非線形データの近似曲線作成について
―Excel、R、Python―

Excelグラフのべき乗近似、指数近似の問題

(1)井口豊氏（生物科学研究所）によると
Excelのグラフや関数を用いて，累乗近似や指数近似を行なう場合、「データを対数変換して直線近似した」と書かないと正しいとは言えない　とのこと
単純に、データ点に近い曲線を求めたいならば、非線形回帰を行なう必要がある

(2) 多項式回帰は、データを入力すると係数a，b，c・・・は線形回帰で求めることができるので、上記の問題が発生しない

(3)井口豊氏のデータを用いて、Excel、R，Pythonにより近似曲線を作成してみた

結果の概要（べき乗近似）

(1)　Excel：面倒なソルバーを使用する必要がある

(2)　R：短いコードで求めることができる（ライブラリの追加が不要）

(3)　Python：モジュールを追加することにより求めることができる

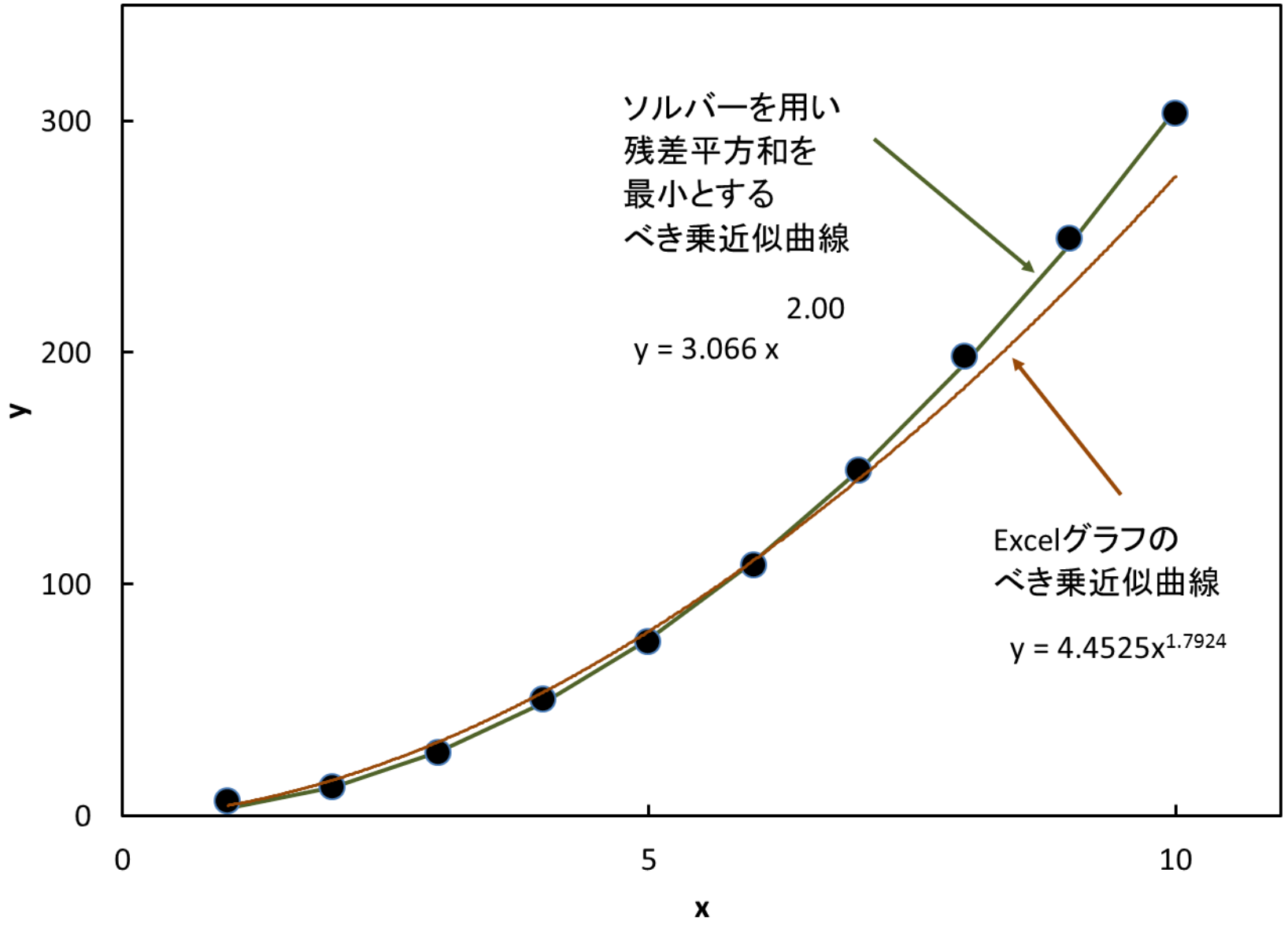
はじめに（Excelグラフの近似について）

1. Excelグラフの近似方法
   Excelのグラフや関数を用いて、累乗近似や指数近似を行なう場合、それは、データを対数変換した上で、直線回帰として最小二乗法を適用している。したがって、その結果を論文やレポートに記す場合は、「データを対数変換して直線近似した」と書かないと正しいとは言えない
   単純に、データ点に近い曲線を求めたいならば、非線形回帰を行なう必要がある[1]。

2. 問題のないケース
   x, yに数値を入力すると、係数a, b,・・・が線形で求まるもの
   y = a・x + b、y = a・x^2 + b・x + c、 y = a・x^3 + b・x^2 + c・x + d、・・・

3. 近似曲線がデータにフィットしないケース
   x, yに数値を入力すると、係数a, b,・・・が非線形のもの
   y = a・x^b：べき乗近似
   y = a・exp(bx)：指数近似
   Excelでデータをフィットさせるにはソルバーを用いて、係数を求める必要がある。次ページにグラフの近似とソルバーを用いた非線形近似を示す

Excelグラフのべき乗近似（Office2010）



Excelグラフのべき乗近似は、べき乗データにフィットしていない。ソルバーでの計算が必要。

Excelワークシート／ソルバー計算例

	A	B	C	D
1	a	3.065896		
2	b	1.997792		
3				
4	x	y	予測値	残差平方
5	1	6	3.06589583	8.608967
6	2	12	12.24483175	0.059943
7	3	27	27.52622118	0.276909
8	4	50	48.90443541	1.200262
9	5	75	76.375547	1.89213
10	6	108	109.9365294	3.750146
11	7	149	149.584918	0.342129
12	8	198	195.3186333	7.189727
13	9	249	247.1358762	3.474957
14	10	303	305.0350614	4.141475
15			残差平方和	30.93664
16				

