【レポート#1】ラグランジュ補間

BV20036　大野 弘貴

令和 3 年 10 月 20 日

1　レポート内容

囲いのタイトル左版

- 標本点が次の (真の) 関数により与えられる場合を考える

$$f(x) = \frac{1}{1+2x^2}$$

- n+1 個の標本点 (x_0,f_0), (x_1,f_1), $\cdots$, (x_n,f_n) が区間 [-5, 5] において等間隔で与えられていたとする
  - 上記の真値となる関数から自分で (自動で) 標本点を作成する
- 区間 [-5, 5] を 100 等分したときの分点 101 点ラグランジュ補間を計算し、グラフを作成せよ
- n = 5, 8, 11, 14 について真値の関数, ラグランジュ補完を計算し, グラフを作成せよ
  - n は多項式の次数，標本点の数は (n+1) 個

2　ラグランジュ補間とは

ラグランジュ補間とは, n + 1 個の標本点 $(x_0, f_0), (x_1, f_1), \cdots, (x_n, f_n)$ が与えられた際に, それらすべての点を通る n 次関数の曲線を求めることで, その他の適当な点 $x(x_1 < x < x_n)$ に対応する関数値 f(x) を推定することができる補間法の一つであり, 以下を満たす.

ラグランジュ補完

- 次の n 次多項式を考える

$$f(x) := \frac{(x - x_0)(x - x_1)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_n)}{(x_j - x_0)(x_j - x1_)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_n)}$$

- この多項式は変数 x に x_j を代入すると 1 になり, $x_k(j \neg k)$ を代入すると 0 になる
  - クロネッカーデルタを用いて表すと

$$l_j(x_k) = \delta_{jk} \begin{cases} 1 & (j = k) \\ 0 & (j \neq k) \end{cases}$$

- 補間条件を満たす n 次の多項式 (ラグランジュの n 次補間多項式) は次で与えられる

$$P_n(x) = f_0 l_0(x) + f_1 l_1(x) + \cdots + f_n l_n(x) = \sum_{j=0}^{n} f_j l_j(x)$$

3　実装したコード

ソースコード 1　Lagrange.m

```
function [Pn] = Lagrange (N, x_in, f_in, xi)
```

```
%引数の説明：
%N：補間多項式の次数
%x_in：標本点の x 座標のリスト
%f_in 標本点の y 座標のリスト
%xi：ラグランジュ補間を行う点の x 座標
%戻り値Pn: xi に対するラグランジュ補間を計算した結果(Pn(xi))を戻す。
Pn = 0.0;%戻り値。とりあえず 0.0で初期化。
%最終的にPn に xi に対するラグランジュ補間を行った結果が代入されるようにする。

%以下を作成する。
%やること：与えられたxi (補間したい点の x 座標)に対する Pn(xi)を計算し,変数Pn に代入
    する。

for j = 1 : N + 1
L = 1.0;
for i = 1 : N + 1
if j ~= i
L = L * (xi - x_in(i)) / (x_in(j) - x_in(i));
end
end
Pn = Pn + L * f_in(j);
end
```

ソースコード 2 Lagrange.m

```
% Lagrange 補間のサンプルプログラム(lagrange_sample.m)
% ※Lagrange (N, x_in, f_in, xi)は別途作成して同じフォルダに保存

clear
M=100; %グラフ作成用にM 分割して M+1点のラグランジュ補間を計算

n = 5;
x_in = zeros(n+1,1);
f_in = zeros(n+1,1);

f =@(x) 1./(1 + x.^2);

for n_ = 0 : n
x_in(n_+1) = -5 + n_ * 10 / n;
f_in(n_+1) = f(x_in(n_+1));
end

N=length(x_in)-1; %N 次関数で近似 (標本点の数はN+1個)
a=x_in(1); %グラフを描く区間 [a,b]のa
b=x_in(length(x_in)); %グラフを描く区間 [a,b]のb
x_out = zeros(M+1,1); %結果のx 座標を格納する配列
f_out = zeros(M+1,1); %結果のy 座標を格納する配列
```

```
h = (b-a)/M;%グラフ作成時の刻み幅

% ---ラグランジュ補間の計算--------------------------
for i = 1:M+1
x_out(i) = a + (i-1)*h; %補間を計算する点の計算
f_out(i) = Lagrange(N, x_in, f_in, x_out(i));%ラグランジュ補間の計算(関数Lagrange は別途作成)
end

% ---ラグランジュ補間を行った結果のグラフを作成----------------------------
figure
plot(x_out, f_out, "r-*"), hold on %ラグランジュ補間の結果をplot
plot(x_in,f_in, "o", "MarkerFaceColor", "k", "MarkerEdgeColor", "k","MarkerSize", 10), hold on %標本点をplot
fplot(f, [-5 5],"b"), hold on
legend("ラグランジュ補間","標本点","真の関数"), hold off %凡例を表示
```

