基于三次样条插值法的重构光纤光栅平面曲线

摘 要

本文针对离散曲率数据，采用较好的插值算法——三次样条插值，得到曲率-弧长对数据。在一定的初始条件下，分别运用斜率递推算法和平面坐标转换法两种算法对光伏传感器曲线进行了曲线重构。其次在二维和三维坐标系下，本文综合分析了重构曲线的特征。最后，通过设置多组样本容量不同的数据，精确的探讨了重构曲线产生误差的原因，对后续误差分析给予了一定的数据支撑。

针对问题一，根据波长与曲线曲率关系公式$k=\dfrac{c(\lambda-\lambda_0)}{\lambda_0}$，分别估算出两种状态下 FBG1 到 FBG6 的曲率值，结果如表一所示；其次，假设光纤在平面内受力后 x 无变化，本文基于离散点曲率信息进行合理插值，经比较，最终选用**三次样条插值**法估算在特定横坐标位置的曲率值，结果如表二所示。

针对问题二，在问题一求出各个传感点对应的曲率值的基础上，选用三次样条插值算法得到$s\in[0,3]$的曲率-弧长对；接着分别通过**斜率递推算法**和**平面坐标变换算法**，以初始位置切线与平面方向的夹角为 45°为限定条件，对两组测试状态下的平面曲线进行重构，并将**重构曲线可视化**。经比较，两组算法下重构的平面曲线**无明显差别**。最后，为详细观察曲线图形特性，给予 z 轴正弦函数特征，将二维平面曲线放入三维平面，最后根据重构的曲线图形，分析了曲线的平滑性、对称性、围合区域、凹凸性等特征。

针对问题三，首先对平面曲线方程 $y=x^3+x(0\leqslant x\leqslant 1)$进行等间距弧长采样,将样本容量分别设置为 5、10、12、15、20、30、100，试图探讨最佳样本容量，以便找到最合适的样本容量大小从而减小误差。并将**等间距弧长对应的采样点**投影到 x 轴找出相对应的 x 的值，从而计算该值对应曲线的一阶导数y'和二阶导数y''，带入曲率公式$k=\dfrac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$，计算相对应的曲率，其次本文基于采样曲率对离散样本点的曲率进行插补，且在初始点（0,0）计算k值可知k=1。在此限定下，通过斜率递推算法重构出曲线，并与原始曲线进行可视化比较。为了比较不同样本容量下的重构平面曲线与原曲线的误差大小，计算了平均绝对误差（MAE）、均方误差（MSE）、均方根误差（RMSE）、末端误差和平均位置误差，最后，本文探讨了原始曲线与重构曲线误差产生的原因。其中样本容量大小不同确实会对误差产生一定影响，且影响为，随着样本容量增大，大部分误差先减小后增大最后趋于稳定，另部分基本不变。

关键词：三次样条插值算法、曲率-弧长对、斜率递推算法、平面坐标变换算法

目录

一、 问题重述

1.1 问题背景

光纤传感技术是伴随着光纤及光通信技术发展起来的一种新型传感器技术，光纤光栅传感器的原理是当外界环境发生变化时，会引起光纤传感器中光波参量的变化。随着越来越多的大规模光纤光栅传感器在海陆空等领域的使用，光纤光栅传感器技术变得越来越重要，例如在公路应变计形变监测中，多采用振弦传感器等等，但是这些传感器存在测量点威少、测量时长短、组网困难等等缺点。并且相比于传统的电学类传感器，光纤传感器有许多优点，比如：质地轻、体积小、弯曲性能好，易于安装、灵敏度高、抗电磁干扰能力强等等，可以感知多个物理量，比如：温度、应变等等。由于对光纤光栅技术的研究还有许多缺陷，因此对光纤光栅传感器曲线曲率研究意义非凡。

1.2 待解决的问题

本文具体要解决的问题如下：

问题一：根据题目所给数据及其公式，估算平面光栅各个传感点的曲率，并构建数学模型，估算所个表格中横坐标 x 轴相应位置处的曲率。

问题二：根据问题一求出的曲率数据，构建数学模型，将平面曲线重构，并分析曲线的特点。

问题三：根据平面曲线方程 $y=x^3+x(0\leqslant x\leqslant 1)$,取适当的等距采样，计算采样点点的曲率，重构平面曲线，分析重构曲线与原始曲线出现误差的原因。

二、 问题分析

2.1 问题一的分析

对问题一的子问题一，题目要求根据表 1 给出的测量数据，构建数学模型，估算平面光栅各个传感点(FBG1-FBG6)的曲率，由于波长与曲率之间的关系可以近似表示为 $k=\dfrac{c(\lambda-\lambda_0)}{\lambda_0}$根据表 1 的测量数据，可以计算出曲率 k。对于子问题二，假设光纤在平面受力后，光线变形前后弧长对所对应的 x 无明显变化，根据子问题一传感点(FBG1-FBG6)算出的曲率数据，可以考虑线性插值法和三次样条插值法，但是由于题目所给数据太少过于离散，使用线性插值法，不能得到较好的拟合；据此，在本论文中对于测试 1 和测试 2 两种不同的状态，分别采用三次样条插值法在$x\epsilon[0,3]$上进行插值，得到 x=0.3、0.4、0.5、0.7 的曲率。

2.2 问题二的分析

题目要求根据问题一和表 1 所得出的数据，构建数学模型，分别重构平面曲线，并且分析重构平面曲线的特点。在问题一的基础上，可得到这些数据是离散的。根据曲率的定义可知，重构曲线必须存在一阶导数和二阶导数，据此需要对问题一中的数据进行查补，再次使用三次样条插值，算出多个点的曲率值，得到曲率曲线。根据该曲率曲线的特征，问题一中所给初始位置的角度为 45°，可以采用平面坐标变换算法和斜率递推算法，可以分别得到测试状态一和测试状态二重构平面曲线。

2.3 问题三的分析

题目要求根据平面曲线方程 $y=x^3+x(0\leqslant x\leqslant 1)$，以适当的等间距弧长采样，并计算这些采样点的曲率，然后根据采样点的曲率，构建数学模型，重构平面曲线，最后分析重构曲线与原始曲线出现误差的原因。这一问要求从一个给定的平面曲线出发，考察曲率采样对重构结果的影响。首先根据所给的曲线方程求出其一阶导数和二阶导数，然后

进行适当的等间距弧长采样，将记录各个样本点所对应的 x 坐标的位置，通过算出的各个样本点横坐标 x 可以算出在这些点的一阶导数和二阶导数；其次根据曲率公式 $k=\frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$，求出各个样本点对应的曲率。再根据三次样条插值对曲率曲线进行插值，由问题二可以得出，斜率递推算法和平面坐标变换算法得出的结果几乎没有偏差，故在此问题中，我们采用斜率递推算法对平面曲线进行重构；最后将重构平面曲线和原始曲线进行对比，并计算相应误差，分析误差出现的原因。

三、 模型假设

1、对于问题一的求解过程中我们进行了假设，假设施加力前后 x 无明显变化。
2、假设曲率沿光纤的分布是光滑连续的；
3、假设光纤弯曲满足经典的欧拉贝努利梁理论。
4、假设光纤在整个长度上质地均匀，且传感器在光纤上均匀分布。
5、假设光纤的初始点（原点）的位置和方向是已知的，并且光纤的端点在实验中固定，不会发生位移。
6、在实验过程中，假设光纤传感器的灵敏度不受外界环境（如温度、湿度等）影响。
7、在建模过程中，忽略光纤自身重量对测量结果的影响，以及任何非实验设定的外力作用。
8、假设光纤仅在一个平面内发生弯曲和形变，不存在空间三维变形。

