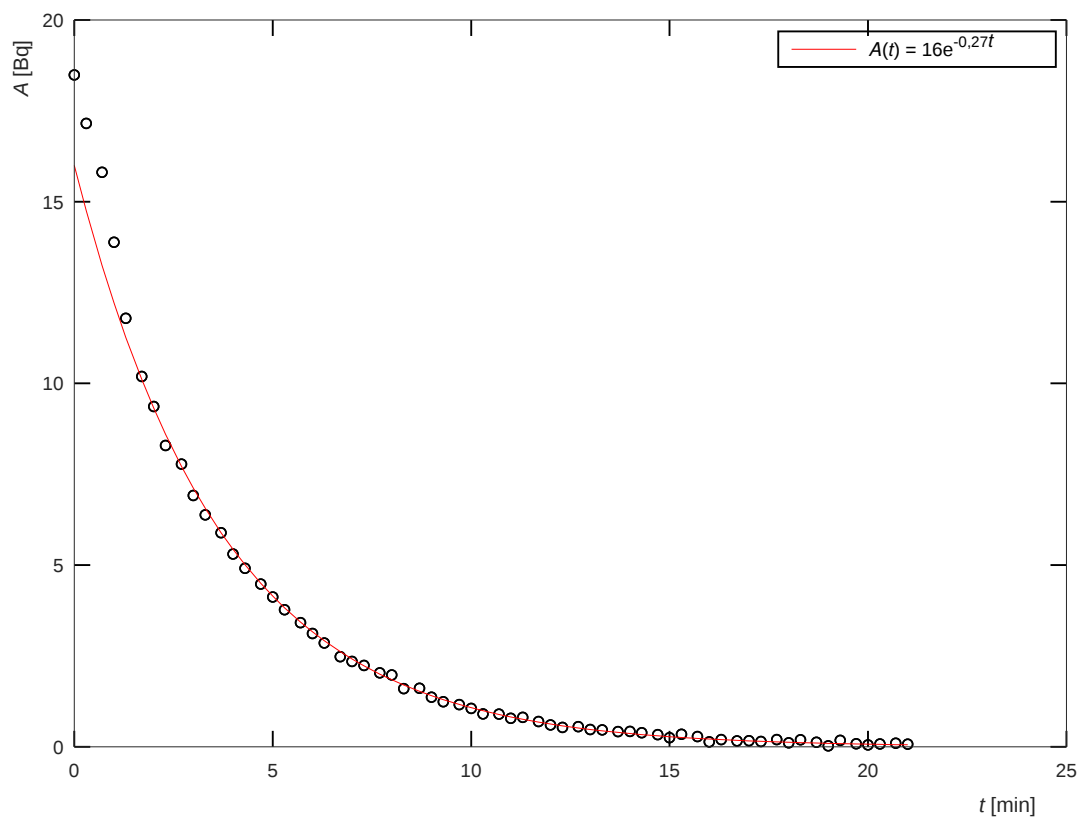
Nyní stačí v Jupyteru napsat kód, který bude v grafu zobrazovat hodnoty naměřené detektorem a průběh exponenciální křivky. Kód tedy bude obsahovat tyto dva specifické zápisy (celý kód je v příloze D):

```
# Zobrazení naměřených dat
scatter(t,A)

# Zoprazení proložené křivky
At=16*exp(-0.27*t);
plot(t,At)
```

kde  $t$  je naměřený čas,  $A$  je naměřená aktivita a  $At$  je funkční hodnota exponenciální křivky.

Dalším příkladem může být graf 5.2 na straně 42, který je proložen Gaussovou křivkou.



Obrázek 9.4.: Aktivita Barya-137 v závislosti na čase

# 10. Vytvoření dokumentu v editoru CoCalc

CoCalc nabízí dva editory, ve kterých lze vytvářet dokumenty. První umožňuje vytvářet dokumenty výhradně v jazyku  $\text{\LaTeX}$ <sup>1</sup> a druhý umožňuje kombinovat jazyky Markdown a  $\text{\LaTeX}$  dohromady. Každý jazyk má své výhody a nevýhody a každý poskytuje jinou úroveň grafického výstupu.

Zatímco  $\text{\LaTeX}$  umožňuje získat profesionální vzhled dokumentu, jeho syntaxe může být pro někoho příliš složitá. Na proti tomu Markdown vyniká svou jednoduchostí, ale musíme se spokojit s méně reprezentativním vzhledem. Domníváme se, že v našem případě by bylo nejvýhodnější umět od obou trochu, a to ze čtyř důvodů:

1. Kombinace jazyků Markdown a  $\text{\LaTeX}$  je výhodná - získáme rychlost a pohodlí díky Markdownu a kvalitní grafický výstup *matematických rovnic* díky  $\text{\LaTeX}$ u.
2. Markdown a  $\text{\LaTeX}$  lze kombinovat také v Jupyter Notebooku, SageWorkshetu a chatovacím okně.
3. Laboratorní protokol není obsáhlý dokument, tudíž se v něm dá ohlídat číslování obrázků a tabulek. Rovněž u něj není vyžadována vysoká typografická úroveň co se týče kvality písma. Čili není nutné psát protokol pouze v  $\text{\LaTeX}$ u.
4. Naučit se Markdown je výhodná investice, neboť je velmi rozšířen.

---

<sup>1</sup>vyslovuje se „latek“ nebo „latech“

Z výše uvedených důvodů se nebudeme  $\text{\LaTeX}$ u věnovat podrobněji. Budou zmíněny jen příkazy, které jsou potřebné pro sázení matematických vzorců. Přesto bychom rádi zmínili jeho přednosti, neboť se jedná o nástroj, který CoCalc plně podporuje a je pravděpodobné, že ho časem začnete používat. Zvědavé čtenáře odkážeme na doporučenou literaturu [6].

## 10.1. Markdown

Markdown je odlehčený *značkovací jazyk*, který pomocí jednoduchých formátovacích značek umožňuje formátovat prostý text. Díky nim lze zvýrazňovat texty, upravovat nadpisy, vytvářet seznamy, hypertextové odkazy atd.

Jak bylo naznačeno, jazyk Markdown uvítají především méně nároční uživatelé, neboť je jednoduchý a lze v něm napsat dokument za několik málo hodin. Je navržený tak, že při tvorbě textu stačí „prostě psát“ a nic neřešit.

Hlavními tvůrci jsou John Gruber a Aaron Swartz. Původním cílem bylo vytvořit jazyk, díky kterému bude možné převést prostý text do kódu HTML. Díky tomu lze Markdown kombinovat i s tímto jazykem.

Mezi uživateli je velmi rozšířen, neboť je využíván v mnoha službách, např. Wordpressu, Drupalu, DokuWiki, GitHubu a dalších. Rovněž se používá pro psaní knih nebo poznámek a v jiných programovacích jazycích, např. v PHP a Pythonu.

