

# Návod na fitování funkce (Základy astronomie)

Martin Piecka

26. února 2023

---

## 1 Úvod

Během vašeho studia se budete často setkávat s tím, že budete mít data a z nich budete sestavovat grafy. V nich bývá někdy jasně patrné, že body v grafu tvoří rozdělení, které zodpovídá nějaké funkci. Vaším úkolem je pak tuto funkci do dat fitovat (často pomocí lineární regrese) a zjistit parametry fitu. Kromě hodnot parametrů bude samozřejmě důležité určit také nejistoty hodnot těchto parametrů.

Cílem dokumentu není detailní rozprava o matematickém postupu při fitování. S tím se setkáte na jiném předmětu astrofyziky nebo matematiky.

## 2 Fitování v jazyce Python

Programovací jazyk **Python** se poměrně často využívá v oblasti astrofyziky. Naleznete pro něj mnoho užitečných knihoven pro tvorbu grafů (**matplotlib**), práci s daty (např. **astropy**, **scipy**) či maticemi (např. **numpy**). Zde jsou uvedeny příklady funkcí, které mohou být pro vaše studium velice užitečné:

- **matplotlib.pyplot** – Velice užitečná knihovna sloužící pro vykreslování dat do grafu. Funkce **matplotlib.pyplot.plot()** slouží pro vykreslování rozdělení bodů nebo křivek. Pomocí dalších příkazů můžete nastavovat rozsah grafu, délky os, barvu a styl bodů (nebo křivek), a jiné. Funkce **matplotlib.pyplot.hist()** je pak užitečná pro tvoření histogramů (dostupná je i 2D verze, **hist2d()**, která je užitečná pro zobrazování tzv. *density maps*).
- **numpy.linalg.solve()** – Slouží pro řešení soustavy rovnic. S tímto problémem se můžete někdy setkat při řešení Maxwellových rovnic nebo při výpočtech týkajících se hvězdných atmosfér (i když zde se spíše setkáte s jazykem **Fortran**).
- **numpy.linalg.eig()** – Výpočet vlastních hodnot (a vektorů) pro vstupní matici. Ve fyzice se s vlastními hodnotami setkáte při studiu deformací nebo v kvantové fyzice (ale najdou se i jiné příklady).
- **scipy.optimize.curve\_fit()** – Tato funkce nás zde bude zajímat nejvíc. Jak může být patrné z názvu, používá se při fitování funkcí typu  $f(x)$  (pro více rozměrů,  $f(x_1, x_2, \dots)$ ), bychom se měli podívat po něčem jiném). Toto je obecně užitečné při analýze dat.
- **numpy.mean()** – Vypočítá střední hodnotu souboru hodnot. Dá se nahradit funkcí **numpy.average()**, kde je možné vložit také váhy (a počítat vážený průměr). Někdy může být vhodné použít **numpy.median()** pro výpočet mediánu hodnot (představuje lepší odhad středu asymetrických rozdělení).
- **numpy.std()** – Určí směrodatnou odchylku souboru hodnot. Ta se používá při odhadu nejistoty měření.

### 2.1 Instalace a používání

Detaily naleznete v souboru **navod\_python.pdf**.

### 2.2 Postup při fitování

Příložený k tomuto dokumentu je také vzorový příklad pro fitování (**fit\_code\_example.py**). Ten budeme používat pro popis postupu.

Prvním úkolem je načíst data, které budeme chtít analyzovat. Ve vzorovém souboru je to přímo definice dvou vektorů (řádky 05 a 06), ale v praktikách se setkáte s externími soubory dat. Tyto budete muset nejprve načíst. Postup je popsán v návodu pro **Python**. Data si nejprve vykreslete do grafu – zde by mělo být zřejmé, jakou funkcí budete chtít body proložit. V našem příkladu to bude přímka, tedy lineární funkce.

Dále si potřebujeme definovat odhadovanou funkci (řádky 08 až 10). Dbejte na to, aby první parametr funkce byla vstupní  $x$ -ová hodnota, pak následují jednotlivé parametry funkce (u lineární funkce  $y = a \cdot x + b$  to jsou parametry  $a$  a  $b$ ).

