

队伍编号	202406603361
题号	(C)

基于光纤传感器的平面曲线重建算法建模 摘要

随着市场对高精度传感器的要求逐渐提高，光纤光栅传感器技术应运而生。本文运用多种插值及拟合的思想实现了对光栅曲率估算模型的建立以及对平面曲线的重构，并运用建立的系统进行求解问题。

对于问题一，首先进行**数据预处理**，并通过处理得到原始数据，利用波长与曲率间的计算公式计算各个传感点（FBG1-FBG6）的曲率。然后通过求得的各个传感点的曲率分别采用球面插值、多项式插值、**三次样条插值**三种方法对曲率估算模型进行建立并求解。最后，对三种插值方法的求得的结果进行了对比，对**误差原因**进行了**分析**。另外，本文还总结了三种算法的优劣及使用场景。

对于问题二，旨在通过波长测量数据和已求得的曲率信息，使用**数值积分法**重构光纤在受力作用下形成的平面曲线。该过程首先定义了坐标系和初始条件，然后采用**欧拉方法**迭代计算传感器点的坐标，进而分析曲线的曲率变化、形状和位移。此外，对曲线的平滑度、对称性、周期性和局部与全局特征进行了深入分析，以确保**重构曲线**的准确性和实用性。通过这一过程，研究得出了光纤在受力后的精确曲线形状，并对曲线的特性进行了全面**评估**。

对于问题三，首先利用**等弧长间距采样**得到数个采样点，并利用曲率计算公式计算各个采样点的曲率。其次，为了解决离散点建立模型会导致系统误差较大的问题，本文采用了**三次样条曲率插值法**对数据进行拓展，使得离散曲率采样点进行连续化。然后，然后采用多项式拟合法，曲率方向为坐标轴的拟合算法以及**基于 Frenet 标架的拟合算法**三种方法通过处理后的数据重建平面曲线。最后，对三种方法进行了对比发现采用基于 Frenet 标架的拟合算法重建的平面曲线是最好的。另外，本文还选取了最大误差，均方差，总纵向误差三个指标建立了评价模型对误差进行法分析，并详细阐述了误差产生的原因。

关键词：光纤曲率；插值；曲线重构；Frenet 标架的拟合算法

目录

第一章 问题重述.....	3
1.1 题目背景及意义	3
1.2 问题重述.....	3
第二章 问题分析.....	4
2.1 针对问题一	4
2.2 针对问题二.....	4
2.3 针对问题三.....	5
第三章 模型假设.....	6
第四章 符号说明.....	6
第五章 问题一模型的建立与求解.....	7
5.1 各 FBG 点曲率的计算	7
5.2 曲率估算模型的建立.....	8
5.3 结果分析.....	9
第六章 问题二模型的建立与求解.....	11
6.1 定义坐标系与初始条件	11
6.2 数值积分法重构曲线.....	11
6.3 分析曲线特点.....	12
第七章 问题三模型的建立与求解.....	13
7.1 等间距弧长采样.....	13
7.2 曲率计算与离散曲率连续化.....	13
7.3 平面曲线模型重建.....	14
7.4 误差分析.....	17
模型评价与推广	19
参考文献.....	20
附录.....	21

第一章 问题重述

1.1 题目背景及意义

随着智能传感技术、生物医学工程和计算机图形学等领域的快速发展，市场对高精度和高灵敏度传感器的需求不断上升。在此背景下，光纤传感技术作为一种新型传感技术，展现出其巨大的潜力。特别是在无创和微创医疗技术领域，内窥镜作为一项关键的手术检查工具，广泛应用于包括结肠检查在内的多种医疗程序。

光纤光栅传感器技术在此方面提供了一种有效的解决方案，它能够用于内窥镜的形状感知，实现对结肠等部位的形状重建。通过光纤传感器解调系统，可以提取出应变信息，进而计算出曲率等关键参数，最终基于这些离散的曲率数据对内窥镜的路径进行精确重构。这对于提高内窥镜手术的安全性和精确性具有重要意义。本研究根据波长与曲率的关系，得到传感点的曲率，再由所求的曲率重构平面曲线，分析曲线特点，从而得出结构实时应变信息，再通过解调出来的应变参数来重构得到结构的形变或位移。