四、 符号说明

符号	说明
λ	光纤在受到外力后测量的波长
λ_0	水平光纤在初始状态下测量的波长
s_1	五个样本容量测出的弧长
s_2	八个样本容量测出的弧长
s_3	十个样本容量测出的弧长
k_1	五个样本点得出的曲率
k_2	八个样本点得出的曲率
k_3	十个样本点得出的曲率
ε_i	表示应变
h	各个测试点的距离
ρ_i	曲率半径
x_i	表示第i个点的横坐标
y_i	第i个点的横坐标

五、 问题一的模型建立与求解

5.1 计算平面光栅各个传感点的曲率

5.1.1 光纤光栅曲率算法

浏览题干可知，光纤传感器解调系统可以调解应变信息，间接求出曲率，据此可以得到曲率与应变之间存在联系，表达式如下：

$$\varepsilon_i=\frac{h}{2\rho_i} \tag{1}$$

其中 h 表示各个测试点之间的距离，ρ_i表示封装体弯曲的曲率半径。基于文献[7]所给的中心波长与应变量的关系，综合可得曲率κ_i，其表达式如下：

$$\frac{2\Delta\lambda_{Bi}}{(1-P_e)\lambda_{Bi}h} \tag{2}$$

将这个表达式化简可得题目所给公式如下：

$$\kappa=\frac{c(\lambda-\lambda_0)}{\lambda_0} \tag{3}$$

5.1.2 计算各个测试点的曲率

其中λ_0表示水平光纤在初始状态下测量的波长(此处有两个初始状态 1529 和 1540)，λ表示光线在受到外力后测量的波长，c 为某个常数（在此处设 c=4200），据此可以估算出平面光栅各个传感点（各个传感点间距为 0.6 米）的曲率，如下表 1 所示：

表 1：各个测试点的曲率

	测试 1 的曲率	测试 2 的曲率
FBG1	2.2195	2.9864
FBG2	2.2167	2.9782
FBG3	2.2332	2.9727
FBG4	2.2305	2.9809
FBG5	2.2360	2.9836
FBG6	2.2222	2.9755

5.2 估算横坐标 x 轴相应位置的曲率

对上述数据特点分析可得，可以利用插值法对 0.3、0.4、0.5、0.7 米的曲率进行估算，对此可以用线性插值和三次样条插值法进行。由于该数据不符合线性关系，用线性插值可能产生较大误差，据此，可以使用拟合程度高，误差更小的三次样条插值。三次样条插值是通过在给定数据点之间构建分段三次多项式函数，来逼近原始函数来进行插值的。这个三次项式函数具有在相邻数据点处具有连续的一阶和二阶导数，从而保证了该曲线的平滑性和连续性，在弯曲幅度较大的曲线中能够比线性曲率插值更接近于真实曲线曲率。在非传感器测量点估算曲率，使用三次样条插值法，其三次样条插值原理如下：

设$y=f(x)$在点 x_0 到 x_n 的值为 y_0 到 y_n，若函数$S(x)$满足下列条件

（1）$S(x_i)=f(x_i)=y_i$, i=0, 1, 2, ⋯, n

（2）在每个子区间$[x_i,x_{i+1}]$ (i=0, 1, 2, ⋯, n)

（3）$S(x)$在$[a,b]$上二阶连续可微

下面是构造方法：

$S(x)$除了满足基本插值条件$s(x_i)=f_i$外还具有如下以下形式：

$$S(x)=\begin{cases} S_0(x),x\in[x_0,x_1], \\ S_1(x),x\in[x_1,x_2], \quad S_i(x)\in C^3([x_i,x_{i+1}]). \\ \vdots \\ S_{n-1}(x),x\in[x_{n-1},x_n]; \end{cases} \tag{4}$$

并且满足条件：

$$\begin{cases} S_{i-1}(x_i)=S_i(x_i), \\ S'_{i-1}(x_i)=S'_i(x_i), \\ S''_{i-1}(x_i)=S''_i(x_i), \end{cases} \tag{5}$$

根据该三次多项式分别估算出在 0.3、0.4、0.5、0.7 米位置的曲率数据，如下表 2：

表 2：非测试点所得曲率

横坐标 x(米)	0.3	0.4	0.5	0.6	0.7
测试 1 曲率 k	2.2120	2.2126	2.2143	2.2167	2.2198
测试 2 曲率 k	2.9832	2.9816	2.9799	2.9782	2.9765

其可视化的图 1 为：

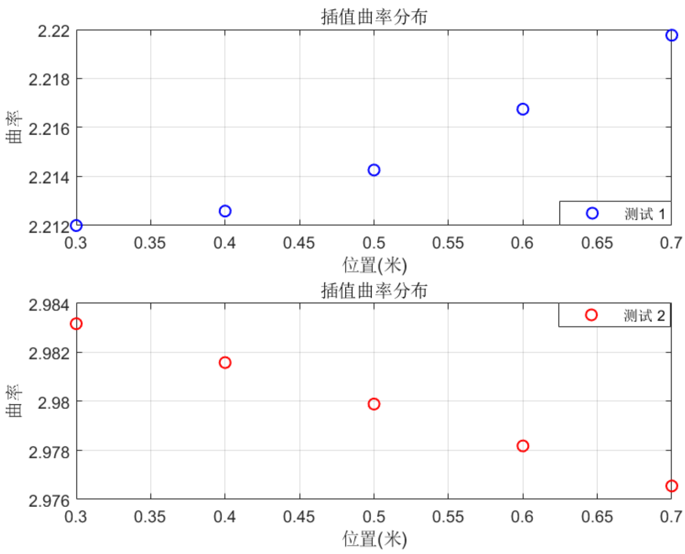


图 1

六、 问题二的模型建立与求解

6.1 离散曲率插值研究

分析表 1 和问题一所求曲率，并且可以得出这些数据都是离散的。由曲率的定义可知，曲线曲率是曲线方程在该处的二阶导数，故曲线各处曲率存在，那么该曲线一定是

连续光滑的，因此需要利用适当的插值法将这些离散的数据平滑连接。若要充分反映曲线的形态，则需要更多的曲率数据，据此采用问题一中所提到的三次样条插值法来获取完整的曲率-弧长对数据。下图 2 分别表示在测试点一和测试点二所得到的曲率-弧长对曲线。