### Výhody

- jednoduchá a rychlá použitelnost
- čitelná syntaxe
- velká uživatelská základna
- rozšířenost na různých platformách
- možnost kombinovat s jazykem HTML

### Nevýhody

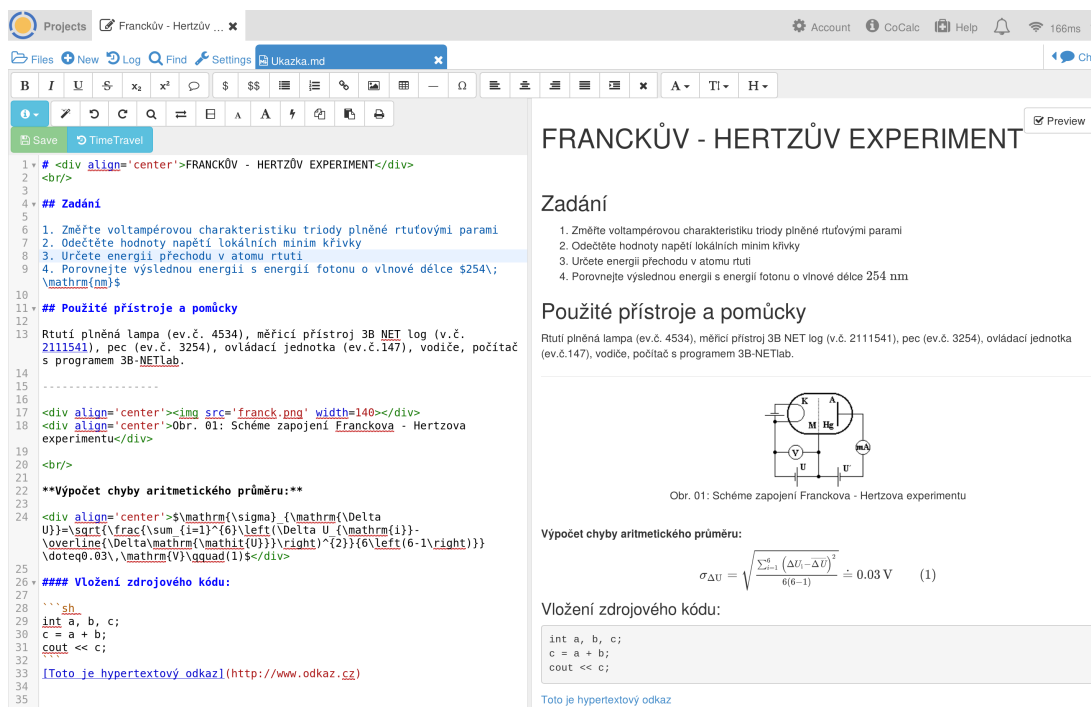
Pro naše potřeby jeho nevýhoda spočívá v udržení jednotného vzhledu dokumentu. Markdown sice obsahuje mechanismy, kterými lze jednotného stylu dosáhnout, ale není automatizovaný jako  $\text{\LaTeX}$ . Jedná se především o číslování obrázků, tabulek a rovnic, kontrolování mezer, vkládání křížových odkazů atd. To vše musíme dělat sami, což vyžaduje důkladnější kontrolu. Fonty rovněž nejsou kvalitní jako v  $\text{\LaTeX}$ u. Z tohoto pohledu se Markdown nehodí pro vytváření obsáhlejších dokumentů a tam, kde je vyžadována vyšší typografická kvalita písma



(např. knihy nebo diplomové práce). Nicméně s trochou snahy i v Markdownu lze vytvořit pěkný dokument, a to je naším účelem.

## Syntaxe


Značky Markdownu není nutné znát nazpaměť, protože editor CoCalc poskytuje nástrojový panel, se kterým se pracuje podobně jako v textovém editoru MS Word. Tak např. pro tučné písmo stačí označit text (nebo blok textu) a kliknout na tlačítko **B**. Editor obklopí text příslušnou značkou a změnu zobrazí v pravém *náhledovém* okně.



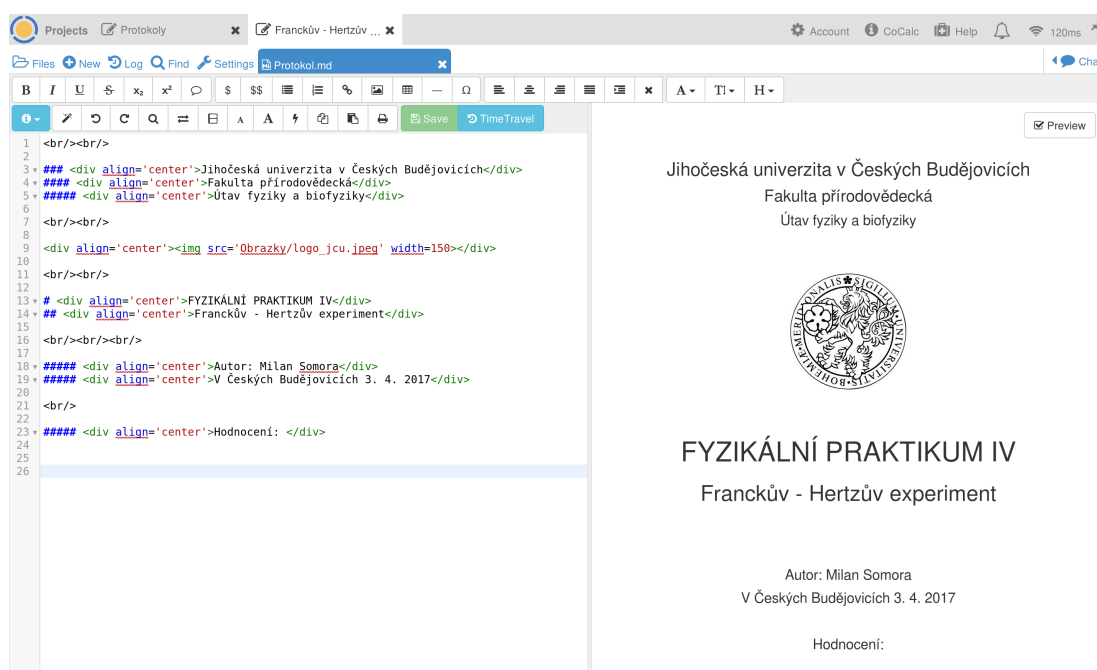
Obrázek 10.1.: Editor CoCalc - Markdown a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  v praxi

## Kombinace s jazykem HTML

Přestože má Markdown spoustu předností, není všemocný a někdy je nutné doplnit nějakou funkci pomocí jazyka HTML. Proto CoCalc zahrnuje do nástro-

jového panelu rovněž nejpoužívanější příkazy HTML. Např. pro vložení obrázku stačí kliknout na tlačítko  a zadat základní parametry jako je cesta k obrázku, jeho rozměr (stačí jenom jednu stranu) a popis. To je vše. Editor obklopí text příslušným HTML kódem a změnu zobrazí v pravém okně. Je nutné upozornit, že HTML kód nesmí obklopit Markdown značku, ale pouze prostý text, jinak se text špatně zobrazí.

V případě potřeby jsou v příloze A uvedeny další značky a příkazy, které CoCalc na panelu nemá.