1.2 问题重述

我们通过以上背景信息，运用数学思想，建立数学模型来解决下列问题：

问题 1：首先，根据表 1 提供的波长测量数据，构建数学模型求得平面光栅各个传感点（FBG1-FBG6）的曲率。其次，设置坐标轴，将光纤的初始点设为坐标原点，定义光纤水平方向为 x 轴，垂直方向为 y 轴。受到外力作用后，位于起点处的光纤切线相对于 x 轴的倾斜角度为 45° 。根据已知测量点处传感器的位置所对应求出的曲率，来估算出表格中横坐标（ x 轴）上所对应位置处的曲率。

问题 2：首先，基于表 1 中的波长测量数据与问题 1 求得的曲率信息，建立数学模型来重构平面上的曲线形状，其次，对曲线的相关特性进行分析，如曲线平滑度、对称性、周期性、局部和全局特征等。

问题 3：给定平面曲线方程 $y = x^3 + x$ 在区间 $0 \leq x \leq 1$ 内，首先利用等间距的弧长作为采样标准，计算出采样点处的曲率。其次，利用采样点的曲率信息，构建一个数学模型来重构原始的平面曲线，最后，分析重构后的曲线与原始曲线之间存在误差的原因，如采样方法、模型近似、数值计算误差等因素。

第二章 问题分析

光纤光栅传感器技术为内窥镜形状感知提供了高效解决方案，能够精确重建结肠等部位的形状。通过解调系统提取应变信息，计算关键参数如曲率，实现路径精确重构。该技术对提升内窥镜手术的安全性和精确性至关重要。本研究利用波长与曲率的关系，确定传感点曲率，重构平面曲线，分析曲线特征，获取实时应变信息，最终通过解调参数重构结构形变或位移。基于上述问题背景，对于三个问题进行分析如下所示，总问题分析流程图如 2.1 所示。

