实验六 函数插值

# 实验目的

1. 掌握拉格朗日插值、牛顿插值的基本原理
2. 理解各种插值法的优缺点和插值的误差
3. 熟悉插值法的一般过程

# 实验环境

1. 计算机
2. MATLAB集成环境

# 实验内容与代码

## 利用函数在100，121，144的值，试用线性插值、抛物线插值求的值

clc;clear;

x = [100, 121, 144]

x = 1×3

100 121 144

y = sqrt(x)

y = 1×3

10 11 12

sqrt(115)

ans = 10.7238

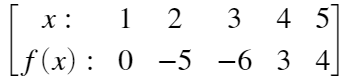
linearInterpolation(x, y, 115)

ans = 10.7143

parabolicInterpolation(x, y, 115)

ans = 10.7228

## 已知函数的观测数据如下，构造拉格朗日插值多项式，并计算的近似值



clc;clear;

x = [1 2 3 4 5]

x = 1×5

1 2 3 4 5

y = [0 -5 -6 3 4]

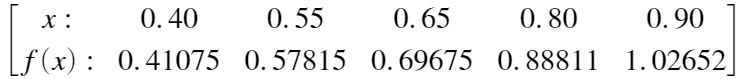
y = 1×5

0 -5 -6 3 4

lagrangeInterpolation(x, y, 1.3333)

ans = -12.6668

## 已知函数表如图，求二次、三次和四次牛顿插值多项式，并分别计算的近似值



clc;clear;

x = [0.4 0.55 0.65 0.8 0.9]

x = 1×5

0.4000 0.5500 0.6500 0.8000 0.9000

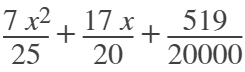
y = [0.41075, 0.57815 0.69675 0.88811 1.02652]

y = 1×5

0.4108 0.5782 0.6967 0.8881 1.0265

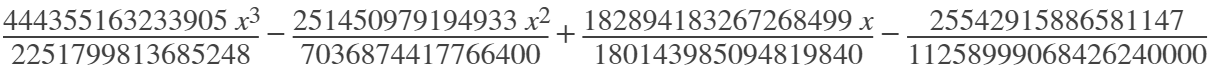
order2formula = newtonInterpolationFormula(x, y, 2)

order2formula =



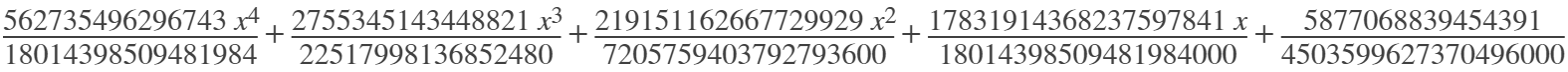
order3formula = newtonInterpolationFormula(x, y, 3)

order3formula =



order4formula = newtonInterpolationFormula(x, y, 4)

order4formula =



newtonInterpolation(x, y, 0.596)

ans = 0.6319

## 此实时脚本中使用的函数：

%{

Solve f(point) using the methods of Linear Interpolation

@Title: linearInterpolation

@param x double 1\*2 array, corresponding to the x-lable of 2 knowing points

in the function's plots

@param y double 1\*2 array, corresponding to the y-lable of 2 knowing points

in the function's plots

@param point double the x-lable of the point you want to solve

@return double

%}

function res = linearInterpolation(x, y, point)

res = y(1)\*((point-x(2))/(x(1)-x(2)))+y(2)\*((point-x(1))/(x(2)-x(1)));

end

%{

Solve f(point) using the methods of Parabolic Interpolation

@Title: parabolicInterpolation

@param x double 1\*3 array, corresponding to the x-lable of 3 knowing points

in the function's plots

@param y double 1\*3 array, corresponding to the y-lable of 3 knowing points

in the function's plots

@param point double the x-lable of the point you want to solve

@return double

%}

function res = parabolicInterpolation(x, y, point)

l1 = (point-x(2))\*(point-x(3))/((x(1)-x(2))\*(x(1)-x(3)));

l2 = (point-x(1))\*(point-x(3))/((x(2)-x(1))\*(x(2)-x(3)));

l3 = (point-x(1))\*(point-x(2))/((x(3)-x(1))\*(x(3)-x(2)));

res = l1\*y(1)+l2\*y(2)+l3\*y(3);

end

%{

Solve f(point) using the methods of Lagrange Interpolation

@Title: lagrangeInterpolation

@param x double 1d array, corresponding to the x-lable of knowing points

in the function's plots

@param y double 1d array, corresponding to the y-lable o knowing points

in the function's plots

@param point double the x-lable of the point you want to solve

@return double

%}

function res = lagrangeInterpolation(x, y, point)

res = 0;

n = length(x);

for i=1:n

xp = x([1:i-1 i+1:n]);

res = res + y(i)\*prod((point-xp)/(x(i)-xp));

end

end

%{

Solve Newton Interpolation Formula of order n.

@Title: newtonInterpolationFormula

@param x double 1d array, corresponding to the x-lable of knowing points

in the function's plots

@param y double 1d array, corresponding to the y-lable o knowing points

in the function's plots

@param order int the order of the formula you want to get

@return sym

%}

function res = newtonInterpolationFormula(x, y, order)

n = length(x);

f = zeros(n,n);

% Assign values to the first column of the difference quotient table

for k = 1:n

f(k) = y(k);

end

% Solve the difference quotient table

for i = 2:n % The difference quotient table starts from order 0; but the matrix is stored from dimension 1

for k = i:n

f(k,i) = (f(k,i-1)-f(k-1,i-1))/(x(k)-x(k+1-i));

end

end

% Solve the Newton Interpolation Formular of order n

p=0;

x\_sym = sym("x");

for k=2:order+1

t=1;

for j=1:k-1

t=t\*(x\_sym-x(j));

end

p=f(k,k)\*t+p;

end

p = f(1,1) + p;

res = simplify(p);

end

%{

Solve f(point) using the methods of Newton Interpolation

@Title: newtonInterpolation

@param x double 1d array, corresponding to the x-lable of knowing points

in the function's plots

@param y double 1d array, corresponding to the y-lable o knowing points

in the function's plots

@param point double the x-lable of the point you want to solve

@return double

%}

function res = newtonInterpolation(x, y, point)

n = length(x);

f = zeros(n,n);

% Assign values to the first column of the difference quotient table

for k = 1:n

f(k) = y(k);

end

% Solve the difference quotient table

for i = 2:n % The difference quotient table starts from order 0; but the matrix is stored from dimension 1

for k = i:n

f(k,i) = (f(k,i-1)-f(k-1,i-1))/(x(k)-x(k+1-i));

end

end

% Solve the Newton Interpolation Formular

p=0;

for k=2:n

t=1;

for j=1:k-1

t=t\*(point-x(j));

end

p=f(k,k)\*t+p;

end

p = f(1,1) + p;

res = p;

end

# 实验小结

通过此次实验，掌握了拉格朗日插值、牛顿插值的基本原理，理解了各种插值法的优缺点和插值的误差，并熟悉了插值法的一般过程。