

# ティティウス・ボーデの法則の係数

The coefficients of Titius-Bode's Law

作花 一志 (京都情報大学院大学)

Kazuyuki Sakka (The Kyoto College of Graduate Studies for Informatics)

## Abstract

The coefficients of Titius-Bode's law are derived by using Excel and R programming of non-linear regression method

### 1. はじめに

ティティウス・ボーデの法則とは惑星の太陽からの距離が簡単な数列で表されるというもので、18世紀から知られている。1766年、ヨハン・ダニエル・ティティウス (1729-1796) が提案したときには、単なるパズルであるとして批判されたが、1772年にヨハン・ボーデ (1746-1826) が、物理的説明を加えた形で数列の重要性を書き加えて有名になった。

当時知られていた太陽系の6つの惑星 (水星, 金星, 地球, 火星, 木星, 土星) の軌道長半径 (太陽からの平均距離)  $a$ は

$$a_n = 0.4 + 0.3 \cdot 2^n$$

で表せ、単位は天文単位 (AU) である。ここで水星は  $n = -\infty$ 、金星は  $n = 0$ 、地球は  $n = 1$ 、火星は  $n = 2$ 、木星は  $n = 4$ 、土星は  $n = 5$  であり  $n = 3$  は欠番である。この法則の提唱後1781年に天王星が発見され、その距離は  $n = 6$  の場合によく合っていたため、法則の信憑性が高まった。そのため、空席である  $n = 3$  に該当する天体の探索が行われ、その結果1801年に小惑星ケレスが発見された。その後、この  $n = 3$  に当たる距離には多数の小惑星が存在していることが判明したが、惑星に相当する天体が発見するには至らなかった。ところが1846年に発見された海王星の平均距離は30.110AUで  $a_7 = 38.8$  から大きくずれており、法則の信憑性について疑問の声があがった。むしろ冥王星の方が平均距離39.445AUで  $a_7$  に近い。2006年冥王星は惑星から外れたが、その軌道付近には多数のカイパーベルト天体が見つっている。

近年の惑星形成のシミュレーションの結果では、形成後も惑星の軌道半径の変化が起きることが分かっている。また近年、太陽系外惑星系が多数発見されたが、そこでは大きな惑星が親星のすぐ近くを公転するケースが多い。そのため、ティティウス・ボーデの法則は、力学的な必然ではないとも言われている。

この小文では実測データからその係数を非線形回帰分析で求めてみる。

### 2. 非線形回帰分析

実測データとしては水星からケレスも含め天王星までの平均距離  $a$  を使う。第2列の  $n$  は前節の値に従う。第4列は第1階差であるが何の規則性も見当たらない。第2階差をとっても同じである。第5列は各  $a_n$  より水星の値を引いたものでこれを新たな数列  $b_n$  とし、これを非線形関数で近似する。

表1 惑星の平均距離

	n	a	階差	a-a(水)
水星	$-\infty$	0.3871		
金星	0	0.7233	0.3362	0.3362
地球	1	1	0.2767	0.6129
火星	2	1.5237	0.5237	1.1366
ケレス	3	2.767	1.2433	2.3799
木星	4	5.2034	2.4364	4.8163
土星	5	9.5371	4.3337	9.15
天王星	6	19.1913	9.6542	18.8042

#### 2.1 Excelによる解

最も簡単な方法は散布図 ( $b_n$  vs  $n$ ) に近似式を書き加えるというものである。散布図 (図1) からしてこれを近似する数式としては曲率が正の増加関数であろう。Excelでは近似曲線の式として

指数 線形 対数 多項式 累乗 移動 から選ぶようになっているが、適するものとしては指数関数と偶数次多項式である。指数関数を選ぶと近似関数として  $y = 0.3163e^{0.6758x}$  となるが、底を変換すると  $y = 0.3163 \cdot 1.966^x$  となる。

2次関数では

$$y = 0.8134x^2 - 2.1603x + 1.2266 \quad \text{となるがズレが目立つ。}$$

4次関数では

$$y = 0.0353x^4 - 0.2497x^3 + 0.8179x^2 - 0.4715x + 0.3661$$

となり、よくフィットしている。

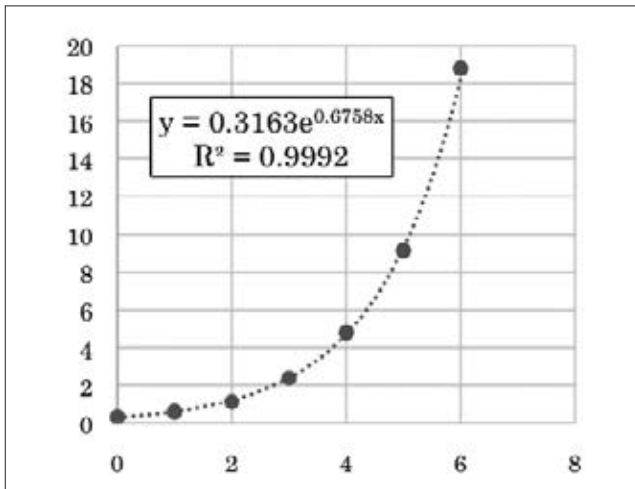


図1 Excelによる指数関数近似

## 2.2 Rによる解

次ページに記すRプログラミングではデータフレームという形式でデータを扱う。指数関数 $y=a*bx$  で近似する場合のコードは

```
u1<-nls(y~a*b^x, start=c(a=0.2, b=0.2),
trace=TRUE)
```

start=以下は $a=0.2$ ,  $b=0.2$ を初期値として繰り返し演算を行うという意味である。6回の繰り返し演算で収束し

```
a=0.2885 b=2.0051
```