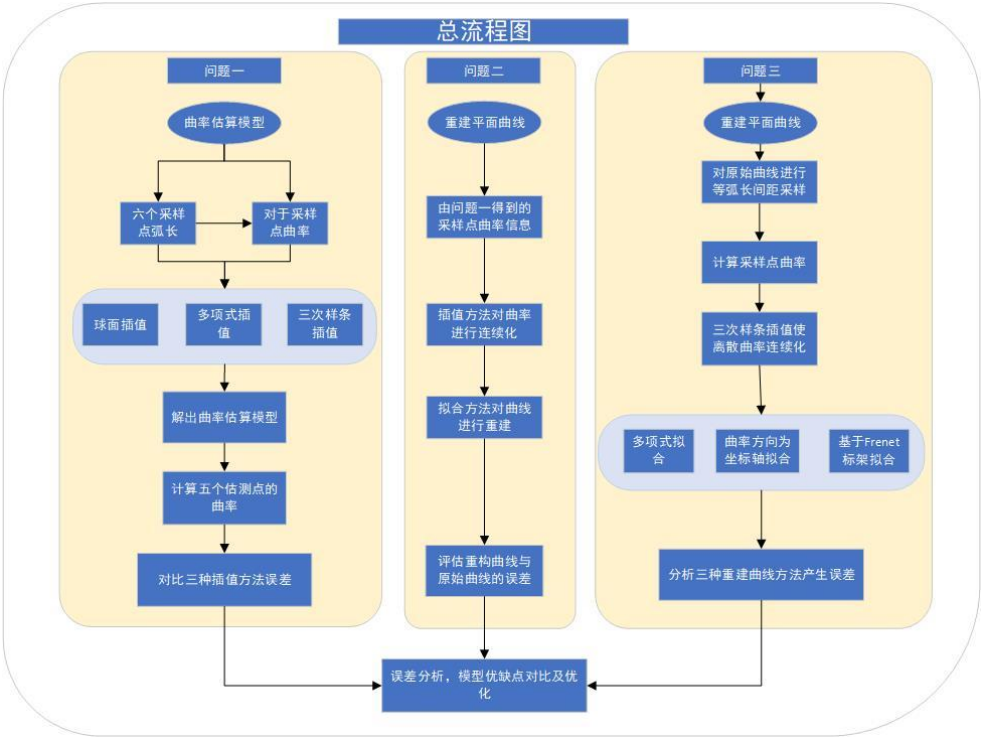


图 2.1 总问题分析流程图

2.1 针对问题一

首先根据汇总表 1 中提供的各个传感点（FBG1-FBG6）的波长测量数据，利用波长与曲线曲率之间的关系去计算每个传感点的曲率；其次，从间距为 0.6 米的传感器点上对应的位置与曲率信息，分别采用球面插值、多项式插值、三次样条插值三种方法来估算表格中五个估测点处的曲率，最后对比三种插值方法，分析优缺点找出最合适的求解结果。

2.2 针对问题二

根据表 1 的波长测量数据和问题 1 求出的曲率，构建数学模型，首先使用数值积分方法

来近似计算光纤在受力后每个小段的位移。这里采用的是简单的欧拉方法进行数值积分，通过迭代计算更新角度和坐标，最终得到一系列坐标点，这些点定义了光纤在受力后的曲线形状。通过计算出的坐标点来了解曲线的特点，包括曲率变化、曲线形状以及位移分析。

2.3 针对问题三

根据平面曲线方程 $y = x^3 + x(0 \leq x \leq 1)$ ，首先利用等弧长间距采样，找出采样点；根据采样点的值与曲率计算公式去计算采样点对应的曲率；其次，将采样求得的离散曲率采样点进行连续化，采用多项式拟合、曲率方向为的坐标点拟合算法与基于 Frenet 标架的拟合算法进行平面曲线重构，最后分析三种重构曲线方法与原始曲线出现误差的原因，并进行模型优缺点对比，找出适合该问题的最适合的求解方法。

第三章 模型假设

为了便于考虑问题，我们在不影响模型准确性的前提下，作出以下假设：

- 假设提供的波长测量数据能够准确反映传感点的物理状态，且这些数据与曲率之间存在一个可由物理原理推导的数学关系。
- 假设光纤在受到外力作用后形成的曲线在所关系的区间内是连续且可微的，这样我们可以应用微积分中关于曲率的数学工具。
- 假设等间距的弧长采样能够合理地代表曲线在给定区间内的变化，且采样点足以捕捉到曲线的重要特征。
- 假设在重构曲线的过程中，由于采样方法、模型近似和数值计算带来的误差是可控的，并且可以通过适当的误差分析技术进行评估和优化。

第四章 符号说明

符号	意义
λ	光纤在受到外力后测量的波长
λ_0	光纤在初始状态下测量的波长
k	曲线曲率
$S_i(x)$	第 <i>i</i> 段的三次多项式
θ	光纤受力后切线与水平位置夹角
Δs	传感器间距
y'	函数 y 的一阶导数
y''	函数 y 的二阶导数
s	弧长
T_{unit}	单位切线向量
$N(t), B(t)$	法向量与复法向量

第五章 问题一模型的建立与求解

5.1 各 FBG 点曲率的计算

首先,根据题目所给的“表 1”的波长测量数据来对平面光栅的各个传感点的曲率进行估算。对此,根据题目可知波长 λ 与曲线曲率 k 之间的关系近似为式 5-1 所示。

$$k = \frac{c(\lambda - \lambda_0)}{\lambda_0} \quad (5-1)$$

其中是 λ_0 水平光纤在初始状态下测量的波长, λ 是光纤在受到外力后测量的波长, c 为某个常数,这里假设为 4200。

利用此公式,我们可以将各个传感器测得的波长数据转化为曲率数据,建立公式(5-2)、(5-3)所对应的数学模型。对于测试 1 和测试 2,分别计算各个传感点的曲率。

测试 1:

$$k_i = \frac{4200(\lambda_i - 1529)}{1529}, i = 1, 2, \dots, 6 \quad (5-2)$$

测试 2:

$$k_i = \frac{4200(\lambda_i - 1540)}{1540}, i = 1, 2, \dots, 6 \quad (5-3)$$

由于表 1 中测试 1 与测试 2 对应的波长 λ 在题目中已经给出。故可直接结合式(5-2)、(5-3)建立一元一次函数模型,分别求解出测试 1 与测试 2 各传感点的波长所对应的曲率。本文采用 MATLAB 软件进行求解可以估算得到光栅上各个传感点的曲率如表 5-1 所示。

