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 $f(x) = \sqrt{x}$ 在 **100**，**121**，**144** 的值，试用线性插值、抛物线插值求 $\sqrt{115}$ 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 $f(x)$ 的观测数据如下，构造拉格朗日插值多项式，并计算 $f(1.3333)$ 的近似值

$$\begin{bmatrix} x: & 1 & 2 & 3 & 4 & 5 \\ f(x): & 0 & -5 & -6 & 3 & 4 \end{bmatrix}$$

```
clc;clear;
x = [1 2 3 4 5]
```

x = 1×5

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix}$$

```
y = [0 -5 -6 3 4]
```

y = 1×5

$$\begin{matrix} 0 & -5 & -6 & 3 & 4 \end{matrix}$$

```
lagrangeInterpolation(x, y, 1.3333)
```

ans = -12.6668

已知 $f(x) = \mathrm{sh}(x)$ 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 $f(0.596)$ 的近似值

$$\begin{bmatrix} x: & 0.40 & 0.55 & 0.65 & 0.80 & 0.90 \\ f(x): & 0.41075 & 0.57815 & 0.69675 & 0.88811 & 1.02652 \end{bmatrix}$$

```
clc;clear;
x = [0.4 0.55 0.65 0.8 0.9]
```

x = 1×5

$$\begin{matrix} 0.4000 & 0.5500 & 0.6500 & 0.8000 & 0.9000 \end{matrix}$$

```
y = [0.41075, 0.57815 0.69675 0.88811 1.02652]
```

y = 1×5

$$\begin{matrix} 0.4108 & 0.5782 & 0.6967 & 0.8881 & 1.0265 \end{matrix}$$

```
order2formula = newtonInterpolationFormula(x, y, 2)
```

order2formula =

$$\frac{7\,x^2}{25}+\frac{17\,x}{20}+\frac{519}{20000}$$

```
order3formula = newtonInterpolationFormula(x, y, 3)
```

order3formula =

$$\frac{444355163233905\,x^3}{2251799813685248}-\frac{251450979194933\,x^2}{7036874417766400}+\frac{182894183267268499\,x}{180143985094819840}-\frac{25542915886581147}{11258999068426240000}$$

```
order4formula = newtonInterpolationFormula(x, y, 4)
```

order4formula =

$$\frac{562735496296743\,x^4}{18014398509481984}+\frac{2755345143448821\,x^3}{22517998136852480}+\frac{219151162667729929\,x^2}{7205759403792793600}+\frac{17831914368237597841\,x}{18014398509481984000}+\frac{5877068839454391}{4503599627370496000}$$

```
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。