（予測値ーy）^2

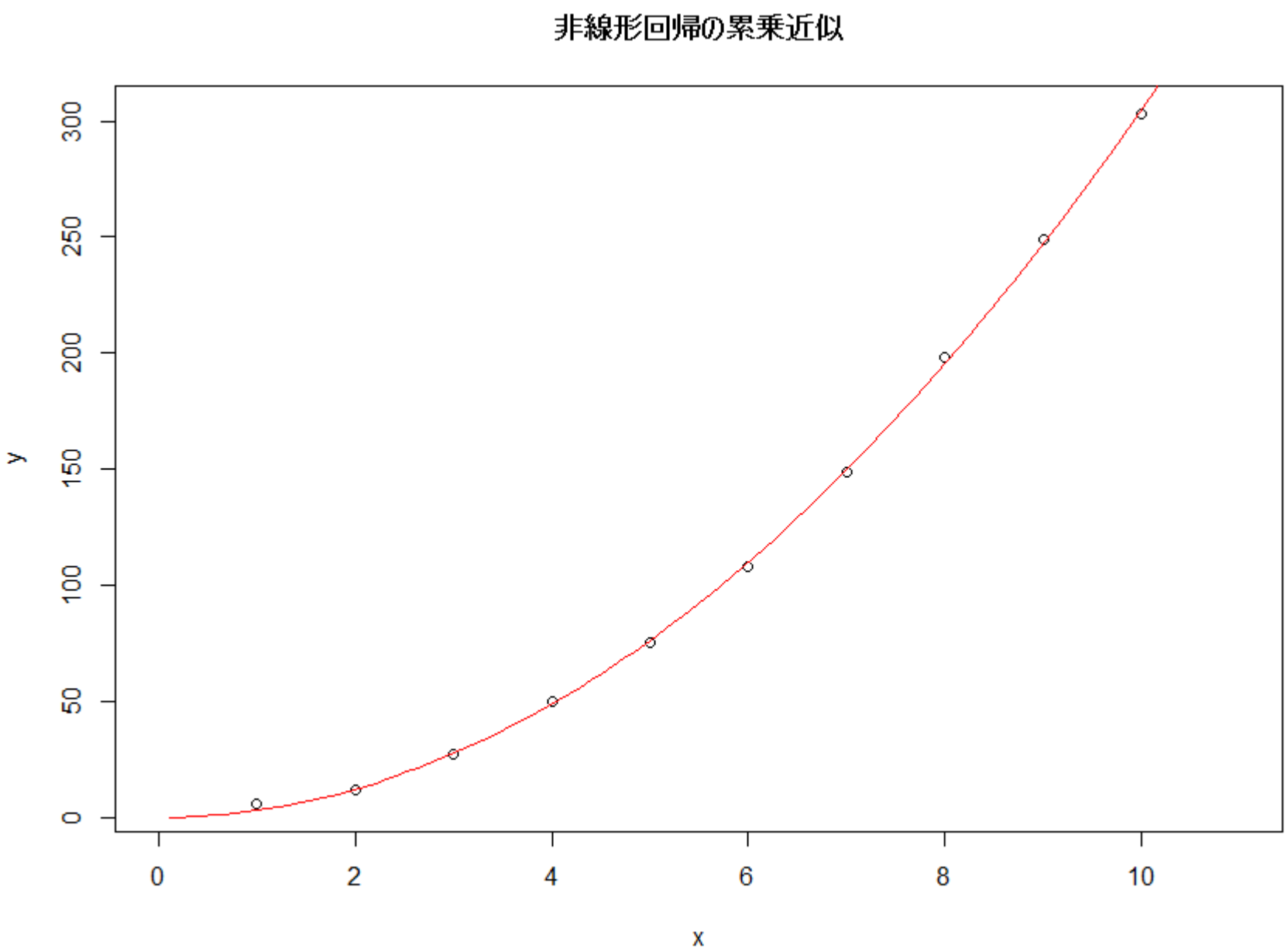
$$y = ax^b$$

ソルバーにより残差平方和が最小になる係数a,bを求める

Rによるべき乗近似とグラフ作成／コード[1]

```
##### Rのコード ########
x<- c(1: 10)
y<- c(6, 12, 27, 50, 75, 108, 149, 198, 249, 303)

# 非線形回帰
nonline<- nls(y~at*x^bt, start=c(at=1, bt=1))

# パラメータ a
(at<- coef(nonline)[1])

# パラメータ b
(bt<- coef(nonline)[2])

# 残差平方和
sum((y-at*x^bt)^2)

# データとべき乗近似を描画
plot(x, y, xlim=c(0, 11), main="非線形回帰の累乗近似")
curve(at*x^bt, add=T, col="red")
```