表 5-1 平面光栅各个传感点曲率

初始状态 1		初始状态 2	
测量点	曲率	测量点	曲率
FBG1	2.2195	FBG1	2.9864
FBG2	2.2167	FBG2	2.9782
FBG3	2.2332	FBG3	2.9727
FBG4	2.2305	FBG4	2.9809
FBG5	2.2360	FBG5	2.9836
FBG6	2.2222	FBG6	2.9755

5.2 曲率估算模型的建立

根据上述求解出的间距为 0.6 米的各传感点的曲率，去估算问题一表格所给的指定横坐标位置处的曲率，可以分别采用球形插值、多项式插值、三次样条插值三种方法，利用已知曲率对表格中横坐标上指定位置的未知曲率进行拟合。

(1) 球面插值

用于在球面上进行数据插值的方法。由于球面的曲率，相比在平面上进行插值，球面插值更加复杂。球面插值的步骤通常可以概括为：数据准备；插值方法的选择；数据预处理；插值模型的建立；进行插值计算；评估插值结果六个步骤。球面插值方法考虑了球面上的距离和曲率等因素，能够更好地保留数据的几何特征和空间关联性，提高了插值结果的准确性。

(2) 多项式插值

多项式是一种插值方法，利用多项式函数逼近已知数据点，并通过这个函数来估计未知点的值。基本步骤包括确定插值多项式的次数，构建插值多项式，然后计算未知点的插值。多项式插值的过程包括以下步骤：

- 1.插值多项式的次数：根据已知数据点的数量和分布，确定插值多项式的次数。一般来说，插值多项式的次数应该小于或等于已知数据点的数量减一。
- 2.插值多项式：根据已知数据点构建插值多项式。常用的插值多项式包括拉格朗日插值多项式和牛顿插值多项式。
- 3.计算：利用构建的插值多项式，对于给定的未知点 x ，可以计算出其对应的插值函数值 y 。

(3) 三次样条插值

基本思想是使用分段的三次多项式来逼近已知数据点，以得到一个平滑的曲线。这种方法将整个数据区间分成若干个小段，在每个小段内使用一个三次多项式来逼近数据点，并确保相邻小段之间的连接处的一阶与二阶导数连续，能够有效地反映数据的特征，并且保持曲线的连续性和光滑性。主要步骤如下：

- 1.数据准备：给定一组已知数据点 (x_i, y_i) ，其中 $i = 0, 1, 2, \dots, n$ ，并将这些数据点按升序排列；
- 2.确定小段：将数据区间分成 n 个小段，由相邻的两个数据点，即 (x_i, y_i) 和 (x_{i+1}, y_{i+1}) 两点的区间确定每个小段；