と求まった。

近似関数として2次関数 $y=a*x^2+bx+c$ を用いた場合のコードは

```
u2<-nls(y~a*x^2+b*x+c, start=c(a=0.2, b=0.2,
c=0.2),
trace=TRUE)
```

であり次のように求まる。

```
a= 0.8134 b=-2.1603 c=1.2266
```

同様にして近似関数として4次関数 $y=a*x^4+bx^3+cx^2+dx+e$ を用いた場合には

```
a=0.03527724 b=-0.24968234 c=0.81789407
d=-0.47146919 e=0.36611291
```

が求まった。

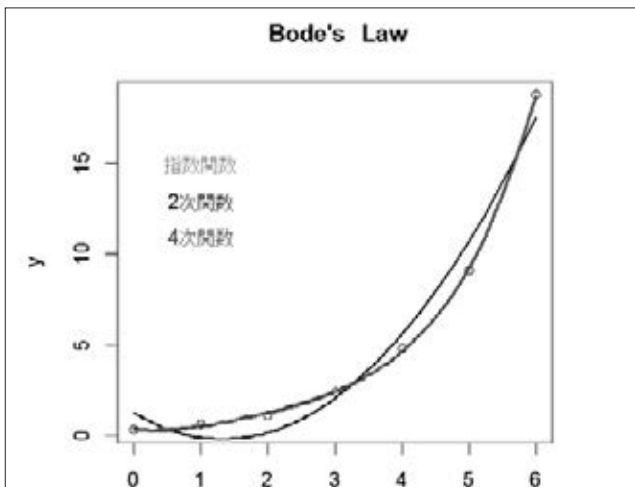


図2 Rによる3つの近似曲線

この3曲線を描いたのが図2である。指数関数と4次曲線はどちらもよくフィットしている。

ExcelでもRでも同様な近似指数関数が得られ結局ボーデの法則は

$$a_n=0.3871+0.3163*1.966^n \quad \text{Excel}$$

$$a_n=0.3871+0.2885*2.005^n \quad \text{R}$$

と書き表される。残差平方和はExcelでは0.327919, Rでは0.073607403でRの方が優れているといえよう。

また指数関数と4次関数はどちらもよくフィットするがそれはこの区間だけのことである。 $x<0$ ,  $x>6$ では両者の値は大きく異なる。

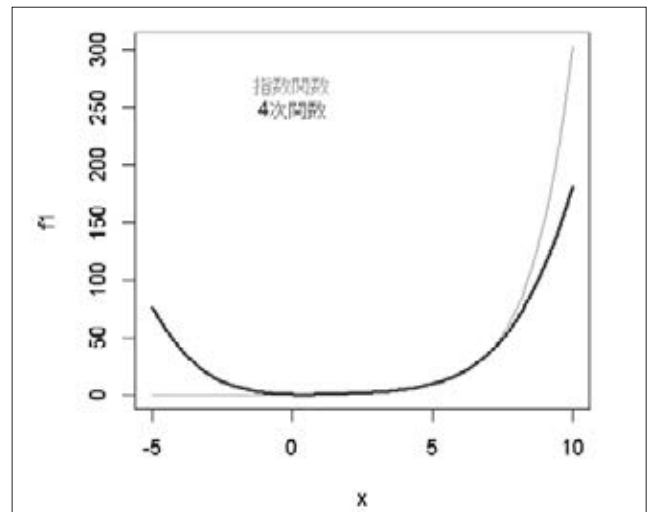


図3 指数関数(細線)と4次関数(太線)

この7点にフィットする曲線は多数あるということである。

```

##### Bode ###
x<-c(0:6)
y<-c(0.3362, 0.6129, 1.1366, 2.3799, 4.8163, 9.15, 18.8042)
xy<- data.frame(x, y)      # x, y データフレーム
u1 <-nls(y~a*b^x, start=c(a=0.2, b=0.2), trace=TRUE)
u2 <-nls(y~a*x^2+b*x+c, start=c(a=0.2, b=0.2, c=0.2), trace=TRUE)
u4 <-nls(y~a*x^4+b*x^3+c*x^2+d*x+e, start=c(a=0.2, b=0.2, c=0.2, d=0.2, e=0.2), trace=TRUE)

plot(x, y) # 散布図
coef(u1)  # 指数回帰係数
coef(u2)  # 2次回帰係数
coef(u4)  # 4次回帰係数
#回帰曲線関数定義&プロット
f1 <- function(x) coef(u1)[1]*coef(u1)[2]^x
plot(f1, 0, 6, col=3, add=T)
f2 <- function(x) coef(u2)[1]*x^2+coef(u2)[2]*x+coef(u2)[3]
plot(f2, 0, 6, col=4, add=T)
f4 <- function(x) coef(u4)[1]*x^4+coef(u4)[2]*x^3+coef(u4)[3]*x^2+coef(u4)[4]*x+
coef(u4)[5]
plot(f4, 0, 6, col=2, add=T)
text(1, 15, "指数関数", col=3);
text(1, 13, "2次関数", col=4);
text(1, 11, "4次関数", col=2);
title(" Bode's Law")

dev.new()
plot(f1, -5, 10, col=3)
plot(f4, -5, 10, col=2, add=T)
text(0, 270, "指数関数", col=3);
text(0, 250, "4次関数", col=2);

```

#### 【参考文献】

金明哲「Rによるデータサイエンス」 森北出版 2007