牛顿插值多项式

问题背景：

有时候测量的数据是一些点的集合，并不能很好的反映结果。为了研究其变化规律，需对这些点进行数据处理，可采用牛顿多项式法对其进行拟合。此外还有泰隆、拉格朗日等，本文以牛顿法为例。牛顿法的好处在于思路简单，易理解。

详细过程如下（建模时简写）：

设函数 $f(t)$ 在 $n+1$ 个相异点 $x_0, x_1, \cdots, x_n$ 上函数值分别为

$$f(x_0), f(x_1), \cdots, f(x_n)$$

或者记为

$$y_0, y_1, \cdots, y_n$$

一阶均差：称 $\dfrac{f(x_0)-f(x_1)}{x_0-x_1}$ 为 $f(x)$ 关于节点 x_0, x_1 的一阶均差，记为

$$f[x_0, x_1]$$

二阶均差：一阶均差 $f[x_0, x_1]$， $f[x_1, x_2]$ 的均差

$$\frac{f[x_0, x_1]-f[x_1, x_2]}{x_0-x_2}$$

记为

$$f[x_0, x_1, x_2]$$

n 阶均差：

$$f[x_0, x_1, \cdots, x_n]=\frac{f[x_0, x_1, \cdots, x_{n-1}]-f[x_0, x_1, \cdots, x_n]}{x_0-x_n}$$

将 n 阶均差表示为 $n+1$ 个函数值 $y_0, y_1, \cdots, y_n$ 的线性组合，即

$$f[x_0, x_1, \cdots x_n]=\sum_{k=0}^{n}\frac{y_k}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

表 2 均差

x_i	$f(x_i)$	一阶均差	二阶均差	$\cdots$
x_0	$f(x_0)$			$\cdots$
x_1	$f(x_1)$	$f[x_0, x_1]$		$\cdots$
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

由

$$f[x, x_0]=\frac{f(x)-f(x_1)}{x-x_1}$$

可得 0 式

$$f(x)=f(x_0)+f[x, x_0](x-x_0) \tag{4}$$

由

$$f[x,x_0,x_1]=\frac{f[x,x_0]-f[x_0,x_1]}{x-x_1}$$

得到 1 式

$$f[x,x_0]=f[x_0,x_1]+f[x,x_0,x_1](x-x_1) \tag{5}$$

同理可得到 n 式

$$f[x,x_0,\cdots,x_{n-1}]=f[x_0,x_1,\cdots,x_n]+f[x,x_0,x_1,\cdots,x_n](x-x_n) \tag{6}$$

将 n 式代入 $n-1$ 式，$n-1$ 式代入 $n-2$ 式，……，1 式代入 0 式，得到

$$\begin{aligned}f(x)=&f(x_0)+f(x_0,x_1)(x-x_0)+f(x_0,x_1,x_2)(x-x_0)(x-x_1)+\cdots\\&+f(x_0,x_1,x_2,\cdots x_n)(x-x_0)(x-x_1)\cdots(x-x_n)\end{aligned} \tag{7}$$

最后一项中，均差含 x 为余项部分，记作

$$R_n(x)=f(x_0,x_1,x_2,\cdots x_n)(x-x_0)(x-x_1)\cdots(x-x_n)$$

前 $n+1$ 项是关于 x 的 n 次多项式，记为 $N_n(x)$，即为该插值方程。

于是，式（7）成为

$$f(x)=N_n(x)+R_n(x)$$

将均差部分记为 C_n，其中 $n=1,2,3,\cdots$，于是得到

$$f(x)=C_0+C_1(x-x_0)+\cdots+C_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})+C_{n+1}(x-x_0)(x-x_1)\cdots(x-x_n)$$

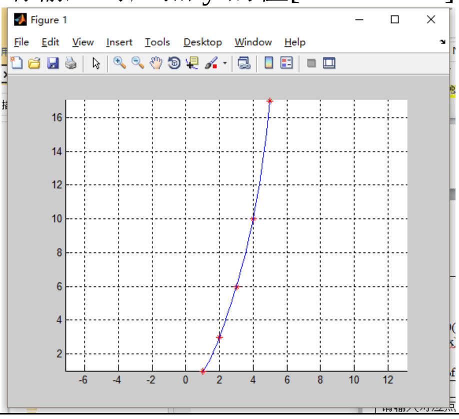
附录	运行环境：Matlab2011a

```
clc;
clear;
syms x
x0=input('请输入对应点 x 的值');
y0=input('请输入对应点 y 的值');
m = length(x0);
n= length(x0);
for t = 1 : m
A = zeros(n,n);
A(:,1) = y0';
s = 0.0;
p = 1.0;
q1 = 1.0;
c1 = 1.0;
for j = 2 : n
for i = j : n
A(i,j) = (A(i,j-1) - A(i-1,j-1))/(x0(i)-x0(i-j+1));
end
q1 = abs(q1*(x-x0(j-1)));
c1 = c1 * j;
end
C = A(n, n);
q1 = abs(q1*(x-x0(n)));
for k = (n-1):-1:1
C = conv(C, poly(x0(k))); %语法格式 w=conv(u，v)，其中 u 和 v 分别是有限长度序列向量，w 是 u 和 v 的卷积结果序列向量。如果向量 u 和 v 的长度分别为 N 和 M，则向量 w 的长度为 N+M-1.如果向量 u 和 v 是两个多项式的系数，则 w 就
```

```
是这两个多项式乘积的系数
d = length(C);
C(d) = C(d) + A(k,k);%在最后一维，也就是常数项加上新的差商
end
end
plot(x0,y0,'r*');
axis equal
grid on
hold on
x=linspace(x0(1),x0(end),20)';
y=polyval(C,x);
plot(x,y)
C           %f(x)的系数
```

运行后输入及结果

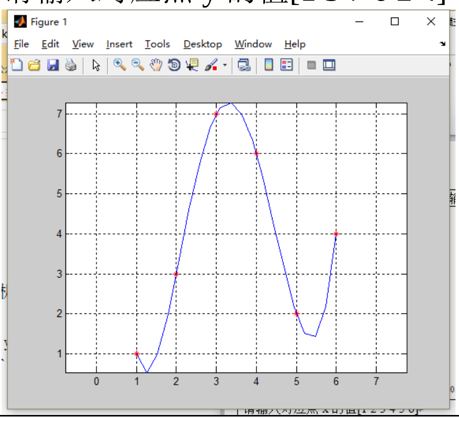
请输入对应点 x 的值[1 2 3 4 5]
请输入对应点 y 的值[1 3 6 10 17]




```
C =

    0.0833   -0.8333    3.4167   -3.6667    2.0000
```

请输入对应点 x 的值[1 2 3 4 5 6]
请输入对应点 y 的值[1 3 7 6 2 4]




```
C =

   -0.0167    0.6250   -6.3333   24.8750  -37.1500   19.0000

>>
```