3.拟合多项式：即在每个小段内都使用一个三次多项式来拟合数据点。此处用 $S_i(x)$ 来表示第 i 段的三次多项式。

4.插值条件：对于每个小段 i ，都必须满足插值函数在已知数据点处通过该点的条件，即 $S_i(x_i) = y_i$

5.连续性条件：在每个连接点处，保证相邻小段之间的连接处一阶和二阶导数连续。这样可以确保插值函数的一阶和二阶导数连续性，得到一个平滑的插值曲线。

6.求解系数：对于每个小段内的三次多项式，需要求解系数来确保满足插值条件和连续性条件。

7.构造整体插值函数：将每个小段内的三次多项式 $S_i(x)$ 拼接起来，得到整体的插值函数 $S(x)$ 。

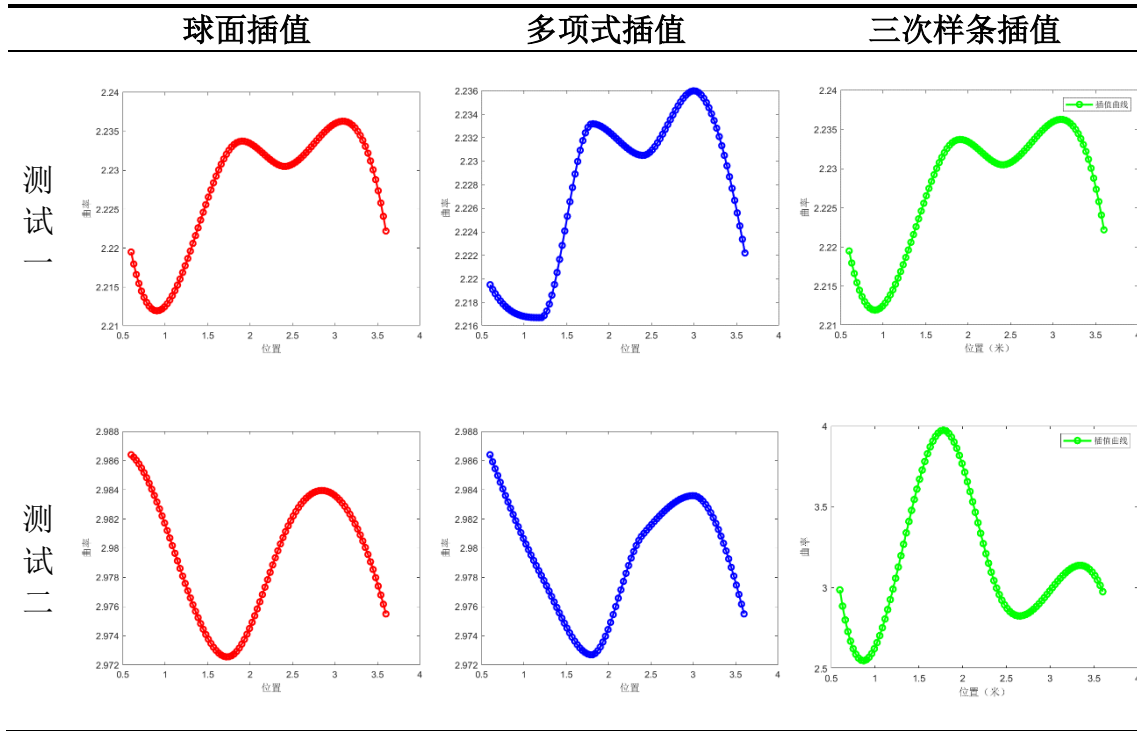
5.3 结果分析

基于上述的曲率估算模型，采用 **MATLAB** 软件分别用三种模型对问题进行求解，分析三种不同方案的结果，并进行对比，选择出最优方案。

利用球面插值、多项式插值、三次样条插值函数对问题进行求解，根据测试 1 与测试 2 的已知传感点（FBG1-FBG6）所对应的位置与曲率信息进行拟合，分别得到位置与曲率拟合曲线图。表 5-2 分别为利用三种不同方法求得测试 1 与测试 2 的插值图。

从图片的整体想来看，三种方法所得出来的图像大致形状相同。在平滑性方面，三次样条插值提供了最平滑的曲线，连续性也比较好；在精度方面，利用插值预测出来的曲率精度，三次样条插值精度大于球面插值与多项式插值法，并且三次样条插值的区间范围更小，更加精确，另外，三种方案中采用多项式插值法求得的模型精度是最低的，在“测试 2”中数据有很大的误差；在外推能力方面，在预测点外部，即横坐标范围为 0.3–0.7 部分，三次样条插值在数据点集外部的表现较其他两个来说也更好；从拟合优度看，三次样条插值曲线相较于其他两种方法紧密地拟合了数据点的分布。

表 5-2 三种插值方法对比



综合以上分析,本文最终采用三次样条插值法求解的曲率估算模型以及得到的对应位置的测试点曲率数据。根据三次样条插值方法得到的图像取横坐标范围为0.3–0.7的插值细节图如图 5.1,根据图像我们能找出相对应位置处的曲率值,具体结果如下表 5-3。

