

## 1. 必要なもの

python version 3

SciPy version 0.17.0 以上

Matplotlib バージョンへの制限は不明

NumPy バージョンへの制限は不明

## 2. 使用方法

```
usage: fit.py [-h] [-without_plot] [-without_fit] [-sample_range SAMPLE_RANGE]
              [-method {lm,trf,dogbox}]
              fnames area dbfname outfname
```

positional arguments:

fnames	Input file names
area	Area
dbfname	database file name for initial parameters
outfname	output file name

optional arguments:

-h, --help	show this help message and exit
-without_plot	Without result plot
-without_fit	Without fit
-sample_range SAMPLE_RANGE	Sample wavelength ranges
-method {lm,trf,dogbox}	Algorithm to perform minimization.

### **fname**

データキューブ FITS ファイル

### **Area**

フィッティングに用いる領域

コンマで区切って次のように記載する。 X1,X2,Y1,Y2

### **dbfname**

入力のデータベースファイル。

1 行目 ファイルの作成日時

- 2行目 データキューブ FITS ファイル名
- 3行目 フィットに用いた領域
- 4行目 フィットに用いた波長範囲
- 5行目 入力データファイル名
- 6行目 フィットに用いたアルゴリズム名
- 7行目 フィットに用いた **input database file** 名とフィット終了時のメッセージ
- 8行目以降 フィットに用いる関数のパラメーター情報

最初の行は関数名と、その関数が利用するパラメーター数が記載されている。

次からパラメーターの情報で、最初の数値からそれぞれ以下の情報を表している。

初期値、下限値、上限値、ステップ、収束条件、フラグ

上下限値に制限を設けない時には十分広い範囲を指定する。

ステップはフィッティングの際に最初に用いられるステップ。フラグでリンクを選択した場合には、リンク先に対するスケーリングファクター。

フラグは固定値にしたければ-1、最適化したければ0、他のパラメーターとリンクさせたければ、その成分番号。

### **outfname**

フィット結果が記載されたデータベースファイル

**dbfname** と同じフォーマットにしているので、出力結果を入力データベースとして使える。

### **-without\_plot**

フィット後に結果をプロット表示しない。

### **-without\_fit**

フィットせずに初期パラメーターを用いて表示だけ行う。初期パラメーターのチェック用。これを指定した場合、アウトプットファイルは本来必要ないが、指定しておかないとエラーとなる。

### **-sample\_range SAMPLE\_RANGE**

フィッティングに用いる波長範囲を指定する。波長範囲 (**SAMPLE\_RANGE**) の記述方法は 6500-6600,7000-7300 のように複数の範囲をコンマでつなげて指定することもできる。  
\* (アスタリスク) を指定すると、全ての範囲を用いる (特に指定しなくてもデフォルト設定になっている)。

### **-method {lm,trf,dogbox}**

フィットのアルゴリズム。デフォルトは `trf`。

### **3. 関数とパラメーター**

#### **3.1. gaussian**

パラメーター数：3

パラメーター 1：ピーク強度

パラメーター 2：中心波長（単位はデータファイルで使われている単位）

パラメーター 3：輝線幅 (km/s)

#### **3.2. lorentzian**

パラメーター数：3

パラメーター 1：ピーク強度

パラメーター 2：中心波長（単位はデータファイルで使われている単位）

パラメーター 3：輝線幅 (km/s)

#### **3.3. linear**

パラメーター数：2

パラメーター 1：傾き

パラメーター 2：波長 0 での強度

#### **3.4. 関数の追加**

`funcs.py` に関数情報を追加すればよい。

関数自体の定義と、パラメーターのスケーリング方法を定義する。