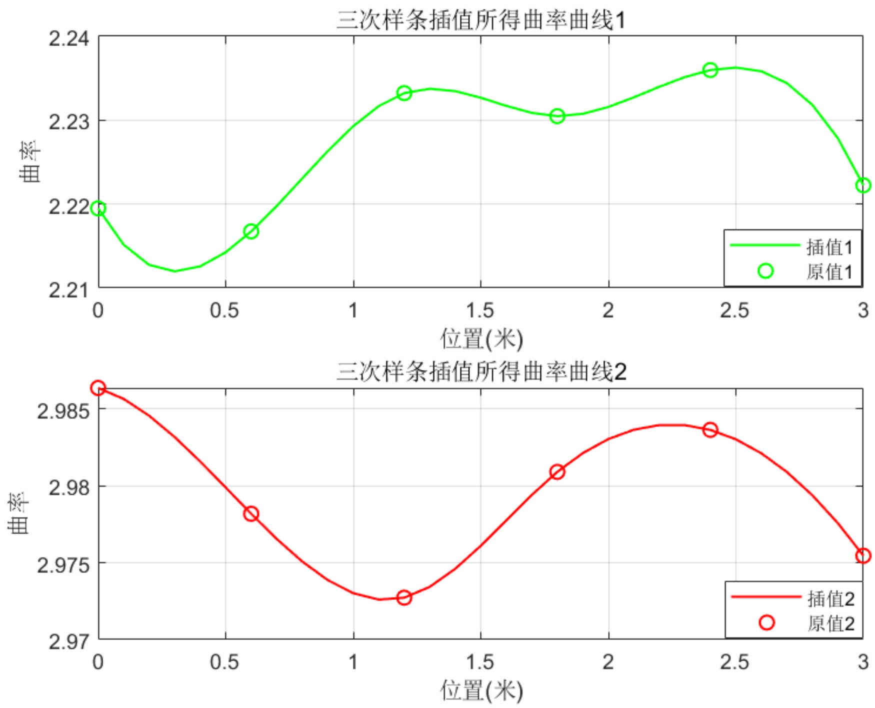


图 2

6.2 平面曲线重构

平面曲线重构是指利用光纤光栅调节系统获取的应变信息，计算光栅处的曲率，获得曲率-弧长数据对，然后将这些数据作为初始值输入并进行迭代计算，得到各个插值点的坐标，最终，利用这些插值点进行曲面重构，形成完整的曲线。此问题中我们采用的是斜率递推算法和平面坐标变换算法。图几结果是我们计算的初始曲率-弧长数据对，且根据问题一中，得知光纤在受力后的初始位置的切线与水平方向为 45°，及初始斜率为 1。

6.2.1 平面坐标变换算法

平面坐标变换算法原理：利用我们上面所求的曲率-弧长对数据，根据平面坐标变换算法的运动坐标运动思想，通过相邻两个点的相对坐标关系，如图 2-9 所示，将某点相对于上一个点为原点的相对坐标系的坐标通过旋转变换和平移变换，如公式（6）和（7）可以得到该点在其前面第 2 个点为原点的相对坐标系中的坐标，则完成了一次坐标变换，最后可以进行迭代计算，得到该点相对于固定坐标系下的绝对坐标。

$$\begin{cases} x = x'\cos\theta - y'\sin\theta \\ y = x'\sin\theta + y'\cos\theta \end{cases} \tag{6}$$

$$\begin{cases} x_i = x' + x_{i+1} \\ y_i = y' + y_{i+1} \end{cases} \tag{7}$$

下面是具体求解过程：

原理图如下：

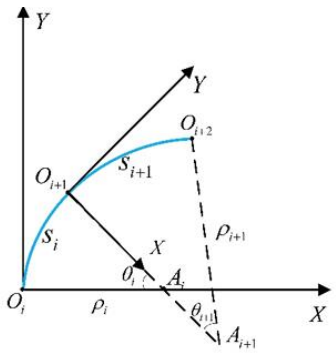


图 3

$$\begin{cases} x_{\text{tem }_{i+1}} = x_{i+1}\cos\theta_i - y_{i+1}\sin\theta_i \\ y_{\text{tem }_{i+1}} = x_{i+1}\sin\theta_i + y_{i+1}\cos\theta_i \end{cases} \tag{8}$$

$$\begin{cases} x = x_{tem_{i+1}} + x_i \\ y = y_{\text{tem }_{i+1}} + y_i \end{cases} \tag{9}$$

又由圆的几何关系：

$$\begin{cases} x_i = \rho_i - \rho_i\cos\dfrac{s_i}{\rho_i} \\ y_i = \rho_i\sin\dfrac{s_i}{\rho_i} \end{cases}, i = 1\ldots n \tag{10}$$

其中ρ_i 表示第i个点的曲率半径，s_i表示第i个点的弧长。

最后，需要将最后一个点迭代到初始固定坐标系中，就可以完成绝对位置坐标的转换，最后通过拟合完成重构。在本论文中，使用该方法对其曲率进行重构，用 MATLAB 软件得到测试一和测试二的重构平面曲线示意图 4，如下：

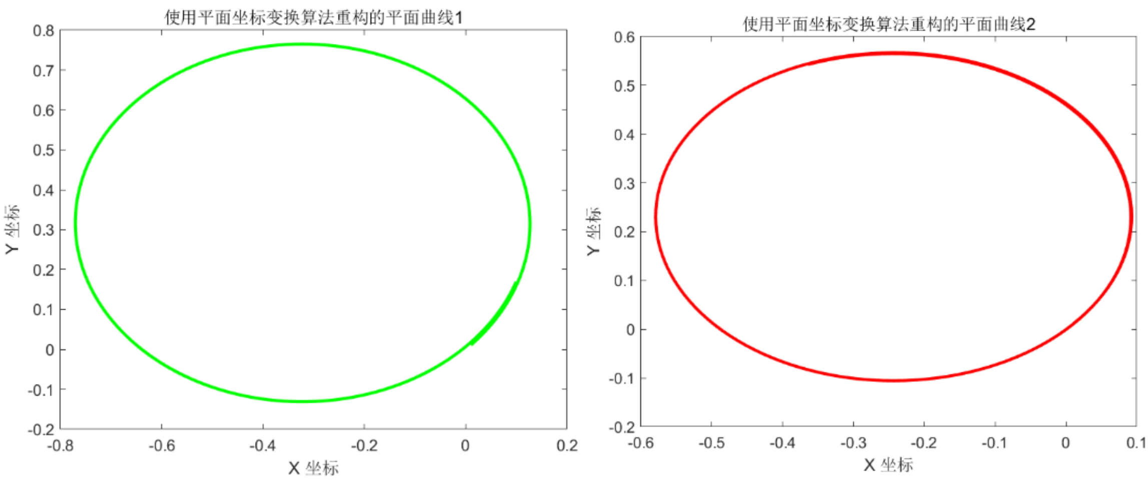


图 4

6.2.2 斜率递推算法

斜率递推算法同样也是利用我们上面所求的曲率-弧长对数据，其特点与平面坐标变换算法不同的是是以斜率作为初始条件。

下面介绍斜率递推算法的原理：

如图 7 所示，设第 n、n+1 点的曲率分别为κ_n、κ_{n+1}；斜率分别为k_n、k_{n+1}；坐标分别为(x_n,y_n)、(x_{n+1},y_{n+1})；该两点的斜率对 x 轴的夹角分别为：θ_n、θ_{n+1}；$\Delta\theta_n$为两点且相交的变化值；Δs_n为两点之间的弧长。原理图如下：

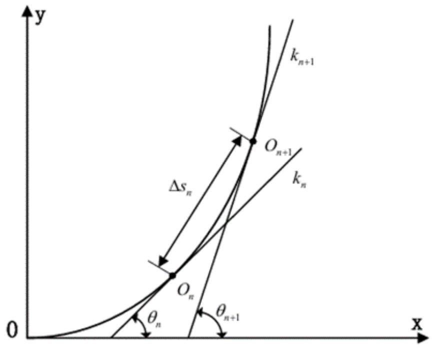


图 5

由几何关系可得：

$$\begin{cases} \theta_n = \operatorname{arctg}(k_n) \\ \theta_{n+1} = \operatorname{arctg}(k_{n+1}) \\ \Delta\theta_n = \theta_{n+1} - \theta \\ \kappa_n = \dfrac{\Delta\theta_n}{\Delta s_n} \end{cases} \tag{11}$$