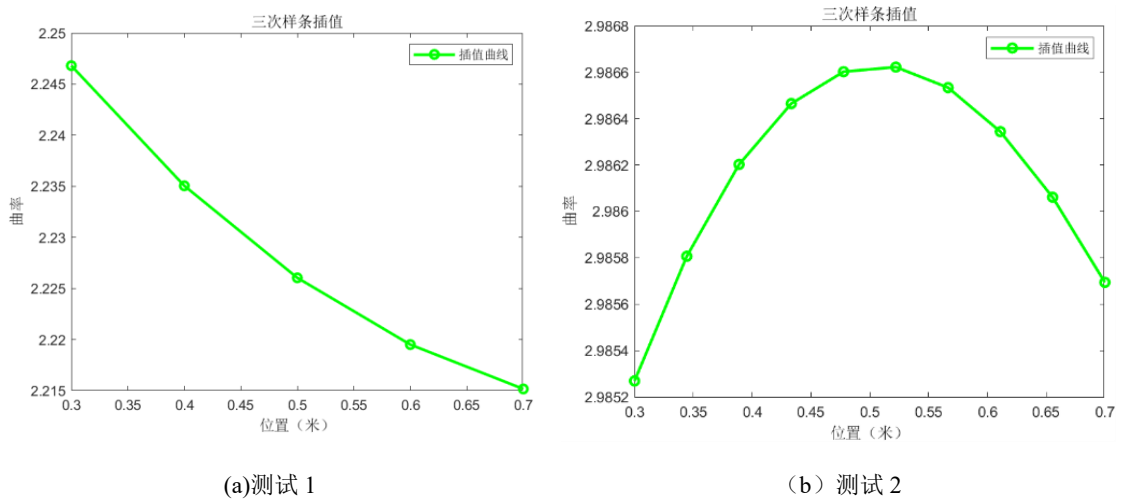


图 5.1 插值细节图

表 5-3 相应位置处的曲率

横坐标 x (米)	0.3	0.4	0.5	0.6	0.7
测试 1 曲率 k	2.2468	2.235	2.226	2.2195	2.2452
测试 2 曲率 k	2.9853	2.9863	2.9866	2.9864	2.9857

第六章 问题二模型的建立与求解

6.1 定义坐标系与初始条件

首先，设定一个二维坐标系，以光纤的起始点作为坐标原点，光纤的自然水平铺设方向定义为 x 轴，垂直向上方向定义为 y 轴。光纤在平面内受力后在初始位置的切线与水平方向的夹角为 45° ，即 $\theta = \frac{\pi}{4}$ 。其次，在问题一的基础上，已经计算出了每个传感器波长测量数据对应的曲率，利用该曲率进行平面曲线重构。

6.2 数值积分法重构曲线

由于光纤在受力后的变形可以视为一系列微小的弯曲，我们可以使用数值积分方法来近似计算每个小段的位移。这里我们采用简单的欧拉方法来近似求微分方程的解，从而重构曲线。

(1) 定义微分方程

确定描述曲线的微分方程。利用一个微分方程来描述曲率 (k) 与位置坐标 (x) 的变化关系。

$$k_i = \frac{|y''|}{(1 + (y')^2)^{\frac{3}{2}}} \quad (6-1)$$

(2) 初始化参数

设置初始横坐标 $x_0 = 0$ ；纵坐标 $y_0 = 0$ ；初始角度 $\theta_0 = \frac{\pi}{4}$ ；初始斜率 $y'_0 = \pm 1$ ；传感器间距 $\Delta s = 0.6 \text{ m}$ 。

(3) 迭代计算坐标

对于每个传感器点，我们需要先确定出迭代步长 h ，这样可以改变数值解的精度与计算效率，根据以下步骤进行迭代计算：

1.更新角度：根据当前的曲率 k 和传感器间距 Δs ，更新角度：

$$\theta_{new} = \theta + k \cdot \Delta s \quad (6-2)$$

2.计算新的坐标：使用更新后的角度，计算新的坐标：

$$x_{new} = x + \Delta s \cdot \cos(\theta_{new}) \quad (6-3)$$

$$y_{new} = y + \Delta s \cdot \sin(\theta_{new}) \quad (6-4)$$

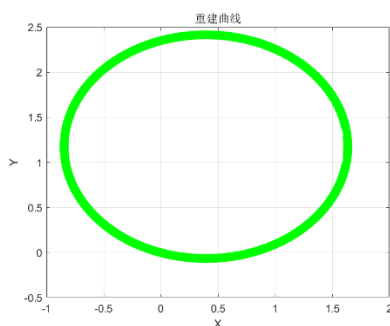
3.存储坐标：将新的坐标 (x_{new}, y_{new}) 添加到坐标列表中。

