

E04a 编程作业解答

姓名：任云玮 学号：516030910586

问题 由实验给出数据表. 试求 3 次、4 次多项式的曲线拟合, 再根据数据曲线形状,

x	0.0	0.1	0.2	0.3	0.5	0.8	1.0
y	1.0	0.41	0.50	0.61	0.91	2.02	2.46

表 1: 数据表

求一个另外函数的拟合曲线, 用图示数据曲线及相应的三种拟合曲线.

1 多项式拟合

1.1 简述多项式拟合的过程

函数 `orthPoly` 用于计算对于给定的样本点 `px` 的前 n 个正交多项式, 返回一个元胞数组, 其中保存了计算得的多项式. 计算是利用递推公式

$$P_{k+1}(x) = \left(x - \frac{(xP_k, P_k)}{(P_k, P_k)} \right) P_k - \frac{(P_k, P_k)}{(P_{k-1}, P_{k-1})} P_{k-1}.$$

函数 `orthCoefficient` 用于计算对于给定的样本点 `(px, py)`, 和给定的正交多项式 `P`, 计算前 n 个系数. 计算是利用公式

$$a_k = \frac{\sum_{i=0}^m y_i P_k(x_i)}{\sum_{i=0}^m P_k^2(x_i)}.$$

函数 `calcPoly` 用于对于给定的正交多项式 `P` 和对应的系数 `cof`, 计算拟合的多项式在 `qx` 处的值. 而函数 `polyfitn` 通过调用其他三个函数, 计算出样本点 `(px, py)` 对应的 n 次拟合多项式.

1.2 编写多项式拟合的函数文件, 命名为 `polyfitn.m`

```
1 function res = polyfitn(n, px, py, qx)
2     n = n + 1; % 0-based to 1-based
3     m = length(px);
4     P = orthPoly(m, px);
5     cof = orthCoefficient(n, px, py, P);
6     res = calcPoly(n, cof, P, qx);
```

```

7 end
8
9 function P = orthPoly(n, px)
10     % return the first n orthogonal polynomials
11     P = cell(n, 1);
12     P{1} = @(x) ones(1, length(x));
13     alpha = (px * ((P{1}(px)).^2)') / (P{1}(px) * (P{1}(px))');
14     P{2} = @(x) (x-alpha);
15     for i = 3:n
16         alpha = (px * ((P{i-1}(px)).^2)') / (P{i-1}(px) * (P{i-1}(px))');
17         beta = (P{i-1}(px) * (P{i-1}(px))') / ((P{i-2}(px) * (P{i-2}(px))'));
18         P{i} = @(x) ((x-alpha) .* P{i-1}(x) - beta .* P{i-2}(x));
19     end
20 end
21
22 function cof = orthCoefficient(n, px, py, P)
23     % return the first n coefficient
24     % px are the sample points and P are the orthogonal polynomials
25     cof = zeros(1, n);
26     for k = 1:n
27         cof(k) = (py * (P{k}(px))') / (P{k}(px) * P{k}(px)');
28     end
29 end
30
31 function res = calcPoly(n, cof, P, qx)
32     res = zeros(1, length(qx));
33     for i = 1:n
34         res = res + cof(i) .* P{i}(qx);
35     end
36 end

```

1.3 编写主程序，命名为 run_polyfitn.m，给出 3 次和 4 次的拟合，并用不同线型在同一幅图中画出拟合结果

见图1. 红色实线为三次多项式拟合，蓝色虚线为四次多项式拟合的结果。

```

1 drawX = linspace(0, 1, 100);
2
3 px = [0, 0.1, 0.2, 0.3, 0.5, 0.8, 1];
4 py = [1, 0.41, 0.5, 0.61, 0.91, 2.02, 2.46];
5
6 drawY = polyfitn(3, px, py, drawX);
7 hold on
8 plot(px, py, 'o');
9 plot(drawX, drawY, 'r-');
10
11 drawY = polyfitn(4, px, py, drawX);
12 plot(drawX, drawY, 'b--');
13 hold off

