【ラグランジュ補間　Lagrange interpolation】

n+1 個の点 (x_j, y_j)　$j = 0, 1, 2, \cdots, n$ を通る高々n 次の多項式 $L(x)$ は次のように書ける.

$$L(x) = \sum_{k=0}^{n} L_k(x) y_k$$

ただし,

$$L_k(x) = \prod_{j=0, j\neq k}^{n} \frac{x - x_j}{x_k - x_j} \quad k = 0, 1, 2. \cdots, n$$

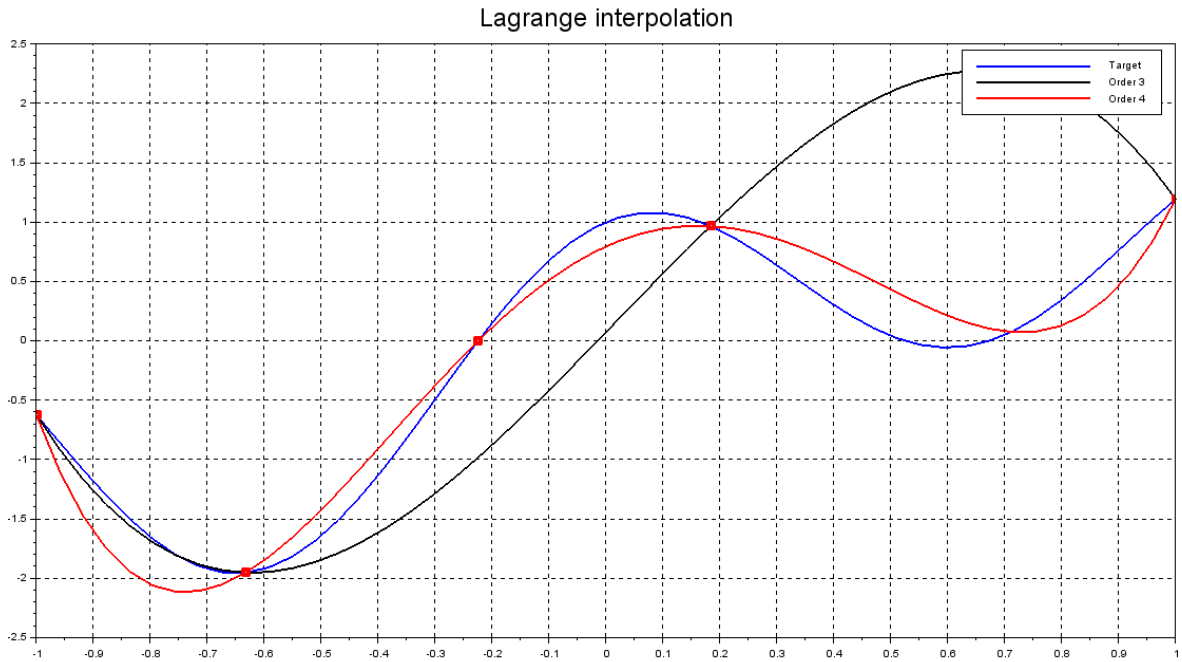

Figure 1: Scilab 実行結果
3 次，4 次比較

Source Code 1: Scilab

```
//////////////////////////////////
//      ラグランジュ補間
//      Lagrange interpolation
//
//                  M.Tsutsui
//////////////////////////////////
clear;

funcprot(0);
function[L]=Lagrange_int(order,x,y,xa);//order:次数,x:x軸サンプル抽出点,y:y軸サンプル抽出点,xa:横軸(ベクトル)

    d_size=length(xa);

    Lk=[];//Lk(x)
    L=[];//近似関数

    for i=1:1:d_size;
        for k=0:1:order;
            calc=1;
```

```
            for j=0:1:order;
                if(j~=k)
                    calc=calc*(xa(i)-x(j+1))/(x(k+1)-x(j+1));
                end
            end
            Lk=[Lk,calc];
        end
        res=sum(Lk.*y);
        L=[L,res];//補間値
        Lk=[];//初期化
    end

endfunction


d_size=50;

order=3;//次数

xt=linspace(-1,1,d_size);//x軸範囲 -1〜1

yt=sin(2*xt)+cos(5*xt);//近似対象

//次数3
x1=[xt(1),xt(10),xt(30),xt(d_size)];//x値抽出 order+1個数必要
y1=[yt(1),yt(10),yt(30),yt(d_size)];//y値抽出 order+1個数必要

//次数4
x2=[xt(1),xt(10),xt(20),xt(30),xt(d_size)];
y2=[yt(1),yt(10),yt(20),yt(30),yt(d_size)];


out=Lagrange_int(order,x1,y1,xt);//次数3 近似曲線
out2=Lagrange_int(order+1,x2,y2,xt);//次数4 近似曲線

plot(xt,yt);//元曲線
a=gce();
c=a.children;
c.thickness = 2;
plot(xt,out,'k');
a=gce();
c=a.children;
c.thickness = 2;
plot(xt,out2,'r');
a=gce();
c=a.children;
c.thickness = 2;
legend(['Target';'Order 3';'Order 4']);

plot(x2,y2,'ro');//次数4 抽出点
a=gce();
c=a.children;
c.thickness = 5;
xgrid();
title('Lagrange interpolation','fontsize',5);
```