由上式（11）可得：

$$\mathrm{k}_{n+1} = \operatorname{tg}[\kappa_n \Delta s_n + \operatorname{arctg}(k_n)] \tag{12}$$

只要给定最初的边界条件，就可以依据式（8）地推得出个点的斜率，并在此基础上地推下个点坐标：

$$\begin{cases} \Delta x = \dfrac{\Delta s_n}{\sqrt{1+k_n^2}} \\ \Delta y = \dfrac{k_n \Delta s_n}{\sqrt{1+k_n^2}} \end{cases} \tag{13}$$

由（13）得：

$$\begin{cases} x_{n+1} = x_n + \Delta x = x_n + \dfrac{\Delta s_n}{\sqrt{1+k_n^2}} \\ y_{n+1} = y_n + \Delta y = y_n + \dfrac{k_n \Delta s_n}{\sqrt{1+k_n^2}} \end{cases} \tag{14}$$

根据（14）可以得到各点坐标值，最后利用光滑的曲线连接，就可以得到斜率递推算法重构曲线。在本论文中使用斜率递推算法得到的平面重构曲线可视化的图见附录。

通过两种算法对光纤光栅平面曲线的重构，得出的重构平面曲线几乎重合，说明这两种算法都适用于该平面曲线的重构，且相互佐证。

6.2.3 重构平面曲线特点分析

为了更好地反映重构曲线特征，我们给予 z 轴正弦函数特征，将重构曲线从二维变到三维（如图 6），最后根据三维重构曲线，分析该曲线的相关特征：

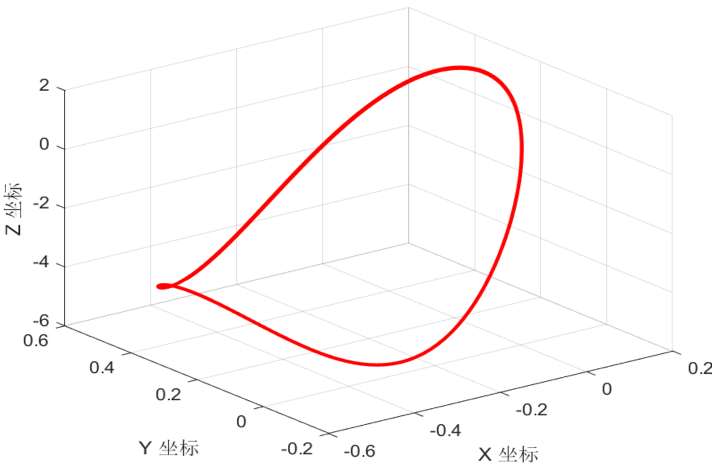


图 6 测试一的三维重构曲面

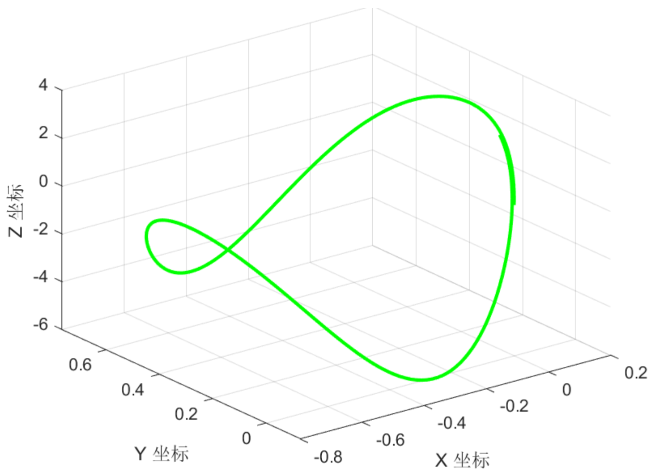


图 7 测试二的三维重构曲面

1、平滑性:该曲线具有平滑性。该重构平面曲线在其定义域内变化时的连续性和光滑程度很高，没有明显的棱角突出。

2、形状：有图形分析可得，形状是不闭合的，根据上面两种算法可知，该重构平面曲线的弯曲程度为显著弯曲，形状是类椭圆形。

3、由重构的平面曲线图象，可知该平面曲线还具有围合区域。

4、坡度和凹凸性：根据对图像的分析可知，该曲线的坡度随着位置变化的规律，并且具有凹凸性。

七、 问题三模型的建立与求解

7.1 等距取样求出曲率

7.1.1 等距弧长采样具体步骤

根据原始曲线方程 $y=x^3+x(0\leqslant x\leqslant 1)$，我们设置多组样本进行等弧长采样来找到最适合的样本容量去减小误差，样本容量分别为 5、8、10、12、20、30、100，500，并且用 MATLAB 计算出各个等距弧长取样点对应的 x 坐标，由于其他样本点数据过多不方便展示，在此只展示样本容量为 5、8、10 所对应的$s_i-x(i=1,2,3)$值，其中 S_1 表示等采样样本容量为 5 的弧长，S_2 表示等距采样样本容量为 8 的弧长，S_3 表示等距采样样本容量为 10 的弧长，如下表所示：

表 3 五个样本点对应的 S_1-x

S1	0	0.5677	1.1354	1.7030	2.2707
x	0	0.3739	0.6487	0.8466	1.0000

表 4 八个样本点对应的 S_2-x

S2	0	0.3244	0.6488	0.9732	1.2976	1.6220	1.9463	2.2707
x	0	0.2237	0.4194	0.5799	0.7114	0.8217	0.9166	1.0000

表 5 十个样本点对应的 S_3-x

S3	0	0.2523	0.5046	0.7569	1.0092	1.2615	1.5138	1.7661	2.0184	2.2707
x	0	0.1757	0.3368	0.4766	0.5958	0.6979	0.7868	0.8655	0.9360	1.0000

7.1.2 求出曲率

根据原始曲线所给方程求出一阶导数$y'=3x^2+1$和二阶导数$y''=6x$，将 7.1.1 中所求出的各个点的横坐标分别求出一阶导数和二阶导数，根据曲率公式$k=\frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$，求出各个样本点的曲率，各个样本点曲率如下表所示：

表 6 五个样本点对应的 x-k_1

x	0	0.3739	0.6487	0.8466	1.0000
K1	0	0.4286	0.2571	0.1407	0.0856

表 7 八个样本点对应的 x-k_2

x	0	0.2237	0.4194	0.5799	0.7114	0.8217	0.9166	1.0000
K2	0	0.3791	0.4134	0.3079	0.2146	0.1524	0.1122	0.0856

表 8 十个样本点对应的 x-k_3

x	0	0.1757	0.3368	0.4766	0.5958	0.6979	0.7868	0.8655	0.9360	1.0000
K3	0	0.3244	0.4321	0.3819	0.2960	0.2233	0.1702	0.1324	0.1054	0.0856

7.2 三次样条插值重构曲线

7.2.1 三次样条插值重构曲率

根据 7.1 所求的数据，我们以样本容量为 10 对曲线进行重构（举例），采用与问题一和问题二同样的三次样条插值对数据进行插补，实现对曲率曲线的构建， 插补得到的曲率曲线可视化图像如下：