Rによるべき乗近似曲線

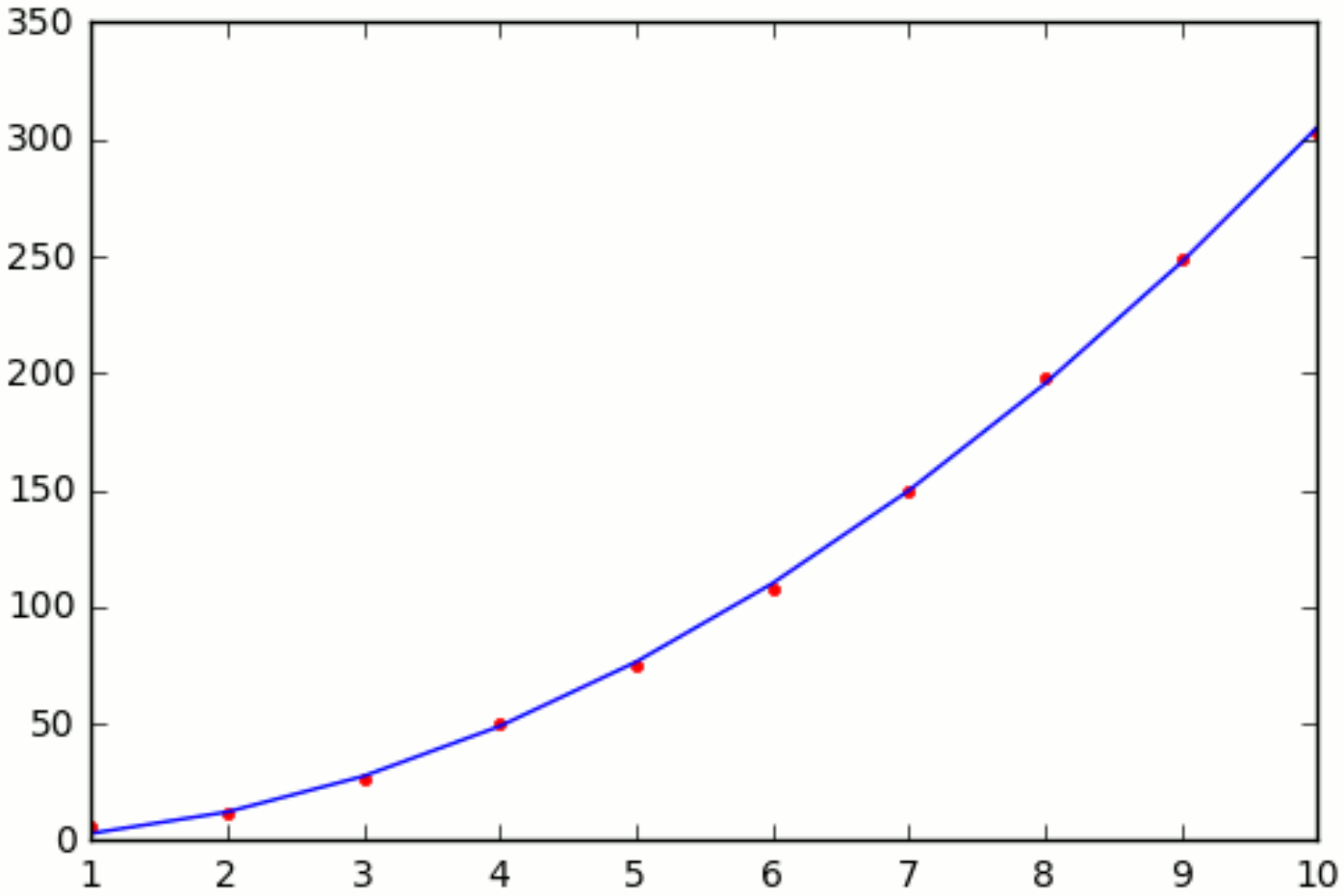


Python3／2つのモジュールによりべき乗近似を描画[3][4]

```
%matplotlib inline
# 非線形近似と描画用のモジュールの読み込み
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# データリストの作成
array_x = range(1,11)
array_y = [6, 12, 27, 50, 75, 108, 149, 198, 249, 303]

# 近似式の定義
def nonlinear_fit(x,a,b):
    return  a * x**b
param, cov = curve_fit(nonlinear_fit, array_x, array_y)

# 描画
list_y = []
for num in array_x:
    list_y.append( param[0] * num**param[1] )
plt.plot(array_x, array_y, 'r.')
plt.plot(array_x, list_y, 'b')
```

Python3／Jupyterに表示されるグラフ

```
Out[4]: [<matplotlib.lines.Line2D at 0x1d508b4ee80>]
```



まとめ

1. Excelグラフの非線形近似には注意が必要（x,yに数値を代入した状態で係数が非線形になるもの）。データにフィットさせるにはソルバーでの計算が必要となる

2. R単独（ライブラリ追加不要）で短いコードで非線形近似が可能
   複数のモジュールを組み込めば、Pythonにおいても非線形近似が可能
   いずれにおいても、コマンド操作が基本のため、「ひな形」を作成すれば敷居が下がる

3. 各ソフトの個人的な見解は下表の通り

	メリット	デメリット
