

# 宇宙科学実習 I : python で遊ぼう (2) : 周期解析

今回は Ubuntu 環境で作業をしてください。

## 1 周期変動する現象の解析 — パワースペクトル

特定の周期をもって時間的（あるいは空間的）に変動する自然現象はたくさんあります。このような現象，系の性質を調べるときに役立つ指標としてパワースペクトル（密度）があります。例えば交流電気回路の端子電圧  $V$  が時間に依存するとき，これをフーリエ展開して

$$\tilde{V}(f) = \int_{-\infty}^{\infty} V(t)e^{2i\pi ft} dt \iff V(t) = \int_{-\infty}^{\infty} \tilde{V}(f)e^{-2i\pi ft} df \quad (1)$$

フーリエ係数  $\tilde{V}(f)$  を得たとします。ここで， $f$  は振動数です。フーリエ係数は複素数です。<sup>1</sup> このとき，パワー  $P(f)$  は

$$P(f) = \tilde{V}^* \cdot \tilde{V} \quad (2)$$

と定義できます。この値は振動数  $f$  で振動する電場成分のエネルギーに比例します。

天体現象における変動する観測量（明るさ，視線速度，etc.）についても，パワーを同様に定義して使えます。<sup>2</sup>

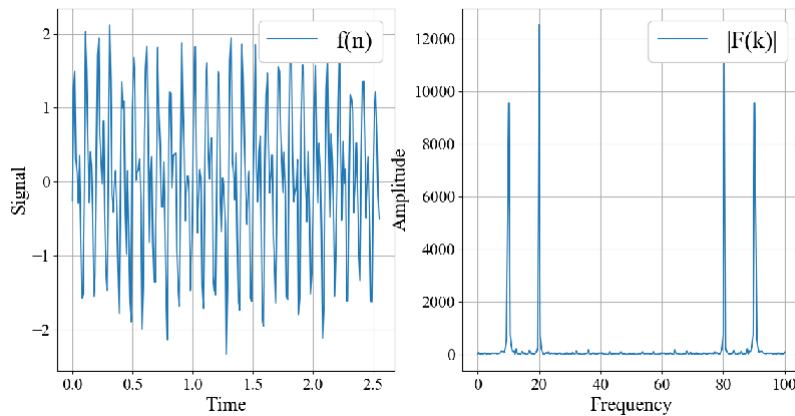


図 1: 変動する信号（左）とそのフーリエ変換（右）。左の信号は振動数の異なる 4 つの成分の和からなっていることが分かる。

<sup>1</sup>大雑把に言うと，この式は， $V(t)$  を異なる振動数をもつ無限個の波形関数（基底関数）で展開するという意味をもちます。そのときの”展開係数”が  $\tilde{V}(f)$  です。

<sup>2</sup>一般の場合にパワーの次元は「エネルギー/時間」と異なってきますが。

## 2 Lomb-Scargle 法

実験、観測で得られたデータは無限個の連続した数からなっているわけではなく、離散化した数値のあつまりになっています。このような離散データにおける変動のパワーを解析する方法はいろいろあります。よく使われるものに数値的なフーリエ変換である、高速フーリエ変換 (Fast Fourier Transform, FFT)、離散フーリエ変換 (Discrete Fourier Transform, DFT) があります。特に FFT は大規模なデータの処理を高速に行えるため、よく使われる強力な方法ですが、データのサンプリング間隔 (データをとる) が一定で (例えば、1 秒毎にとったデータ)、データ数が 2 のべき乗個の場合にしか使えないという欠点があります。

天文現象ではデータが同じ時間間隔で取れることは希であるので、不等間隔の離散データのパワーを調べる方法が必要になります。ここではそのような方法の一つである Lomb-Scargle 法を使ってみましょう。

### 2.1 Lomb-Scargle ピリオドグラム

$N$  個のデータ点を  $(t_j, y_j)$  とします ( $j = 1 \sim N$ )。このとき、振動数  $f$  の関数である次の関数 (ピリオドグラム) を導入します [1, 2, 3]

$$P(f) \equiv \frac{1}{2\sigma^2} \left[ \frac{\left( \sum_{j=1}^N (y_j - \bar{y}) \cos \omega(t_j - \tau) \right)^2}{\sum_{j=1}^N \cos^2 \omega(t_j - \tau)} + \frac{\left( \sum_{j=1}^N (y_j - \bar{y}) \sin \omega(t_j - \tau) \right)^2}{\sum_{j=1}^N \sin^2 \omega(t_j - \tau)} \right] \quad (3)$$

ただし、 $\omega = 2\pi f$  [角振動数] で、 $\bar{y} = \frac{1}{N} \sum_{j=1}^N y_j$  [データの平均]、 $\sigma = \frac{1}{N-1} \sum_{j=1}^N (y_j - \bar{y})^2$  [データの分散] です。また、位相因子  $\tau$  は

$$\tan 2\omega\tau = \frac{\sum_{j=1}^N \sin 2\omega t_j}{\sum_{j=1}^N \cos 2\omega t_j} \quad (4)$$

で定義されます。このピリオドグラムの値が大きな値をとる振動数の成分が、データ点の時間変動に大きな寄与をしています

このピリオドグラムは、与えられたデータ点を、振動数の与えられた三角関数  $\eta(t) = A \cos \omega t + B \sin \omega t$  で最小二乗フィットしたときのパワーを与えます。