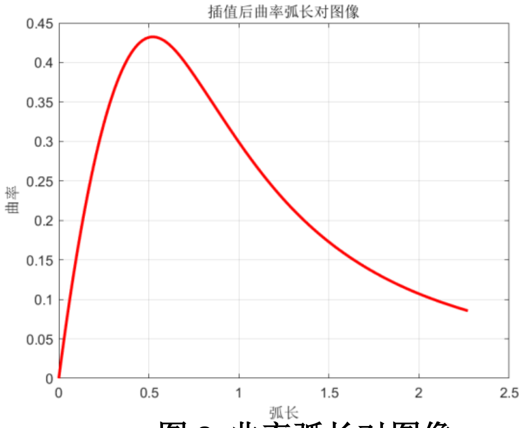


图 8 曲率弧长对图像

7.2.2 斜率递推法重构平面曲线

在上一问对于斜率递推算法和平面坐标转化法研究的时候，我们发现两种算法所构建的平面曲线几乎完全一致，进而对于此问题的研究上，我们认为采取其中一种算法就能很好的重构出平面曲线。

在问题二的求解过程中，我们已经详细介绍了斜率递推算法的原理，本小结将不在过多赘述。

在运用斜率递推算法求解的过程中，需要指定起始点的 k 值，且每次的取样点中我们均以（0,0）点为起始点，从而只需计算出（0,0）点的切线斜率即可得到限定条件。通过计算(0,0)点的一阶导数可知起始点的 k 值为 1。并将此设置为此算法的约束条件进行算法求解。

在此之前，我们在尝试求解的过程中，曾忽略过起始点的 k 值约束，导致重构的曲线与原曲线方程有较大的误差。由此可见，对起始点进行 k 值约束是很有必要的，对曲线的重构起到绝对性作用。

将通过斜率递推算法重构的曲线与原曲线进行对比，将其可视化如下图 8：

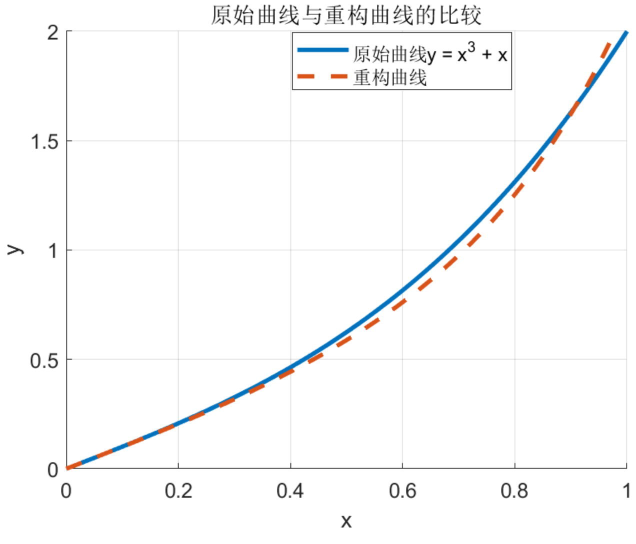


图 9 重构曲线与原曲线的比较

可以看出通过斜率递推算法重构的曲线与原始曲线几乎重合，初步说明通过斜率递推法重构的曲线的方法是正确。但仍存在不重合的部分，接下来我们对误差进行分析。

7.3 不同样本容量的误差对比和分析

在本论文试图考虑到各个因素的影响，通过查阅文献可知，方向性，末端位置变换和平均位置变换等其他因素也会影响误差，据此将从多个角度分析产生误差的原因。

以下是对各种误差原理解释：

平均绝对误差（MAE）：重构数据与原始数据之间的绝对误差的平均值，不考虑这些误差的方向，它的计算方法为将所有数据点的绝对误差相加，并除以数据点的总数，得到平均值。其公式如下：

$$MAE=\frac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right| \tag{15}$$

均方误差（MAE）：均方误差是指重构曲线数据与原始数据之间误差平方的平均值，平方误差消除了误差正负号的影响，其计算方法如下：

$$MSE=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2 \tag{16}$$

均方根误差（RMSE）：均方根误差是均方误差的平方根，表示每个数据误差的标准差的平均值，其计算方法如下：

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2} \tag{17}$$

末端误差：末端误差是指预测值与实际值在时间序列或曲线末端的差异，计算方法与平均绝对误差类似，只是针对末端数据进行计算，其计算公式如下：

$$\text{末端误差}\ =\frac{1}{k}\sum_{i=1}^{k}\left(\hat{y}_{n-k+i}-y_{n-k+i}\right)^2 \tag{18}$$

平均位置误差：通常用于评估定位或估计预测准确性，对预测位置和实际位置之间的误差进行平均，计算步骤首先确定预测位置点或者估计点以及相应的实际位置或者轨迹点，如何对每个位置点或者轨迹点，计算预测位置与实际位置之间的距离，最后将所有位置点或者轨迹点距离值相加，求其平均值，其计算公式如下：

$$\text{平均位置误差}\ =\frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(x_i-x_i'\right)^2+\left(y_i-y_i'\right)^2} \tag{19}$$

接着计算不同样本容量的平均绝对误差(MAE)、均方误差(MSE)、均方根误差(RMSE)、末端误差和平均位置误差，计算结果如下表 9 所示：

表 9 不同样本容量的五组误差

平均绝对误差	0.03771	0.02927	0.03056	0.03129	0.03165	0.03176	0.03178	0.03179	0.03179
均方误差	0.00584	0.00168	0.00162	0.00167	0.00170	0.00171	0.00171	0.00171	0.00171
均方根误差	0.07644	0.04101	0.04024	0.04085	0.04121	0.04133	0.04135	0.04136	0.04136
末端误差	0.33969	0.15574	0.11537	0.10314	0.09822	0.09682	0.09659	0.09658	0.09658
平均位置误差	0.12146	0.11386	0.11481	0.11541	0.11570	0.11579	0.11580	0.11580	0.11580

将上表中数据进行可视化处理，如下图 7：

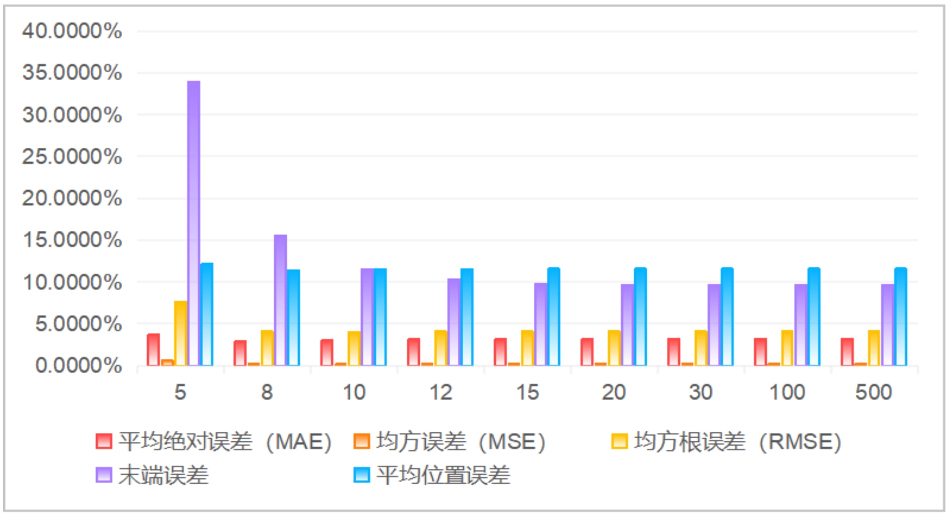


图 9 可视化后的不同样本容量的五组误差

分析各种计算的误差数据得出平均绝对误差（MAE）和均方误差（MSE）随样本容量的增大先减小后增大，最后趋于稳定；均方根误差（RMSE），末端误差，平均位置误差当样本容量达到一定值时，趋于稳定；