Excel	最も普及しているソフトの一つ マウスの操作によりグラフを作成できる	遅い 入力と出力を同時に見えない べき乗、指数近似はソルバーが必要
R	書籍が多い ライブラリが多い 統計学のフリーソフトとして有名	コマンド操作が基本で「ひな形」が必要 Windows版は計算が遅い（Linuxではopenblasによる高速化が容易）[5]
Python	Rと比べて汎用性が高く、機械学習などに使用されている Anacondaには、Intel MKLが組み込まれ高速	コマンド操作が基本で「ひな形」が必要 近似曲線を求める程度であれば、Rの方がコードが短い Rと比べ実用的な書籍が少ない

参考資料

1. Excelグラフ累乗，指数，多項式近似の論文記載の注意
   https://sites.google.com/site/fishermultiplecomparison/excel-graph

2. フリーソフトによる偏微分方程式計算
   http://opencae.gifu-nct.ac.jp/pukiwiki/index.php?%C2%E8%A3%B4%A3%B3%B2%F3%CA%D9%B6%AF%B2%F1%A1%A7H271031

3. Pythonで非線形関数モデリング
   http://qiita.com/hik0107/items/9bdc236600635a0e61e8

4. Python SciPy : 非線形最小二乗問題の最適化アルゴリズム
   https://org-technology.com/posts/scipy-least-square-fitting.html

5. 無償BLASーLAPACKライブラリによる浮動小数点演算ベンチマーク
   http://opencae.gifu-nct.ac.jp/pukiwiki/index.php?%C2%E8%A3%B4%A3%B5%B2%F3%CA%D9%B6%AF%B2%F1%A1%A7H280206