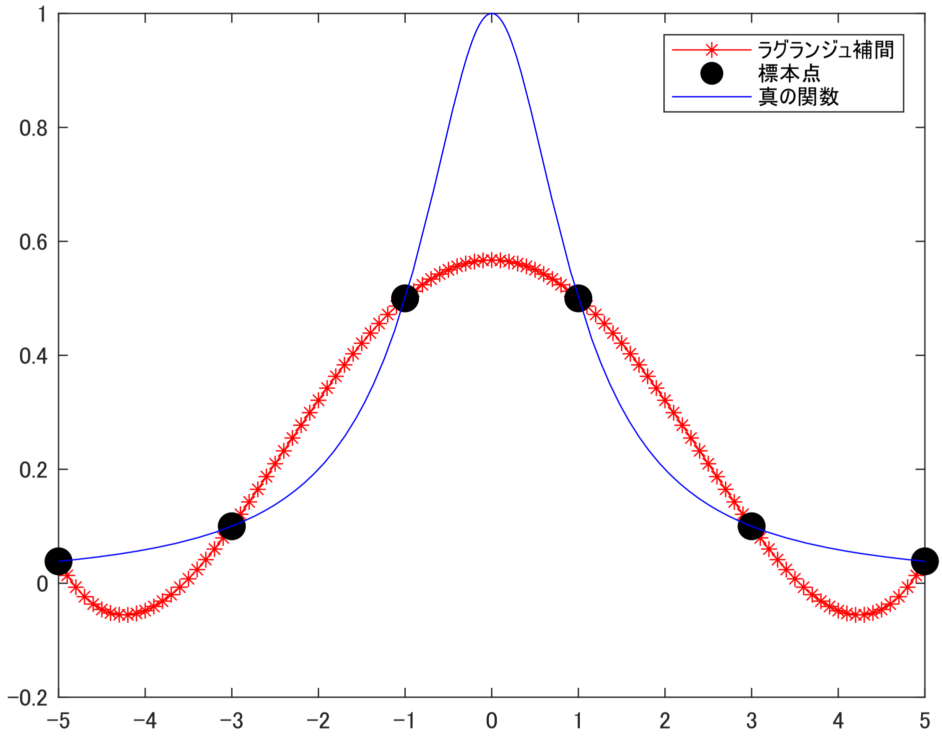

図 1　キャプション

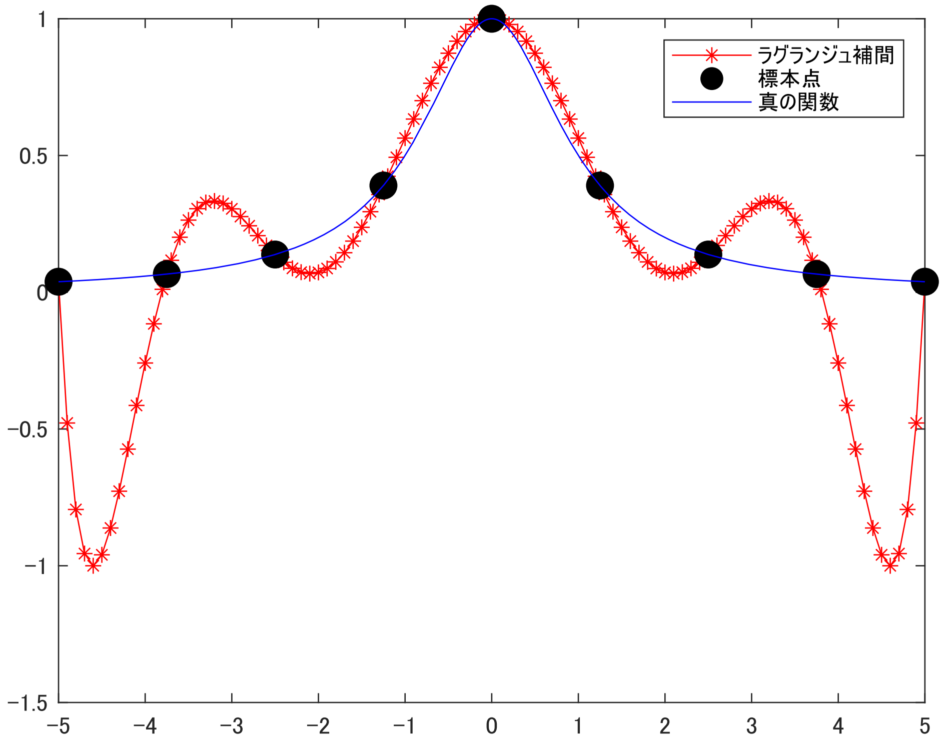

図 2　キャプション

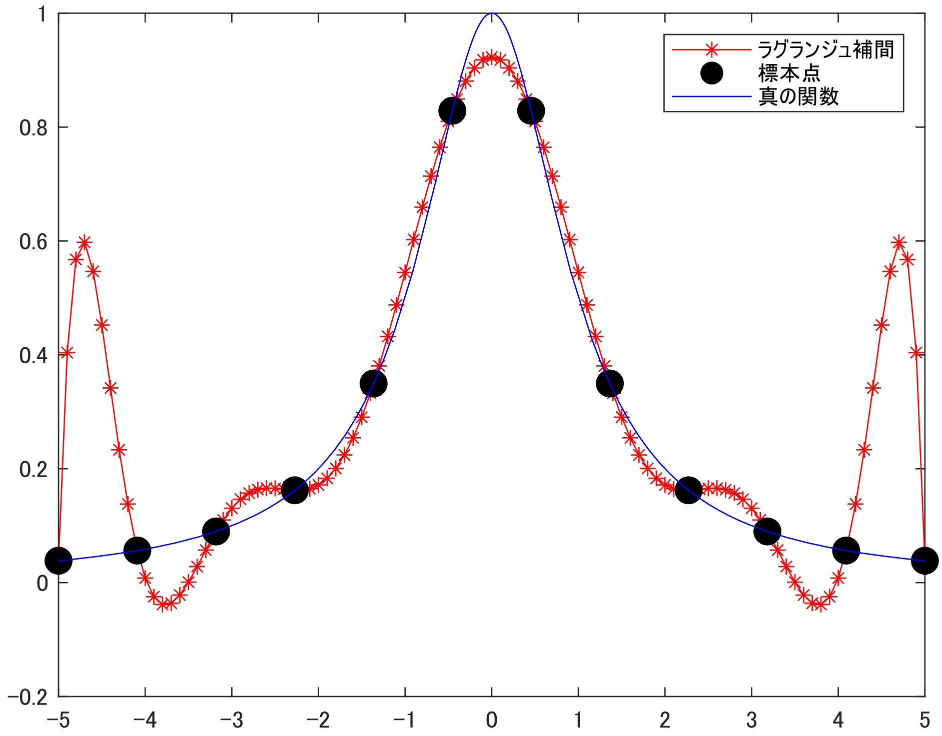

図 3　キャプション

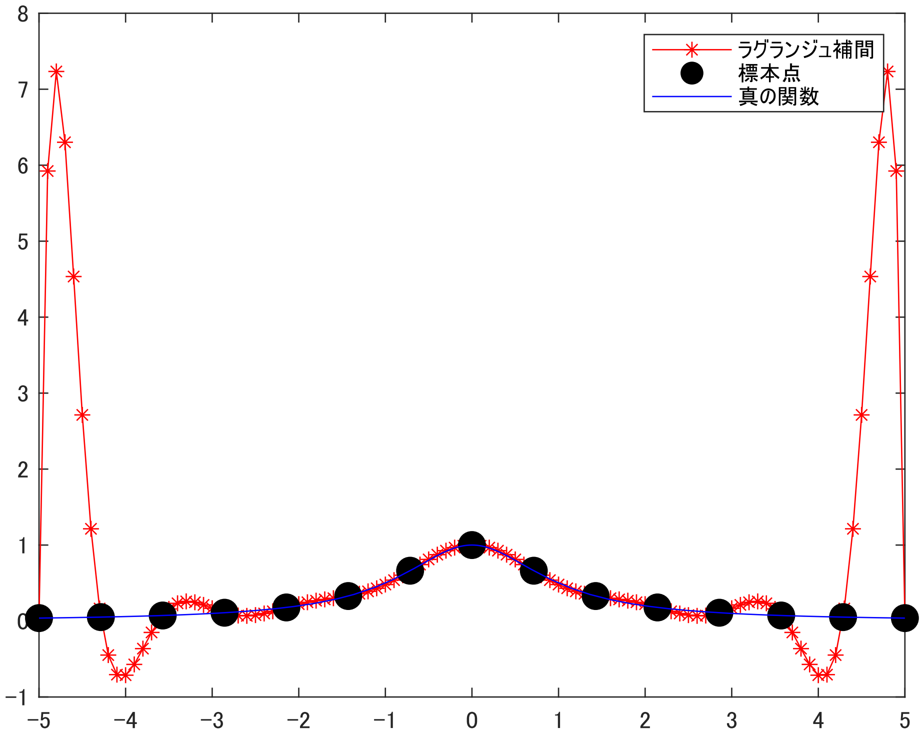

図 4　キャプション

4　考察

以上の結果から標本点が偶数、つまり n が奇数のときの方が精度が高いことがわかった。また、n の値が大きくなるほど端の値の精度が悪くなる。このことから精度を高くするためには、細かくするつまり n を大きくすればよいという安易な考えが間違っていることがわかった。また、この現象をルンゲ現象と言うらしい。このルンゲ現象を回避するためにはうまいこと間隔をとれば良いらしい。この間隔のとりかたはチェビシェフノードを使えば求まる。

また、ラグランジュ補間に限らずニュートン補完やスプライン補間では、線形補完などの単純な補間とは違い、高階の差分商が行われている。これによって桁落ちが発生しやすいことを忘れていたがとても大切な特性なことだとわかった。

参考文献

[1]　福田 亜希子, 数値解析 (II)【第 3 回】