Obrázek 10.2.: Editor CoCalc - vytvoření úvodní stránky protokolu v jazyce Markdown a HTML

## Sázení matematických rovnic v jazyku $\text{\LaTeX}$

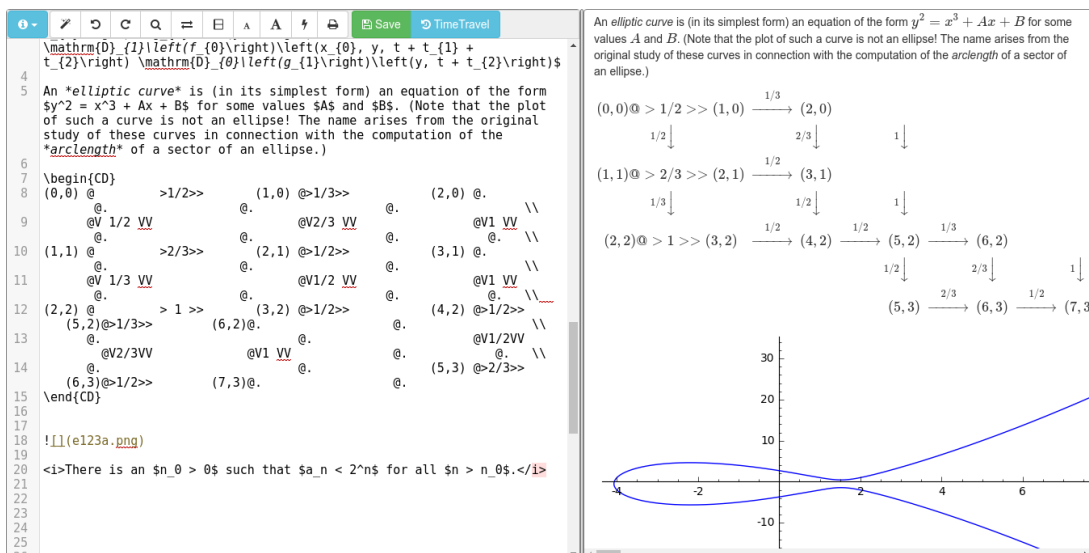
V jazyce Markdown matematické rovnice sázet nejdou, ale díky vývojářům CoCalc je možné do editoru zapisovat rovnice pomocí jazyka  $\text{\LaTeX}$ . Získáme tak dokument, který bude obsahovat matematické rovnice ve vysoké typografické kvalitě. O sázení rovnic v  $\text{\LaTeX}$ u bylo pojednáno v sekci 7.

Na obrázku 10.1 jsou jazyky Markdown,  $\text{\LaTeX}$  a HTML v akci. Z obrázku je zřejmé, že se nejedná o nic složitějšího.

Na obrázku 10.2 je vytvořena titulní strana laboratorního protokolu. A na obrázku 10.3 je ukázka, že Markdown umí zvládnout i složitější strukturu dokumentu. K tomu jsou však zapotřebí hlubší znalosti - viz dokumentace CoCalc [2].

Je nutné upozornit, že v editoru CoCalc je při vytváření dokumentu zobrazováno bezpatkové písmo Arial, ale po exportu do formátu PDF (tlačítko ) bude mít dokument patkové písmo Times New Roman (pravděpodobně chyba, kterou CoCalc doufejme odstraní).

Na internetu lze dohledat další nástroje, které mohou více usnadnit práci. Např. generátor tabulky [4] nebo převod MS Word dokumentu do Markdownu [11].



Obrázek 10.3.: Editor CoCalc - složitější struktura dokumentu [1]

## 10.2. L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X je série příkazů, kterými lze vytvářet dokumenty ve vysoké typografické kvalitě a s jednotným profesionálním vzhledem. Je velmi populární zejména v oborech jako je matematika, fyzika, informatika, ale i v humanitních oborech. Obecně je považován za nejlepší nástroj pro sazbu matematických vzorců. U studentů získal oblibu jako nástroj pro psaní diplomových prací a využívá se také pro běžnou sazbu, např. psaní knih.

Dokument v L<sup>A</sup>T<sub>E</sub>Xu se píše formou prostého textu, podobně jako v jazyku HTML. V CoCalc se musí vytvořit soubor s příponou \*.tex, po jehož otevření se zobrazí editor, který funguje obdobným způsobem jako editor pro psaní v jazyku Markdown (viz obr. 10.4).

### Stručná historie

L<sup>A</sup>T<sub>E</sub>X je přesněji řečeno série maker, která ovládá základní program T<sub>E</sub>X, který v 70. letech vytvořil profesor Donald Ervin Knuth, neboť sám nebyl spokojen s tím, že školní nakladatelství sázelo skripta určená studentům s chybami, a to především v matematických vzorcích.

Proč tedy nepsat rovnou v T<sub>E</sub>Xu? Protože T<sub>E</sub>X je uživatelsky nepřívětivý nízkourovňový jazyk, který vyžaduje znalosti z programování. Proto byla v roce 1985 panem Lamportem vytvořena sbírka maker nesoucí jméno L<sup>A</sup>T<sub>E</sub>X, díky kterým může uživatel bez znalosti programování vytvořit rozmanitý dokument za několik dní v knižní kvalitě.

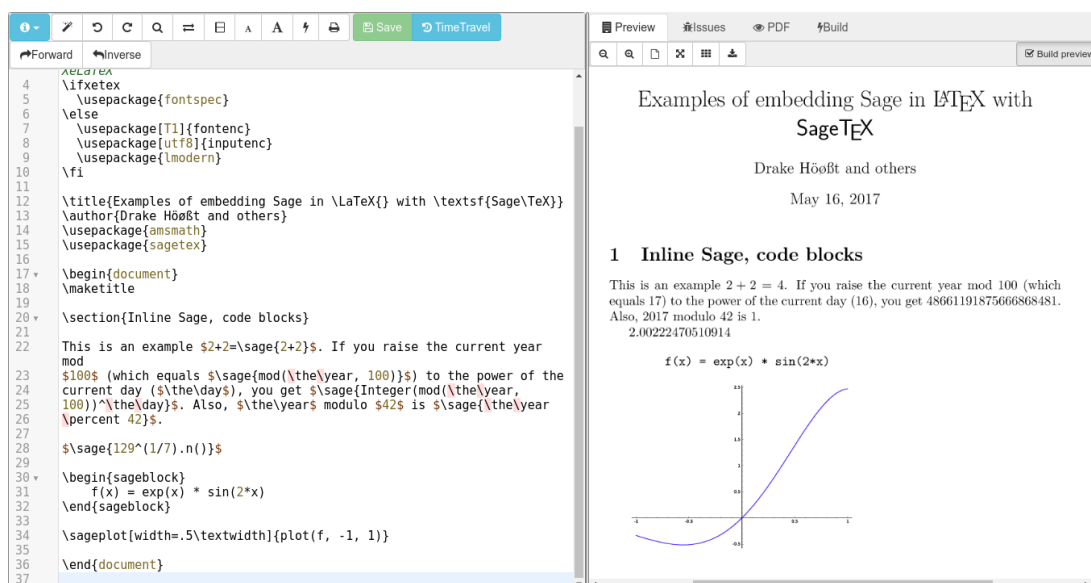
CoCalc umožňuje plnou integraci L<sup>A</sup>T<sub>E</sub>Xu a díky vývojářům CoCalc ho lze kombinovat s dalšími pokročilými funkcemi jako je např. SageTeX<sup>2</sup>.

### Jednotný vzhled dokumentu

První silnou stránkou L<sup>A</sup>T<sub>E</sub>Xu je, že umí dodržet jednotný formát dokumentu v tom smyslu, že se autor nemusí starat o typ fontu, velikosti textu a nadpisů,

---

<sup>2</sup>umožňuje vkládat do dokumentu výpočty přímo ze Sage Worksheetu



Obrázek 10.4.: Vytváření dokumentu v jazyku  $\text{\LaTeX}$  [1]

mezery mezi odstavci, číslování stránek a kapitol, umístění obrázků, jednotný formát tabulek atd. To vše díky profesionálně předdefinovanému vzhledu, jehož styl je možné změnit pouze jedním příkazem.