通过对多种样本容量误差综合分析，得出当样本容量为 10 时，平均绝对误差(MAE)和均方误差（MSE）误差最小，最后三个误差趋于稳定，上述图 6 以样本容量 10 得到的重构曲线方案是最佳的。

对比原始曲线，将这些误差综合考虑所得到的误差曲线如下图 8：

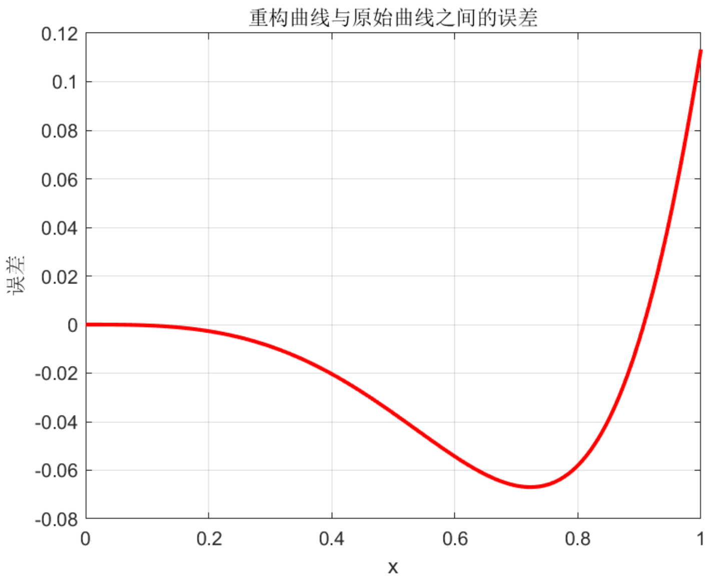


图 10 综合误差分析

据图可知，综合误差在-0.06 到 0.12 之间，说明重构曲线与原始曲线几乎重合。

7.4 产生误差的原因

1. 采样点不足，可能无法预测到某一突变值，导致误差增加。

2. 插值方法的选择：本论文中采用的是三次样条插值，即便与原始曲线几乎重合，但仍然避免不了该插值法所产生的微小误差。

3. 模型假设的选择，在此问题中，采用了斜率递推算法，对曲线进行重构时，由于曲率弧长对数据是由插值所得不能完全反应原曲线上的曲率特征，从而造成误差。

八、 模型评价与改进

8.1 模型优点：

1. 使用三次样条插值：这种方法在处理离散数据时能够生成光滑的曲线，尤其适用于在不同横坐标位置进行曲率估算，能够有效地逼近真实曲线，保持曲率变化的连续性。
2. 斜率递推与平面坐标变换算法结合：通过这种方法重构曲线可以更好地保留原始数据的几何特性，尤其是在初始切线方向设定为特定角度（如 45 度）时，通过比较两种算法可以更准确地模拟实际情况，且两种算法存在互相印证结果的作用，增加了重构曲线准确性。
3. 三维曲线展示：将二维曲线转换为三维显示，通过加入 z 轴的正弦函数特征，增加了视觉效果，有助于更直观地分析曲线的特性，如平滑性和对称性。
4. 综合误差分析：通过查找文献，考虑了多种误差度量（MAE、MSE、RMSE 等）并在文献的基础上加入了其他误差，这有助于全面评估模型的精度和可靠性。
5. 对样本容量的影响进行了探讨，为模型的改进提供了数据支持。

8.2 模型缺点：

1. 样条插值的局限性：虽然三次样条插值能提供光滑曲线，但在极值点或拐点处可能会出现过度拟合的现象，即可能在某些区域内引入不真实的波动。
2. 对曲率计算的假设：在曲率和波长之间的关系上可能存在一定的简化，这种假设可能会影响曲率的计算精度。

8.3 模型的改进：

1. 曲率计算方法的改进：考虑使用更灵活的曲率计算方法，例如非参数方法或机器学习方法，以适应更复杂或噪声更大的数据环境。
2. 曲线重构算法的优化：探索除三次样条插值之外的其他曲线拟合技术，如 Bezier 曲线或 B-splines，这些方法可能在处理特定类型的数据时表现更好。
3. 更复杂的三维曲线分析：在三维空间中考虑更多的几何和物理属性，如扭曲度或局部弯曲强度，以提供更多维度的曲线特性分析。

九、 参考文献

[1]田金容.基于多芯光纤和光频域反射的三维曲线重构方法研究[D]. 华 中科技大学,2022.

[2]Shuyang C, Fengze T, Weimin L, et al. Deep learning-based ballistocardiography reconstruction algorithm on the optical fiber sensor[J]. Optics express, 2022, 30 (8): 13121-13133.

[3]吕安强,黄崇武,乐彦杰,等.基于分布式应变的三芯光纤形态重构算法 研究[J].光电子·激光, 2021, 32(07): 784-790.

[4]程文胜.基于超弱光纤光栅的曲线重构方法研究[D].三峡大学, 2021.

[5]冯荻.基于光纤光栅应变传感的结构变形重构技术研究[D].大连理工 大学, 2020.

[6]陈世凯.光纤光栅重构方法研究及实验[D].国防科学技术大学, 2016.

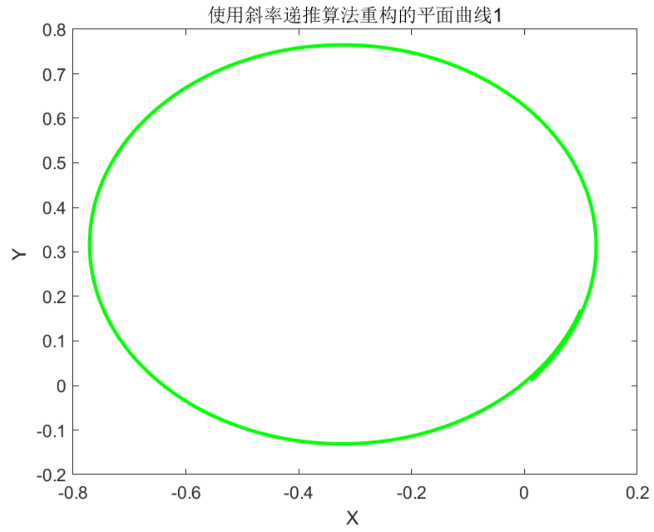
[7]章亚男,肖海,沈林勇. 用于光纤光栅曲线重建算法的坐标点拟合[J]. 光学精密工程, 2016, 24(09):2149-2157.