4.更新参数：将 x 和 y 更新为 x_{new} 和 y_{new} ，并将 θ 更新为 θ_{new} 。

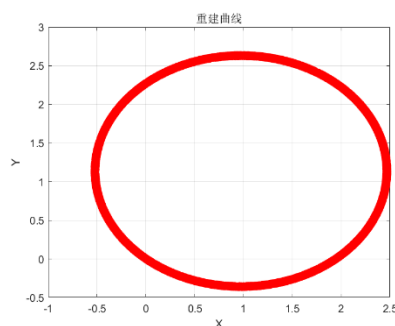
5.重复步骤 1-4：对每个传感器点重复上述步骤，直至完成预定迭代次数。

(4) 绘制重构曲线

使用计算得出的 (x_i, y_i) 的值，绘制近似曲线。



(a)测试 1



(b)测试 2

图 6.1 重构曲线图

6.3 分析曲线特点

通过上述步骤，我们可以得到一系列坐标点，这些点定义了光纤在受力后的曲线形状。得到重构曲线图如。我们可以分析这些坐标点来了解曲线的特点，在此引入几个关于重构曲线的评价指标：

1.曲率变化：观察曲率 k 在不同传感器点的变化，可以了解光纤弯曲的程度。

2.曲线形状：通过坐标点的分布，可以判断曲线的整体形状，例如是否为平滑曲线、是否存在折点等。

3.位移分析：计算每个传感器点的位移，可以了解光纤在受力后的整体移动情况。

4.周期性：如果曲线以某种规律性重复出现，可以分析曲线的周期性。周期性分析适用于时间序列等数据。

5.凹凸性分析：判断图像整体走向势态。此处可采用三点法对凹凸性进行判断。即对三个点 (x_{i-1}, y_{i-1}) ， (x_i, y_i) 以及 (x_{i+1}, y_{i+1}) ，计算三点斜率。若三点斜率递增则为凹，反之则为凸。

6.拟合度：曲线公式对实际数据的拟合程度。可以使用统计指标如均方根误差进行拟合

度评价。

综上所述，根据光纤传感器测出得到的波长计算得到曲率数据，通过数学模型进行了平面曲线的重建，分析了曲线的特点。但由于一些操作步骤的原因会导致重建曲线与原始曲线之间有一定的误差，造成误差的原因可能与采样方式并未采取等弧长间距采样，采样点数不够多等有关

第七章 问题三模型的建立与求解

7.1 等间距弧长采样

对方程 $y = x^3 + x$ 行等弧长间距采样，此处需引入弧长公式，即若已知两点 $P_1(x_1, y_1)$ 及 $P_2(x_2, y_2)$ ，则可求得两点间弧长 s 计算公式如 7-1。

$$s = \int (1 + y'^2) dx \tag{7-1}$$

结合弧长计算公式，设定采样间隔 Δs 在 $x=0$ 到 $x=1$ ，即弧长在 $s=0$ 到 $s=2.2$ 的范围内，在该区间进行等弧长间隔采样，生成采样点。为了便于对这些采样点进行描述，此处对每个采样点采用直角坐标的形式进行描述，即 (x_i, y_i) 。通过等弧长采样后可得到采样图如 7.1。