$\text{\LaTeX}$  automaticky člení text, čísluje obrázky, tabulky, rovnice, citace, křížové odkazy, umí vytvořit obsah nebo rejstřík dokumentu a mnohem víc.

## Vysoká typografická úroveň

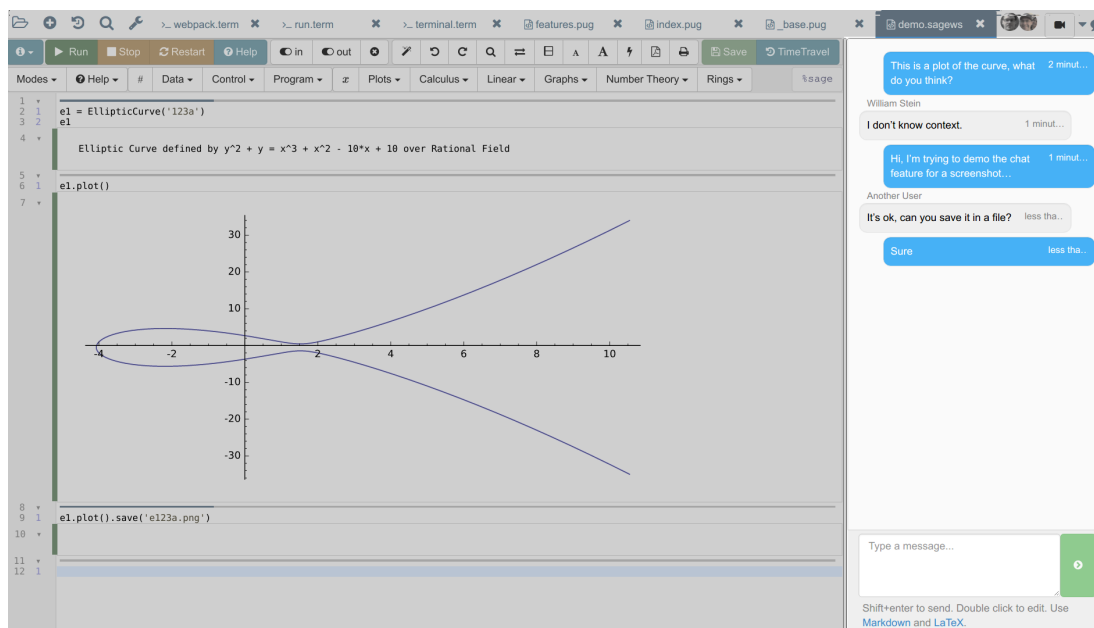
Druhou silnou stránkou jsou jeho fonty, které mají vysokou typografickou kvalitu. Typografie je obor, který se zabývá tvarem písma a jeho dalšími vlastnostmi. Snaží se o co nejlepší čitelnost textu, aby se v něm čtenář dobře orientoval a také aby působil esteticky. Např. některé fonty v  $\text{\LaTeX}$ u obsahují ligatury.

Typografie rovněž zavádí pravidla, jejichž dodržováním dokument působí profesionálním dojmem. Typografickým pravidlům byla věnována kapitola 6.

# 11. Další možnosti v CoCalc

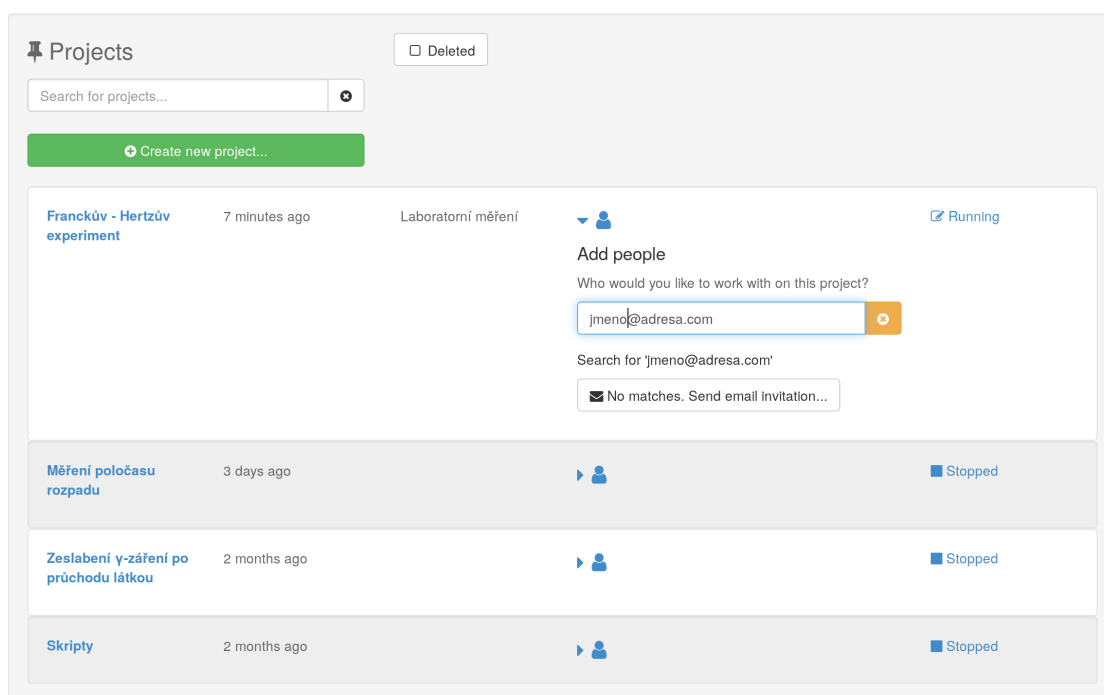
## 11.1. Týmová práce

CoCalc umožňuje pracovat v týmu. Spolupráce na projektu probíhá s ostatními uživateli v reálném čase (lze vidět, jak se jejich kurzory pohybují). Spolupracovníci mohou upravovat soubory stejně jako Vy. Myšlenky lze mezi sebou sdílet pomocí chatovacího okna, které je k dispozici v každém otevřeném panelu a otvírá se ikonou v pravém horním rohu (obr. 11.1). CoCalc rovněž umožňuje streamovat video.