Samotné fitování už je jednoduchou záležitostí (řádek 12). Stačí nám zavolat funkci `curve_fit()`, vložit do ní odkaz na fitovanou funkci a seznamy hodnot  $x$  a  $y$ . Pokud je rozložení bodů nejasné, nebo je fitovací funkce komplikovanější (např. kombinace několika Gaussovek), bude nutno také upřesnit počáteční odhad parametrů (to nás však v praktikách nezajímá, chod programu by měl být bez tohoto problému). Výstupem funkce `curve_fit()` jsou dva seznamy hodnot. Prvním je vektor parametrů v pořadí, v jakém parametry vstupují do definované funkce. Druhým seznamem hodnot je kovariační matice – jestliže z ní chcete dostat nejistotu parametru s indexem  $i$ , budete muset vypsat odmocninu z hodnoty na diagonále, `sqrt(kovariacna[i][i])`. Vypis parametrů a jejich nejistot je proveden na řádcích 13 až 16.

Pro daný rozsah hodnot  $x$  je vykreslení nejlepšího fitu triviální. Stačí pro daný soubor hodnot  $x$  vypočítat funkční hodnoty s pomocí určených parametrů a uložit je do nového souboru hodnot (řádky 18 až 20). Rozložení hodnot  $x$ ,  $y$  si pak můžeme vykreslit společně s fitem  $x$ ,  $yy$ .

### 3 Fitování v aplikaci QtiPlot

Jestliže máte problém s programováním v Python-u, můžete využít existující aplikaci pro analýzu dat. QtiPlot je volně dostupná a poměrně intuitivní aplikace, která poslouží pro naše účely.

#### 3.1 Instalace a používání

Instalace by měla být snadnou záležitostí. Pomocí odkazu na stránce <https://it.muni.cz/sluzby/software/qtiplot> si může každý student univerzity stáhnout licencovanou verzi aplikace.

Při prvním použití si nejprve vyzkoušejme něco snadného. Do sloupce 1 [X] napište hodnoty (0, 1, 2, 3, 4, 5) a do sloupce 2 [Y] pak hodnoty (2, 1, 4, 6, 6, 8). Jestliže si chceme data zobrazit jako body, označíme oba sloupce (klikněte na název prvního sloupce, pak CTRL + klik na název druhého sloupce) a vybereme z menu na liště **Plot** → **Symbol** → **Scatter**. Mělo by se v aplikaci objevit nové okno s grafem. Ten si můžeme trochu pozměnit:

- Klikněte na popisek (“1”) na horní straně grafu a smažte ho, učiňte to stejné s dílký na horní straně. Podobně postupujte s pravou stranou grafu. Tyto části grafu teď nebudeme potřebovat.
- Dvojklikem na název grafu (“Title”) můžeme text přepsat – zvolte například název “První graf”.
- Legendu zde nepotřebujeme – klikněte na ni (černý bod, číslice 2 na pravé straně, ohraničeno obdélníkem) a smažte ji.
- Nastavme teď popisky obou os. Přepište je například na “X [m]” a “Y [m]”.
- Vzhled bodu můžete měnit v menu, které se zobrazí po dvojkliknutí na body v grafu. Změňte velikost bodů na 6, barvu ohraničení na modrou, šířku ohraničení na 2,0 a barvu vnitřku bodů na žlutou. Pro potvrzení nastavení klikněte na tlačítko **Apply** (případně to aplikace udělala automaticky). Jistě vidíte, že zde existují i další nastavení (symbol bodů, průhlednost, atd.).
- Klikněte dvakrát na pozadí grafu. Zobrazí se okno, kde si můžete nastavit mnohé vlastnosti okna (klikněte na **Graph1**), velikost a pozadí grafu (**Layer1**) nebo vnitřní vlastnosti grafu (**Table1: ...**), jako je například vzhled bodů.
- Jistě vám přijde, že vnitřní mřížka se pro náš typ grafu nehodí. Dvakrát klikněte na čísla při popisku os (je jedno které). V okně je několik možností – vyberte si na liště položku **Grid**. Zde můžete editovat horizontální a vertikální část mřížky zvlášť. Jestliže chcete mřížku méně hustou, nechte zaškrtnuté pouze políčko **Major Grids**. Jestliže se chcete mřížky zbavit, nechte obě políčka nezaškrtnutá.
- Ve stejném okně můžete editovat rozsah grafu pod položkou **Scale**. Nastavte  $x$ -ovou osu (**Bottom**) na rozsah (-2,5 , 8,5). Jestli chcete upravit  $y$ -ovou osu, v seznamu popisků nalevo vyberte **Left**.
- Ve stejném okně můžete taky nastavit počet dílků v položce **General** nebo vlastní názvy jednotlivých dílků v položce **Special Ticks**.