```

1.4 用 matlab 自带命令重复 1.3 的过程

见图2. 红色实线为三次多项式拟合，蓝色虚线为四次多项式拟合的结果。

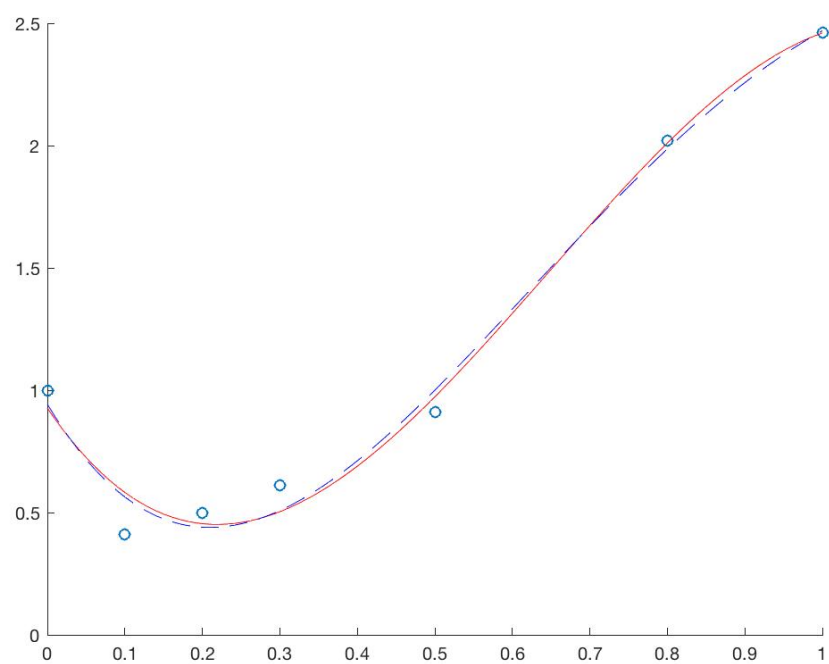


图 1: 多项式拟合结果

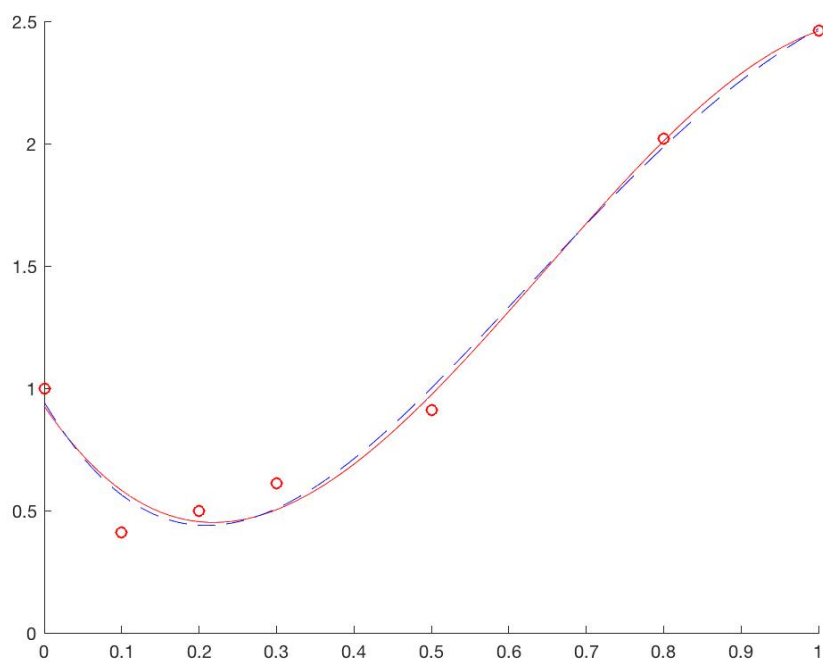


图 2: matlab 多项式拟合结果

```

1 drawX = linspace(0, 1, 100);
2
3 px = [0, 0.1, 0.2, 0.3, 0.5, 0.8, 1];
4 py = [1, 0.41, 0.5, 0.61, 0.91, 2.02, 2.46];
5
6 P = polyfit(px, py, 3);
7 hold on
8 plot(px, py, 'ro');
9 plot(drawX, polyval(P, drawX), 'r-');
10
11 P = polyfit(px, py, 4);
12 plot(drawX, polyval(P, drawX), 'b-');
13 hold off

```

2 其他函数拟合

2.1 图示数据曲线，猜测可能曲线，并给出拟合的求解过程

根据数据点，一个形如

$$f(x) = e^{P_3(x)} \Rightarrow \log f(x) = P_3(x).$$

的函数。对于数据点的 y ，都取 \log ，然后进行 3 次多项式拟合。

2.2 直接编程，画出拟合图形（程序命名为 run_ployfit_nd.m）

见图3. 红线为拟合结果。

```

1 drawX = linspace(0, 1, 100);
2
3 px = [0, 0.1, 0.2, 0.3, 0.5, 0.8, 1];
4 py = [1, 0.41, 0.5, 0.61, 0.91, 2.02, 2.46];
5 py = log(py);
6
7 drawY = polyfitn(3, px, py, drawX);
8 plot(px, exp(py), 'o', drawX, exp(drawY), 'r-');

```

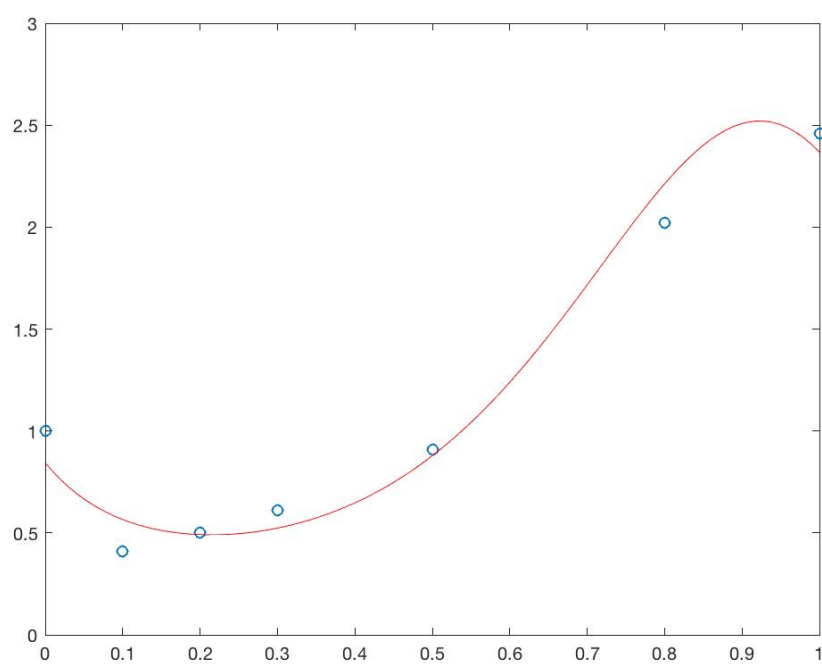


图 3: 拟合结果