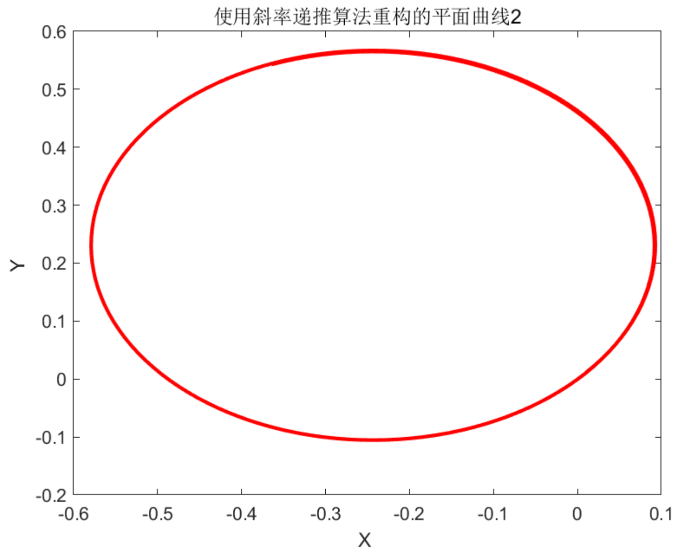
[8]肖海,章亚男,沈林勇,等.光纤光栅曲线重建算法中的曲率连续化研究 [J].仪器仪表学报,2016,37(05):993-999

附录

附录 1
介绍：支撑材料的文件列表
1. 重构平面曲线 2. 程序代码

附录 2
介绍：用斜率递推算法不同初始状态下的重构平面曲线图





附录 3
问题 1 代码（matlab）

```
% 使用三次样条插值估算曲率结果
%原始数据
bochang_cs_1 = [1529, 1529, 1529, 1529, 1529, 1529];
bochang_ts_1  =  [1529.808,  1529.807,  1529.813,  1529.812,  1529.814,
1529.809];
bochang_cs_2 = [1540, 1540, 1540, 1540, 1540, 1540];
bochang_ts_2  =  [1541.095,  1541.092,  1541.090,  1541.093,  1541.094,
1541.091];
c = 4200;
%求曲率
qulv_1 = c * (bochang_ts_1 - bochang_cs_1) ./ bochang_cs_1;
qulv_2 = c * (bochang_ts_2 - bochang_cs_2) ./ bochang_cs_2;
po = 0.6 * (0:5);
qulv_po = [0.3, 0.4, 0.5, 0.6, 0.7];
new_ka_1 = spline(po, qulv_1, qulv_po);
new_ka_2 = spline(po, qulv_2, qulv_po);
disp('测试 1 在指定位置的插值曲线:');
disp(array2table([qulv_po; new_ka_1]' , 'VariableNames',{'位置', '曲率
'}));
disp('测试 2 在指定位置的插值曲线:');
disp(array2table([qulv_po; new_ka_2]', 'VariableNames', {'位置', '曲率
'}));

subplot(2,1,1);
plot(qulv_po, new_ka_1, 'bo', 'LineWidth', 1, 'DisplayName', '测试 1');
title('插值曲率分布');
xlabel('位置(米)');
ylabel('曲率');
legend show;
grid on;

subplot(2,1,2);
plot(qulv_po, new_ka_2, 'ro', 'LineWidth', 1, 'DisplayName', '测试 2');
title('插值曲率分布');
xlabel('位置(米)');
ylabel('曲率');
legend show;
grid on;
```

附录 4
问题 2 代码（matlab）

```
%用三次样条插值求测试 1，2 的曲率曲线
bochang_cs_1 = [1529, 1529, 1529, 1529, 1529, 1529];
bochang_ts_1 = [1529.808, 1529.807, 1529.813, 1529.812, 1529.814, 1529.809];
bochang_cs_2 = [1540, 1540, 1540, 1540, 1540, 1540];
bochang_ts_2 = [1541.095, 1541.092, 1541.090, 1541.093, 1541.094, 1541.091];
c = 4200;

qulv_1 = c * (bochang_ts_1 - bochang_cs_1) ./ bochang_cs_1;
qulv_2 = c * (bochang_ts_2 - bochang_cs_2) ./ bochang_cs_2;
po = 0.6 * (0:5);

%更精细的插点
qulv_po = 0:0.1:3;
new_ka_1 = spline(po, qulv_1, qulv_po);
new_ka_2 = spline(po, qulv_2, qulv_po);

subplot(2,1,1);
plot(qulv_po, new_ka_1, 'g-', 'LineWidth', 1, 'DisplayName', '插值 1');
hold on;
plot(po, qulv_1, 'go', 'LineWidth', 1, 'DisplayName', '原值 1');
title('三次样条插值所得曲率曲线 1');
xlabel('位置(米)');
ylabel('曲率');
legend show;
grid on;

subplot(2,1,2);
plot(qulv_po, new_ka_2, 'r-', 'LineWidth', 1, 'DisplayName', '插值 2');
hold on;
plot(po, qulv_2, 'ro', 'LineWidth', 1, 'DisplayName', '原值 2');
title('三次样条插值所得曲率曲线 2');
xlabel('位置(米)');
ylabel('曲率');
legend show;
grid on;
```

附录 5

问题 2 代码（matlab）

```
% 计算测试 1 的重构曲线（斜率递推算法）
c = 4200;
bochang_cs_1 = [1529, 1529, 1529, 1529, 1529, 1529];
bochang_ts_1 = [1529.808, 1529.807, 1529.813, 1529.812, 1529.814,
1529.809];
po = 0.6 * (0:5); % 传感器位置,此时的位置是弧长

%计算曲率
qulv_1 = c * (bochang_ts_1 - bochang_cs_1) ./ bochang_cs_1;
%使用样条插值在更细密的点上估算曲率
po_1 = linspace(0, max(po), 200);
fy = spline(po, qulv_1, po_1);
%初始化，重新建立平面
x = 0; y = 0;
toi = pi/4;
ds = po_1(2) - po_1(1);
x_cs = zeros(1, length(po_1));
y_cs = zeros(1, length(po_1));
%%%%斜率递推算法
for i = 1:length(po_1)
    toi = toi + fy(i) * ds;
    x = x + cos(toi) * ds;
    y = y + sin(toi) * ds;
    x_cs(i) = x;
    y_cs(i) = y;
    if x > 3
        break;
    end
end
%做出图像
figure;
plot(x_cs, y_cs,'g','LineWidth', 2);
title('使用斜率递推算法重构的平面曲线 1');
xlabel('X ');
ylabel('Y ');
grid on;
```

附录 6

问题 2 代码（matlab）

```matlab
%计算测试 2 重构曲线（斜率递推算法）
c = 4200;
bochang_cs_2 = [1540, 1540, 1540, 1540, 1540, 1540];
bochang_ts_2 = [1541.095, 1541.092, 1541.090, 1541.093, 1541.094,
1541.091];
qulv_2 = c * (bochang_ts_2 - bochang_cs_2) ./ bochang_cs_2;

po = 0.6 * (0:5);
po_1 = linspace(0, max(po), 200);
fy = spline(po, qulv_2, po_1);
x = 0; y = 0;
%设置初始角度
toi = pi/4;
ds = po_1(2) - po_1(1);
x_cs = zeros(1, length(po_1));
y_cs = zeros(1, length(po_1));
for i = 1:length(po_1)
    toi = toi + fy(i) * ds;
    x = x + cos(toi) * ds;
    y = y + sin(toi) * ds;
    x_cs(i) = x;
    y_cs(i) = y;
    if x > 3
        break;
    end
end
figure;
plot(x_cs, y_cs,'r','LineWidth', 2);
title('使用斜率递推算法重构的平面曲线 2');
xlabel('X ');
ylabel('Y ');
grid on;
```