Obrázek 11.1.: Chatovací okno - otevírá se ikonou v pravém horním rohu [1]

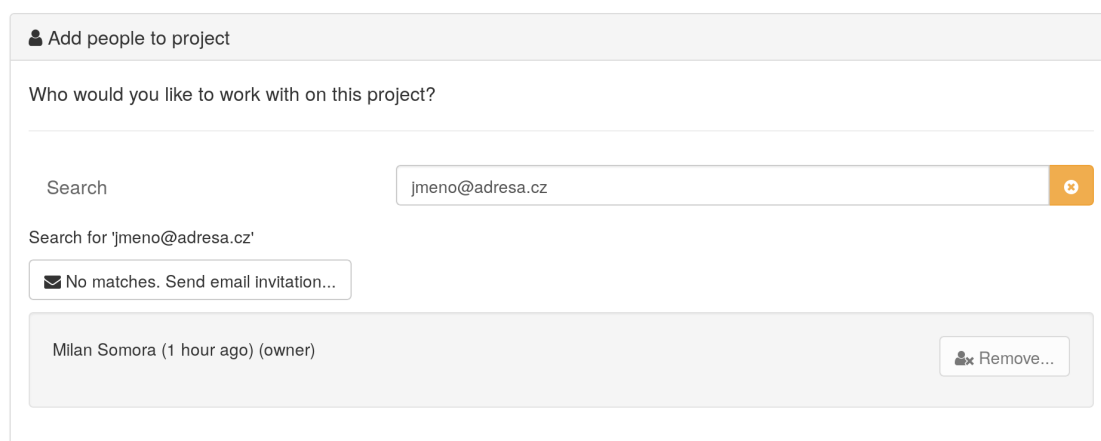
Princip týmové práce spočívá v tom, že se do projektu přidají spolupracovníci pomocí jejich emailové adresy. Lze to provést ze dvou míst - na stránce projektů (obr. 11.2) nebo v nastavení projektu (Settings and configuration - obr. 11.3). Uživatel, který se chce stát spolupracovníkem, se musí v CoCalc zaregistrovat. Nezaregistrovanému uživateli lze poslat pozvánku do emailové schránky, která ho vyzývá, aby se zaregistroval (obr. 11.4).



Obrázek 11.2.: Stránka projektů - přidání dalšího uživatele do projektu

## 11.2. Možnosti výuky

Potenciál služby CoCalc je natolik velký, že v ní lze rovněž organizovat kurz. Postup je obdobný jako v předchozích případech - založit projekt a v něm vytvořit soubor s příponou (.course). Po jeho otevření se otevře okno, ve kterém je možné:

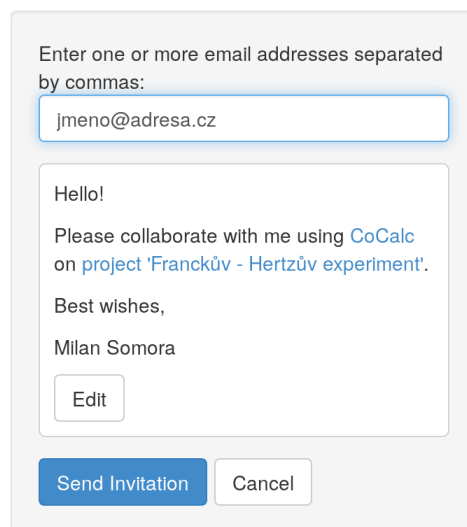


Obrázek 11.3.: Stránka Settings and configuration - přidání dalšího uživatele do projektu

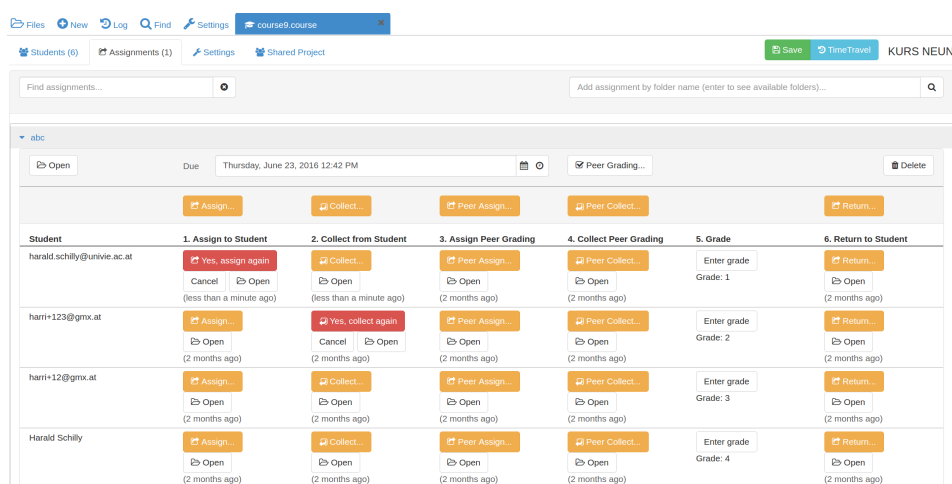
- přidávat studenty podle emailové adresy
- dělat si ke studentům poznámky
- vytvářet úkoly a podklady pro studenty
- prohlížet studentovy projekty
- vytvářet společné projekty
- chatovat se studenty
- kontrolovat a klasifikovat práci studentů a další

Jedno z oken, které umožňuje správu kurzu, je na obrázku 11.5. Nicméně vedení kurzů se v CoCalc považuje za doplňkovou službu, čili je nutné koupit si předplatné, které k dnešnímu dni stojí sedm dolarů měsíčně. Toto předplatné zahrnuje další výhody v podobě větší rychlosti serveru, většího výpočetního výkonu a většího prostoru v cloudu.







Obrázek 11.4.: Přidání dalšího uživatele do projektu - odeslání pozvánky



Obrázek 11.5.: Správce kurzu [1]

## 11.3. Dokumentace a podpora

Vzhledem k tomu, že se CoCalc dříve jmenovala SageMath Cloud, lze najít různé návody i pod tímto názvem.

- Souhrnné informace o CoCalc a mnoho dalšího lze získat kliknutím na ikonu  CoCalc, jak bylo popsáno ve 4. kapitole.
- Uživatelskou podporu a dokumentaci pro CoCalc získáme kliknutím na ikonu  Help.
- CoCalc tutoriál: <https://tutorial.cocalc.com/>.
- CoCalc blog vedený společností SageMath Inc.: <https://blog.sagemath.com/>.
- SageTex: <https://doc.sagemath.org/html/en/tutorial/sagetex.html>.

V případě využití modulu SageMath, který tvoří jednu z nejsilnějších stránek CoCalc, uvádíme ještě tyto odkazy:

- Český návod: <http://www.math.muni.cz/~plch/diplomky/Sage.pdf>, případně návod v anglickém jazyce: <https://doc.sagemath.org/html/en/tutorial/>.
- Ukázkové worksheets: <http://user.mendelu.cz/marik/sage/>.
- Příklady k procvičení: <http://user.mendelu.cz/marik/akademie/index.php#sage>.
- Domovská stránka SageMath: <http://www.sagemath.org/>.
- SageMath Feature Tour: <http://www.sagemath.org/tour.html>.
- Podpora, kterou lze získat v Google skupině SageMath Support Group na adrese <https://groups.google.com/forum/#!forum/sage-support>.

## 12. Shrnutí a závěr

Máme za sebou poměrně velké množství informací, které by bylo vhodné shrnout.

Nejdříve byla představena služba CoCalc a její možnosti. Bylo naznačeno, v čem spočívá její síla, jak může být užitečná z hlediska zpracování vědecko-technických dokumentů a kde může být její další potenciál (týmová práce, možnosti výuky ad.). Následovalo vysvětlení, jak v CoCalc založit projekt, jak v něm pracovat se složkami a soubory a jak se v něm orientovat.

Dále byla vysvětlena podstata laboratorního protokolu - proč se vytváří, jakou má osnovu, jaké jsou na něj požadavky a v této souvislosti byla shrnuta základní pravidla (typografie, tabulky, grafy), která by měl respektovat každý laboratorní protokol. Rovněž bylo vysvětleno, jak v protokolu správně zapisovat matematické rovnice a fyzikální veličiny a jejich jednotky s využitím jazyka  $\text{\LaTeX}$ .