### 2.2 astropy で Lomb-Scargle 法を使ってみよう

astropy では Lomb-Scargle 法のライブラリが使えます。

#### 例題

テストデータ `testLS.dat` は、ランダムなノイズの上に周期関数を加えたデータファイルです。このデータファイルの Lomb-Scargle ピリオドグラムを描き、周期関数を見つけましょう。

1. まず、データの横軸、縦軸をそれぞれ 1, 2 列目としてグラフを描いてみなさい (gnuplot を使っても良い)。
2. 次に、ダウンロードした python コード (`LS.py`) を `spyder` で実行し、ピリオドグラムを描いてみなさい。
3. 図の x 軸の範囲を狭めて、鋭いピークの付近だけをプロットしなさい。

例題の周期関数は振動数が 5.6 と 7.3 の正弦関数が 2 : 3 の割合で混じったものです。

この例題の python コードの 1~23 行目と, 30~34 行目までは, 以前の課題で pyplot を使って CSV ファイルからグラフを描いた際のものと同じです。

26~27 行目が今回の Lomb-Scargle 法を適用している部分です。26 行目では astropy の統計処理モジュール stats から, LombScargle 関数を呼び出しています。実際にピリオドグラムを計算をしているのは 27 行目で, frequency, power はそれぞれ振動数 (横軸), パワー (縦軸) の値が代入される配列です。

#### 課題 1 : 太陽活動周期

第一, 四回目の実習で使った黒点数のデータファイル SN\_m\_tot\_V2.0.csv をもう一度ダウンロードします。平均黒点数の年変化のデータから, 黒点数のピリオドグラムを作りなさい。ピリオドグラムで最も高いピークをもつ振動数に対応する周期は何年ですか?

### 3 変光星の周期解析

脈動変光星とは, 星内部の構造に起因したメカニズムによってその固有振動が励起され, 特徴的な周期で明るさ (等級) や表面のガスの視線速度を変化させる星です。ここではそのような星の光度変化の周期を調べてみましょう。

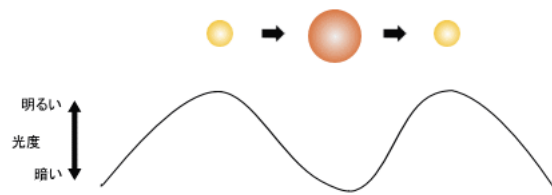


図 2: 脈動変光星の概念図。収縮時に表面温度が上がって明るくなり, 膨張時には表面温度が下がって暗くなる, という過程を周期的に繰り返す。

#### 課題 2 : 炭素星 LINEAR7744346

ファイル LINEAR7744346.dat には, 炭素星 LINEAR7744346 の変光データが格納されています。第一列は観測のユリウス日 (単位「日」), 第二列は観測装置の情報 (無視できます), 第三列は見かけの等級, 第四列はその誤差 (不定性), 第五, 六列はフラックスとその誤差です。ここでは等級とその誤差を用いて変光の周期を求めましょう。

1. 横軸を観測日, 縦軸を等級にしてプロットしてみなさい。
2. Lomb-Scargle 法を適用して変光周期 (日) を推定しなさい。
3. 1 周期分の位相を  $\varphi: [0, 1]$  として, 位相を横軸に, 等級を縦軸にしてプロットしなさい。データ点にはエラーバーをつけること。

「課題 2 の 3.」のヒント

まず, LombScargle 関数で作成した frequency と power の配列を使って最もパワーの大きい振動数を見つけ出します。これには numpy の argmax を使うと便利でしょう。配列 power に対

して、`numpy.argmax(power)` はその配列の要素の最大値を与える配列番号を返します。この番号に対応する配列 `frequency` の要素が振動数のベストな推測値で、その逆数が周期の推測値です。

```
import numpy as np
best_frequency = frequency(np.argmax(power))
best_Period = 1/best_frequency
print('period (days): ', best_Period)
```

次に、時刻を変動の位相に直します。上で求めた周期でデータの時刻 `x_list` を割った値の「1を法とする剰余」を考えます。<sup>3</sup>python では浮動小数点の剰余も演算子「%」で出来ます。観測時刻の配列 `x_list` から位相の配列 `phase` を作りましょう：

```
phase = (x_list/best_Period) % 1
```

あとは、横軸に `phase`、縦軸に等級 `y_list` をとってプロットします。データ点を結ばずにエラーバーをつけてプロットするには少しコツがあります。誤差の配列を `dy_list` として作ったとすると、

```
plt.errorbar(phase, y_list, yerr=dy_list, marker='o', capthick=1, \
             capsize=10, lw=1, linestyle='None')
```

などとします。最後に、等級は上に行くほど小さくしたいので

```
plt.gca().invert_yaxis()
```

とおまじないをしておきます。

## 参考文献

- [1] Lomb, N.R., *Astrophysics and Space Science*, **39**, 447 (1976)
- [2] Scargle, J.D., *The Astrophysical Journal*, **263**, 835 (1982)
- [3] VanderPlas, J., *The Astrophysical Journal Supplement*, **236**:16 (2018)

---

<sup>3</sup>時刻を周期で割った値の整数部が周期の何倍の時刻が経過したかを表し、少数部が位相を与えます。