### 3.2 Postup při fitování

Nejprve si musíte načíst data – použijte na to soubor `fit_test.txt`. Z menu si na liště si zvolte **File** → **Import** → **Import ASCII** (nebo můžete použít klávesovou zkratku CTRL+K). Soubor si vyberete kliknutím na tlačítko při možnosti **Name**. Data by teď měla být načtena, ale ještě budete muset změnit formátování dat.

- **Separator** – Zde ponechte možnost **TAB**. Je možné také použít **SPACE**.
- **Ignore first** – můžete ignorovat řádky v souboru. V našem příkladu ponechte **0 lines**.
- **Decimal Separator** – Jestliže to již není zvolené, vyberte **1,000.0**.
- **Use first row as** – ponechte zaškrtnuté, vyberte **Column Names**. První řádek souboru je tedy použit pro výběr názvů sloupců v tabulce.
- **Use second row as** – nezaškrtnuté.
- **Remove white spaces from line ends** – zaškrtnuté. Ignoruje nadbytečné mezery na konci řádku.
- **Simplify white spaces** – zaškrtnuté. Nahradí vícero mezer za sebou jednou mezerou.
- **Omit thousands separator** – zaškrtnuté.
- **Import as read-only** – nezaškrtnuté.

Náhled tabulky na spodku okna by měl být teď v pořádku. Stisknutím tlačítka **Ok** potvrdíte načtení ze souboru.

Nejdříve musíte vykreslit data do grafu – použijte možnost **Scatter**. Pokud na nic nekliknete, měli byste mít okno grafu označené (jestli nemáte, klikněte na okno, ale na vnitřní částí grafu). Teď zvolte **Analyze** → **Fitting** → **Fit Wizard** (nebo CTRL+Y) – mělo by se vám otevřít okno pro nastavení fitování. Postupujte následovně:

1. Pod **Category** si vyberte možnost **User defined**. Funkci si budete psát sami v políčku pod **Parameters**. Na první pohled je zřejmé, že rozdělení bodů v grafu je podobné kvadratické funkci. Do políčka proto napište  $a*x^2 + b*x + c$  – parametry fitu budou hodnoty  $a$ ,  $b$  a  $c$ . Obecně si zde můžete nadefinovat jakoukoliv funkci.
2. Klikněte na šipku na pravé spodní straně okna (**Start fitting session**). V novém okně si můžete nastavit počáteční odhad parametrů fitu, což je užitečné, pokud řešení nekonverguje. V našem případě (a v praktikách) toto nemusíte řešit a můžete rovnou přejít k fitování. Na pravé spodní straně okna klikněte na tlačítko se zelenou ikonou a textem **Fit**.
3. Výsledky fitu můžete vidět již v okně, ale budeme se hlavně zajímat o výsledky zobrazené v **Results Log** pod hlavním menu okna aplikace **QtiPlot**. Uvedené jsou zde zvolené názvy parametrů (**Parameter**), hodnoty (**Value**) a nejistoty (**Error**). Užitečná je někdy hodnota  $R$  (korelační koeficient) nebo  $R^2$ , která představuje míru kvality fitu. V absolutní hodnotě nabývá hodnoty v rozsahu (0,1). Pokud by byla tato hodnota příliš malá (v absolutní hodnotě blízko nuly), může být potřebné zvolit si jinou funkci.

Fit funkce by měl být zobrazen v grafu automaticky po kliknutí na tlačítko **Fit**.