Následovala nauka, jak zpracovávat naměřená data pomocí nástroje Jupyter Notebook s využitím jazyka Octave. Rovněž bylo popsáno, jak vytvářet grafy a jak je vyexportovat do formátu SVG, aby je bylo možné použít v protokolu. Nakonec bylo vysvětleno, v jakém editoru vytvářet laboratorní protokol a jaké jazyky k tomu používat (Markdown a  $\text{\LaTeX}$ ).

V této fázi by měl být čtenář na základě znalostí z předchozích kapitol schopen samostatně vypracovat laboratorní protokol. Součástí této práce jsou tři vypracované laboratorní protokoly z jaderné a částicové fyziky a další příklady, které lze v případě potřeby využít jako nápovědu. Tento obsah byl však kvůli velkému rozsahu uložen na disk CD, který je součástí tohoto výtisku. Protokoly a další příklady byly pochopitelně vypracovány v CoCalc, přesně jak je uvedeno v této práci. Disk obsahuje rovněž příklady z vybraných kapitol a příloh B-E, aby se daly snadno zkopírovat.

# Literatura

- [1] *Domovská stránka CoCalc* [online]. Dostupný z WWW: <https://CoCalc.com/>.
- [2] *Dokumentace CoCalc* [online]. Dostupný z WWW: <https://github.com/sagemathinc/CoCalc/wiki/Portal>.
- [3] FRIEDLER, Karel. *ČMSS - Python + Jupyter Notebook* [online]. Common.cz. Dostupný z WWW: <https://goo.gl/9s4eB2>.
- [4] *Markdown Tables Generator* [online]. Dostupný z WWW: [https://www.tablesgenerator.com/markdown\\_tables](https://www.tablesgenerator.com/markdown_tables).
- [5] KLEIN, Lukáš. *Základní zásady pro vypracování protokolu o měření* [online]. VŠ Báňská – Technická univerzita Ostrava. Dostupný z WWW: [http://fei1.vsb.cz/kat430/data/Tvorba\\_protokolu.pdf](http://fei1.vsb.cz/kat430/data/Tvorba_protokolu.pdf).
- [6] OETIKER, T. a kol. *Ne příliš stručný úvod do systému L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>* [online]. Verze 2.2. 1996. Dostupný z WWW: <http://www.math.muni.cz/~plch/vyuka/lshort2e-cz.pdf>.
- [7] W. EATORN, John a kol. *GNU Octave* [online]. 14. vyd. Boston: Free Software Foundation, 2017. MA 02110-1301–1307. Dostupný z WWW: <https://www.gnu.org/software/octave/octave.pdf>.
- [8] Wikipedia: *Vědecký zápis čísel* [online]. Dostupný z WWW: <https://goo.gl/2wKWrv>.
- [9] Wikipedia: *Ligatura (typografie)* [online]. Dostupný z WWW: [https://cs.wikipedia.org/wiki/Ligatura\\_\(typografie\)](https://cs.wikipedia.org/wiki/Ligatura_(typografie)).

- [10] Wikipedia: *Typografie* [online]. Dostupný z WWW: <https://cs.wikipedia.org/wiki/Typografie>.
- [11] *Word to Markdown Converter* [online]. Dostupný z WWW: <https://word-to-markdown.herokuapp.com>.

# Přílohy

# A. Dodatečné značky a příkazy pro Markdown a HTML

```
<!-- Komentář -->
```

---

```
<!-- Zalomení řádku (odsazení mezi řádky nebo ostavci) -->  
<br/>
```

---

```
<!-- Vložení bloku zdrojového kódu--
```

```
```sh  
int a, b, c;  
c = a + b;  
cout << c;  
```
```

---

```
<!-- Vložení zdrojového kódu na řádku-->  
Funkce 'main()' se nesmí přetěžovat!
```

---

```
<!-- Odkaz na email -->  
<jmeno@prijmeni.cz>
```

```
<!-- Vytváření podseznamů -->
```

1. Položka jedna
  2. Položka dvě
  3. Položka tři
    - \* Podpoložka jedna
    - \* Podpoložka dvě
  4. Položka čtyři
- 

```
<!-- Zaškrtávací seznamy (checkboxy) -->
```

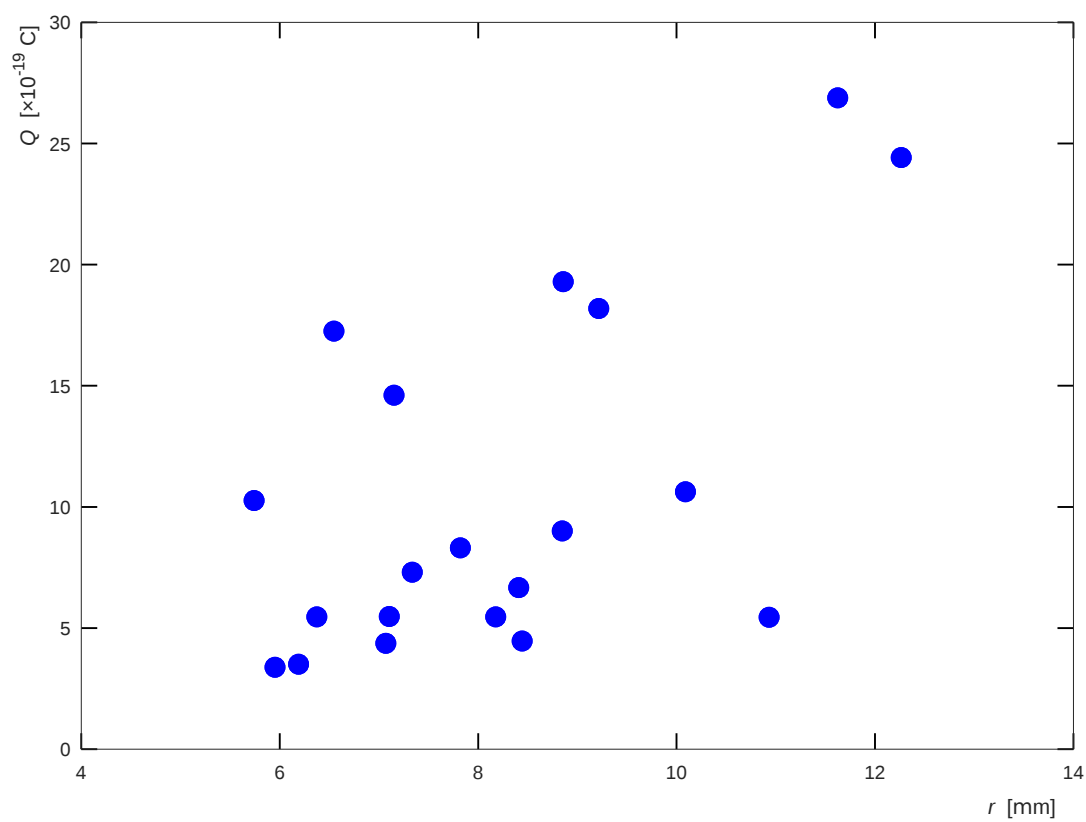
- První úkol
  - Druhý úkol
  - Tento úkol byl dokončen
- 

```
<!-- Vložení tabulky -->
```

| Sloupec1 | Sloupec2 | Sloupec3 |
|----------|----------|----------|
| Vlevo    | Na střed | Vpravo   |
| Vlevo    | Na střed | Vpravo   |
| Vlevo    | Na střed | Vpravo   |