附录 7
问题 2 代码（matlab）

```
%计算测试 1 重构曲线（平面坐标变换算法）
% 假设的曲率数据和传感器位置
po = 0.6 * (0:5); % 六个传感器位置，每 0.6 米一个
qulv_1 = [2.2195 2.2167 2.2332 2.2305 2.2360 2.2222]; % 模拟的曲率数据
% 使用三次样条插值生成曲率函数
po_1 = linspace(0, max(po), 200); % 尽可能将插入数据细化
fy = spline(po, qulv_1, po_1);
x = 0; y = 0; % 起始坐标
% 初始斜率
toi = pi/4;
ds = po_1(2) - po_1(1);
x_s = zeros(1, length(po_1));
y_s = zeros(1, length(po_1));
%%%使用平面坐标变换算法重构曲线
for i = 1:length(po_1)
    toi = toi + fy(i) * ds;
    x = x + cos(toi) * ds;
    y = y + sin(toi) * ds;
    z = 5 * sin(x / max(po_1) * 4 * pi);
    x_s(i) = x;
    y_s(i) = y;
    z_s(i) = z;
end
figure;
plot(x_s, y_s,'g','LineWidth', 2);
title('使用平面坐标变换算法重构的平面曲线 1');
xlabel('X 坐标');
ylabel('Y 坐标');
grid on;
figure;
plot3(x_s, y_s, z_s,'g','LineWidth', 2);
title('使用平面坐标变换算法重构的三维平面曲线 1');
xlabel('X 坐标');
ylabel('Y 坐标');
zlabel('Z 坐标');
grid on;
```

附录 8

问题 2 代码（matlab）

```
%计算测试 2 重构曲线（平面坐标变换算法）
po = 0.6 * (0:5);
qulv_2 = [2.9864 2.9782 2.9727 2.9809 2.9836 2.9755];
po_1 = linspace(0, max(po), 200);
fy = spline(po, qulv_2, po_1);
x = 0; y = 0;
toi = pi/4;
ds = po_1(2) - po_1(1);
x_s = zeros(1, length(po_1));
y_s = zeros(1, length(po_1));
for i = 1:length(po_1)
    toi = toi + fy(i) * ds;
    x = x + cos(toi) * ds;
    y = y + sin(toi) * ds;
    z = 5 * sin(x / max(po_1) * 4 * pi);
    x_s(i) = x;
    y_s(i) = y;
    z_s(i) = z;
end
%画出重构曲线
figure;
plot(x_s, y_s,'r','LineWidth', 2);
title('使用平面坐标变换算法重构的平面曲线 2');
xlabel('X 坐标');
ylabel('Y 坐标');
grid on;
figure;
plot3(x_s, y_s, z_s,'r','LineWidth', 2);
title('使用平面坐标变换算法重构的三维平面曲线 1');
xlabel('X 坐标');
ylabel('Y 坐标');
zlabel('Z 坐标');
grid on;
```

附录 9
问题 3 代码（matlab）

```
%%在原弧长上取 20 个点，利用曲率弧长对重构原曲线，并比较两者误差
%重构原曲线
x = linspace(0, 1, 1000);

%对原函数求导
y_1 = 3*x.^2 + 1;
y_2 = 6*x;
ds = sqrt(1 + y_1.^2);
s = cumtrapz(x, ds);
t_h = s(end);
%设置取点个数，此时分为两类
n_s = 10;%最终选取的取点个数
%n_s = 50;
s_th = t_h / (n_s - 1);
a_p = linspace(0, t_h, n_s);
sa_x = interp1(s, x, a_p, 'pchip');
sa_x = max(min(sa_x, max(x)), min(x));

s_y_1 = interp1(x, y_1, sa_x, 'pchip');
s_y_2 = interp1(x, y_2, sa_x, 'pchip');
qulv_k = s_y_2 ./ (1 + s_y_1.^2).^(3/2);
disp('样本 x 值和相应的曲率：');
disp(table(sa_x', qulv_k', 'VariableNames', {'样品 X', '曲率'}));

%插入 x 值
x = linspace(0, 1, 1000);
y = x.^3 + x;
sa_x = interp1(s, x, a_p, 'pchip');
sa_y = sa_x.^3 + sa_x;

%插入曲率弧长对
ap_1=linspace(0, max(a_p), 200);
new_ka_1=spline(a_p, qulv_k, ap_1);
%绘制曲率弧长对图形
figure;
plot(ap_1,new_ka_1 , 'r-', 'LineWidth', 2);
title('插值后曲率弧长对图像');
xlabel('弧长');
ylabel('曲率');
grid on;

%此时可以作出在原始曲线上插入的点
```

```
figure;
plot(x, y, 'b-', 'LineWidth', 2);
hold on;
plot(sa_x, sa_y, 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
legend('原始曲线 y = x^3 + x', '样本点');
title('等间距弧长采样的原始曲线');
xlabel('x');
ylabel('y');
grid on;
%Step 1:计算
s_y_1 = 3 * sa_x.^2 + 1;
s_y_2 = 6 * sa_x;
qulv_k = s_y_2 ./ (1 + s_y_1.^2).^(3/2);

% Step 2: 进行插值
x_de = linspace(0, 1, 1000);
d_y_1 = spline(sa_x, s_y_1, x_de);
de_k = spline(sa_x, qulv_k, x_de);
n_pos = length(x_de);
tha = zeros(1, n_pos);
x_rd = zeros(1, n_pos);
y_rd = zeros(1, n_pos);
tha(1) = pi / 4;
x_rd(1) = 0;
y_rd(1) = 0;
ds = (s(end) - s(1)) / n_pos;
for i = 1:n_pos-1
    tha(i+1) = tha(i) + de_k(i) * ds;
    x_rd(i+1) = x_rd(i) + cos(tha(i)) * ds;
    y_rd(i+1) = y_rd(i) + sin(tha(i)) * ds;
end

figure;
plot(x_rd, y_rd, 'r--', 'LineWidth', 2);
title('利用精确的弧长和曲率重建曲线');
xlabel('X 坐标');
ylabel('Y 坐标');
grid on;
x_de = linspace(0, 1, 1000);
y_o = x_de.^3 + x_de;

figure;
hold on;
```

```
plot(x_de, y_o, 'LineWidth', 2.5, 'Color', [0 0.4470 0.7410], 'DisplayName', '原始曲线 y = x^3 + x');
plot(x_rd, y_rd, '--', 'LineWidth', 2.5, 'Color', [0.8500 0.3250 0.0980], 'DisplayName', '重构曲线');
legend('show');
title('原始曲线与重构曲线的比较');
xlabel('x');
ylabel('y');
grid on;

set(gca, 'FontSize', 12);
set(gcf, 'Color', 'w');
hold off;
%%误差求解
y_red_p = interp1(x_rd, y_rd, x_de, 'spline');
errors = y_red_p - y_o;
mae = mean(abs(errors));
mse = mean(errors.^2);
rmse = sqrt(mse);
%显示误差
disp(['平均绝对误差 (MAE): ', num2str(mae)]);
disp(['均方误差 (MSE): ', num2str(mse)]);
disp(['均方根误差 (RMSE): ', num2str(rmse)]);
%作出数值上误差曲线
figure;
plot(x_de, errors, 'r-', 'LineWidth', 2);
title('重构曲线与原始曲线之间的误差');
xlabel('x');
ylabel('误差');
grid on;

y_red_p = interp1(x_rd, y_rd, x_de, 'spline');
end_error = sqrt((x_rd(end) - x_de(end))^2 + (y_red_p(end) - y_o(end))^2);
position_errors = sqrt((x_rd - x_de).^2 + (y_red_p - y_o).^2);
average_position_error = mean(position_errors);

disp(['末端误差: ', num2str(end_error)]);
disp(['平均位置误差: ', num2str(average_position_error)]);
%作出位置误差曲线
figure;
plot(x_de, position_errors, 'r-', 'LineWidth', 2);
title('曲线位置误差');
xlabel('x');
```

```
ylabel('位置误差');
grid on;
```