## B. Graf 1: Millikanův experiment



Obrázek B.1.: Millikanův experiment - náboj  $Q$  olejové kapy je funkcí jejího poloměru  $r$

## Výpis kódu

```
# MILLIKANŮV EXPERIMENT
# GNU Octave
# Autor: Milan Somora

# Naměřená data
x=[10.933 8.84754 6.54554 8.17686 5.74197 7.15119 7.06926
    8.44321 11.6239 10.0887 6.18996 8.85786 7.82023 12.2649
    7.10329 5.95224 9.21638 8.40951 7.33632 6.37347];

y=[5.44987 9.01127 17.258 5.45986 10.2634 14.6089 4.37181
    4.46417 26.8878 10.6276 3.50671 19.2988 8.31275 24.4239
    5.47709 3.3815 18.1891 6.67073 7.30947 5.45834];

# Zobrazení naměřených dat
velikost_bodu = 600;
scatter(x,y,velikost_bodu,'b','filled')

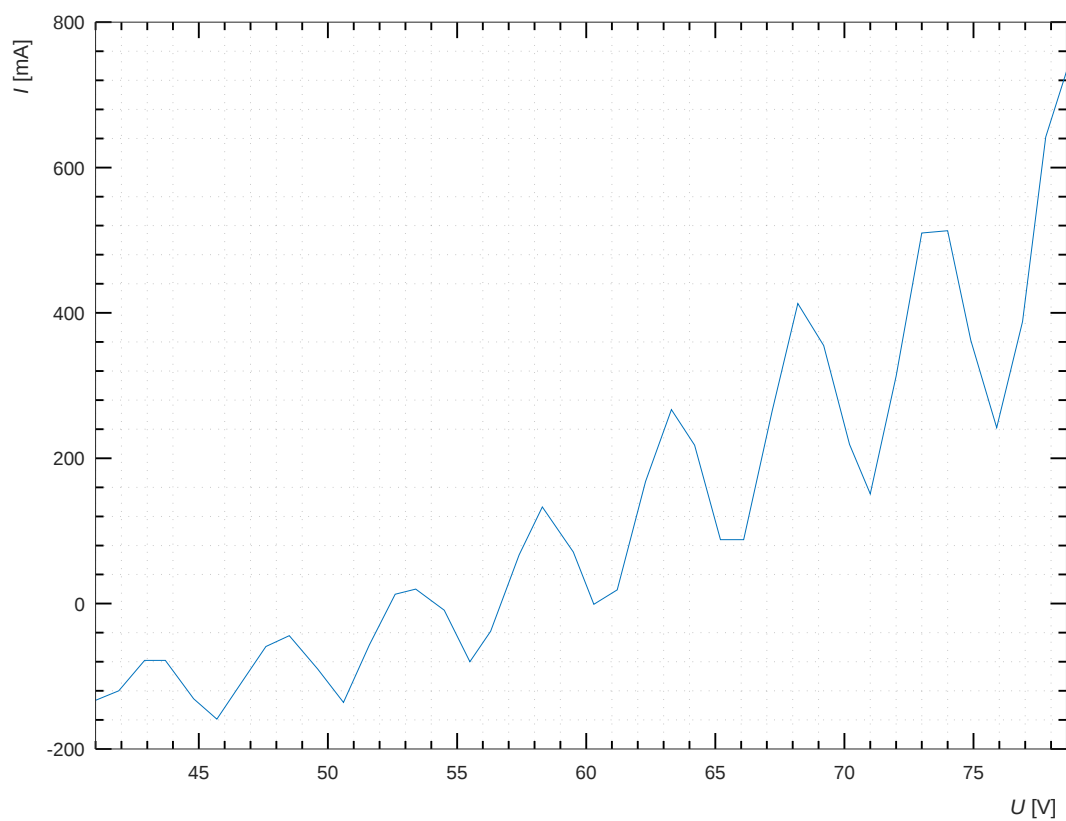
# Popisy os
xlabel('\it{r} [\mum]');
ylabel('\it{Q} [\times 10^{-19} C]');

# Posunutí popisu na konec osy
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [3.7 0 0])
ylabh = get(gca,'yLabel');
set(ylabh,'Position',get(ylabh,'Position') + [-0.1 12 0])

# Uzavření grafu do rámečku
box on

# Export grafu do souboru
print graf1.png - r600
```

## C. Graf 2: Franckova - Hertzova křivka



Obrázek C.1.: Maxima a minima na Franckově - Hertzově křivce

## Výpis kódu

```
# FRANCKOVA - HERTZOVA KŘIVKA
# GNU Octave
# Autor: Milan Somora

# Naměřená data
x=[41.0 41.9 42.9 43.7 44.8 45.7 46.7 47.6 48.5 49.6 50.6
   51.6 52.6 53.4 54.5 55.5 56.3 57.4 58.3 59.5 60.3 61.2
   62.3 63.3 64.2 65.2 66.1 67.2 68.2 69.2 70.2 71.0 72.0
   73.0 74.0 74.9 75.9 76.9 77.8 78.6];

y=[-133 -120 -78 -78 -131 -159 -106 -59 -44 -90 -136 -57 13
   20 -9 -80 -38 67 133 71 -1 19 168 267 218 88 88 265 413
   355 219 151 313 510 513 362 242 388 642 732];

# Zobrazení naměřených dat
plot(x,y)

# Omezení rozsahu osy x
xlim([41 78.6])

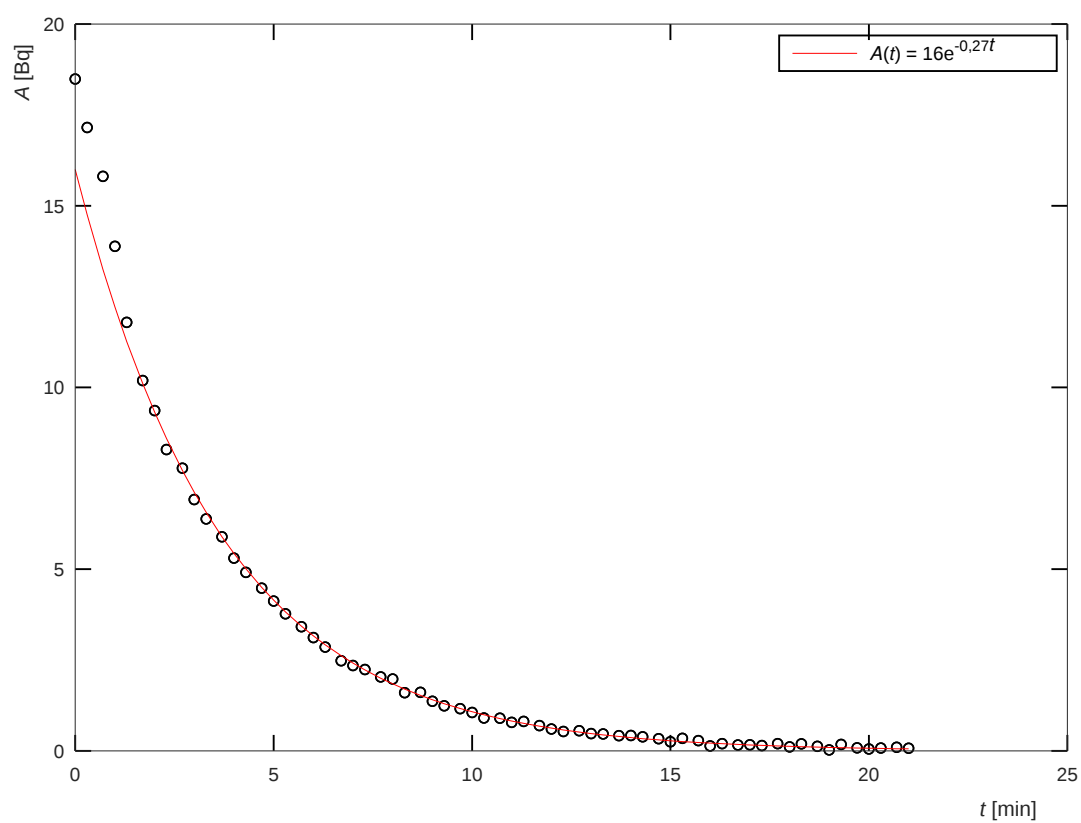
# Popisky os
xlabel('\itU [V]');
ylabel('\itI [mA]');

# Posunutí popisu na konec osy
xlabh = get(gca,'XLabel');
set(xlabh,'Position',get(xlabh,'Position') + [17 0 0])
ylabh = get(gca,'yLabel');
set(ylabh,'Position',get(ylabh,'Position') + [0 450 0])

# Zapnutí mřížky
grid minor;
```

```
# Export grafu do souboru  
print graf2.svg
```

## D. Graf 3: Aktivita Barya-137



Obrázek D.1.: Aktivita Barya-137 v závislosti na čase

## Výpis kódu

```
# AKTIVITA BARYA-137 V ZÁVISLOSTI NA ČASE
# GNU Octave
# Autor: Milan Somora

# Naměřená data
t=[0.0 0.3 0.7 1.0 1.3 1.7 2.0 2.3 2.7 3.0 3.3 3.7 4.0 4.3
  4.7 5.0 5.3 5.7 6.0 6.3 6.7 7.0 7.3 7.7 8.0 8.3 8.7 9.0
  9.3 9.7 10.0 10.3 10.7 11.0 11.3 11.7 12.0 12.3 12.7
  13.0 13.3 13.7 14.0 14.3 14.7 15.0 15.3 15.7 16.0 16.3
  16.7 17.0 17.3 17.7 18.0 18.3 18.7 19.0 19.3 19.7 20.0
  20.3 20.7 21.0];

A=[18.486 17.155 15.81 13.884 11.79 10.19 9.362 8.293 7.781
  6.918 6.384 5.89 5.306 4.91 4.477 4.123 3.773 3.416
  3.116 2.859 2.481 2.348 2.238 2.035 1.975 1.6 1.612
  1.366 1.241 1.159 1.057 0.903 0.898 0.783 0.809 0.694
  0.604 0.531 0.552 0.474 0.467 0.416 0.422 0.386 0.334
  0.249 0.343 0.284 0.137 0.199 0.164 0.165 0.148 0.201
  0.111 0.195 0.124 0.027 0.176 0.085 0.051 0.077 0.099
  0.075];

# Zobrazení naměřených dat
scatter(t,A,60,'k');
hold on

# Zoprazení proložené křivky
At=16*exp(-0.27*t);
plot(t,At,'r')
hold on

# Nastavení grafu
box on
xlabel('\itt [min]');
```

```
ylabel('{\itA} [Bq]');  
legend('Data', '{\itA}({\itt}) = 16e^{-0.27{\itt}}');  
  
# Vyexportování grafu do formátu SVG  
print graf3.svg
```



## E. Aproximace naměřených dat metodou nejmenších čtverců

```
# APROXIMACE NAMĚŘENÝCH DAT METODOU NEJMENŠÍCH ČTVERCŮ
# GNU Octave
# Výpočet křivky pro graf v příloze F (Aktivita Barya-137 v
# závislosti na čase)
# Autor: Milan Somora

# Rovnice aktivity je  $A(t)=A_0 \cdot e^{-l \cdot t}$ , hledáme proměnné A
# , B
# Po zlogaritmování získáme rovnici  $\log A = \log A_0 - l \cdot t$ 
# Substitute:  $\log A = y$ ,  $\log A_0 = C$ ,  $-l = B$ ,  $t = x$ 
# Získáváme lineární rovnici  $Y = C + Bx$ , pro kterou hledáme C,
# B
# Tj. hledáme čísla pro  $A = C \cdot e^{-B \cdot x}$ 

#-----
# Naměřená data
# Čas t
x=[0.0 0.3 0.7 1.0 1.3 1.7 2.0 2.3 2.7 3.0 3.3 3.7 4.0 4.3
  4.7 5.0 5.3 5.7 6.0 6.3 6.7 7.0 7.3 7.7 8.0 8.3 8.7 9.0
  9.3 9.7 10.0 10.3 10.7 11.0 11.3 11.7 12.0 12.3 12.7
  13.0 13.3 13.7 14.0 14.3 14.7 15.0 15.3 15.7 16.0 16.3
  16.7 17.0 17.3 17.7 18.0 18.3 18.7 19.0 19.3 19.7 20.0
  20.3 20.7 21.0];

# Aktivita A
```

```
A=[18.486 17.155 15.81 13.884 11.79 10.19 9.362 8.293 7.781
    6.918 6.384 5.89 5.306 4.91 4.477 4.123 3.773 3.416
    3.116 2.859 2.481 2.348 2.238 2.035 1.975 1.6 1.612
    1.366 1.241 1.159 1.057 0.903 0.898 0.783 0.809 0.694
    0.604 0.531 0.552 0.474 0.467 0.416 0.422 0.386 0.334
    0.249 0.343 0.284 0.137 0.199 0.164 0.165 0.148 0.201
    0.111 0.195 0.124 0.027 0.176 0.085 0.051 0.077 0.099
    0.075];
```

```
# počet naměřených hodnot
```

```
n=64;
```

```
# Suma vektoru x
```

```
x_suma=sum(x);
```

```
# y a suma
```

```
y=log(A);
```

```
y_suma=sum(y);
```

```
# x^2 a suma
```

```
x_kvadrat=x.^2;
```

```
x_kvadrat_suma=sum(x_kvadrat);
```

```
# Součin x*y a suma
```

```
xy=x.*y;
```

```
xy_suma=sum(xy);
```

```
#-----
```

```
# Získali jsme dvě rovnice o dvou neznámých:
```

```
# n*C+x_suma*B=y_suma
```

```
# x_suma*C+x_kvadrat_suma*B=xy_suma
```

```
# Vyřešíme soustavu lineárních rovnic pomocí matic
```

```
# Vektor L (levá strana), vektor P (pravá strana)
```

```
L=[n x_suma;x_suma x_kvadrat_suma];
```

```
P=[y_suma; xy_suma];
```

```

# Hodnost matice a hodnost matice rozšířené:
rank(L);
rank([L P]);
reseni_rovnice=L\P;

# Hodnoty proměnných C, B
g=reseni_rovnice(1,1);

# Získání prvního prvku v matici
C=exp(g); % protože  $\log A_0=C$ , tj.  $A_0=C$ 
B=-reseni_rovnice(2,1); % protože  $-l=B$ , tj.  $l=-B$ 

# Výpis hodnot
C
B

```