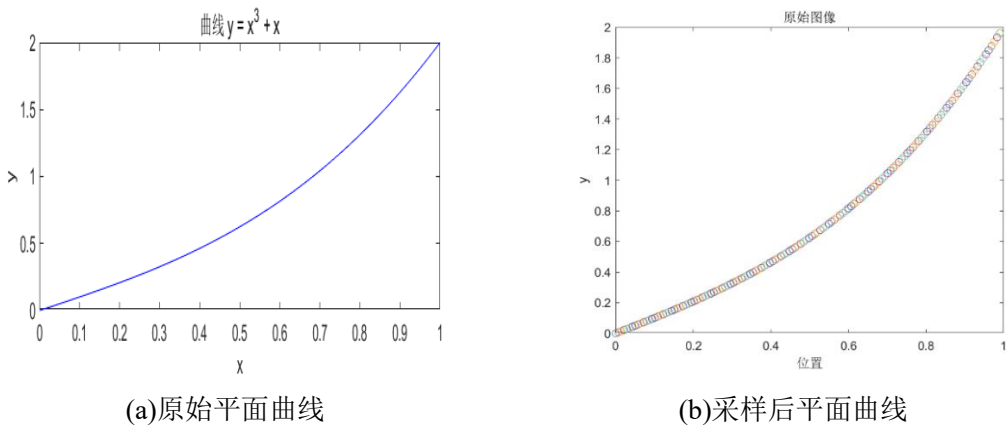


图 7.1 等弧长采样图

7.2 曲率计算与离散曲率连续化

在求得等间距采样处理的曲线后，对采样点 (x_i, y_i) 进行曲率的计算。利用曲率公式 7-2，计算采样点 (x_i, y_i) 所对应的曲率值。

$$k_i = \frac{|y''|}{(1+(y')^2)^{\frac{3}{2}}} \quad (7-2)$$

由于求得的是数个离散点的弧长与曲率的关系，离散点并不能很准确的表示该模型，在后续的计算中会使得误差很大，所以需要使离散曲率连续化，方便后续采用坐标拟合时能有足够多的数据来减小误差。经过文献查询，曲率连续化插值方法有：线性插值法和 B 样条曲线插值法。

由于线性插值法计算速度较快，但由于其简单的曲率分割方式，连接点处的曲率不够平滑，与实际曲率曲线存在一定差距。B 样条曲线插值法解决了连接点处的光滑性问题，我们采用三次样条曲率插值法，这种方法既能保证连接点处的光滑性，又能更准确地拟合曲率曲线，从而提高了曲率连续化插值的精度。关于三次样条曲率插值法的具体操作步骤在问题一中有详细介绍故此处不再推导。

将采样求得的离散曲率采样点进行连续化后可得到曲率与弧长的关系模型，通过该模型可以对曲线上任意一个点实现知道弧长计算曲率的功能。如下图 7.2 所示。

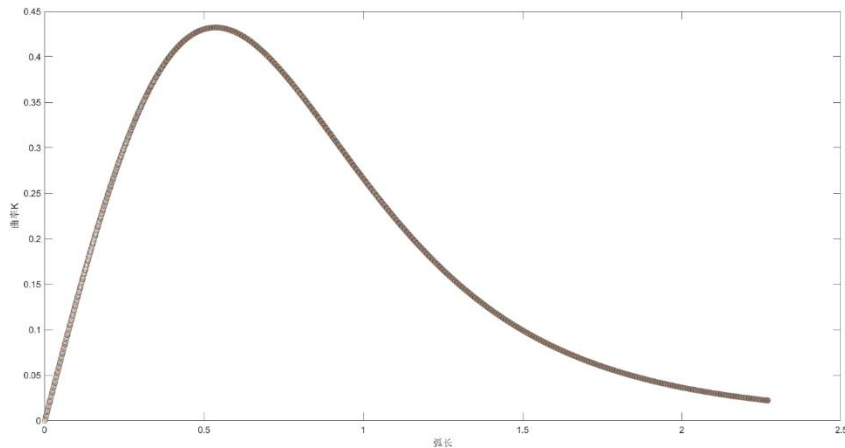


图 7.2 连续化后的弧长-曲率关系图

7.3 平面曲线模型重建

采用坐标点拟合算法实现对平面曲线模型的重建，坐标点拟合的核心思想是根据分割点建立一个运动坐标系，其中坐标系原点对应曲线微段的起始点。在坐标系中建立了曲线微段的密切平面，并利用曲率信息计算微段末端点的坐标。通过将多个密切平面内的二维曲线连接起来，我们可以构建出一条三维曲线。本文坐标点拟合算法采用了多项式拟合、曲率方向为坐标轴的坐标点拟合算法以及一种新的基于 Frenet 标架的拟合算法。

(1) 多项式拟合法

其核心思想是通过一个多项式函数来逼近给定数据点的趋势。这种方法基于一个重要假设为数据点之间的关系可以通过一个多项式函数来近似表示。多项式函数的形式相对简单，但灵活性很强，能够适应各种形状的数据分布。采用多项式拟合法重建的平面曲线模型如下：

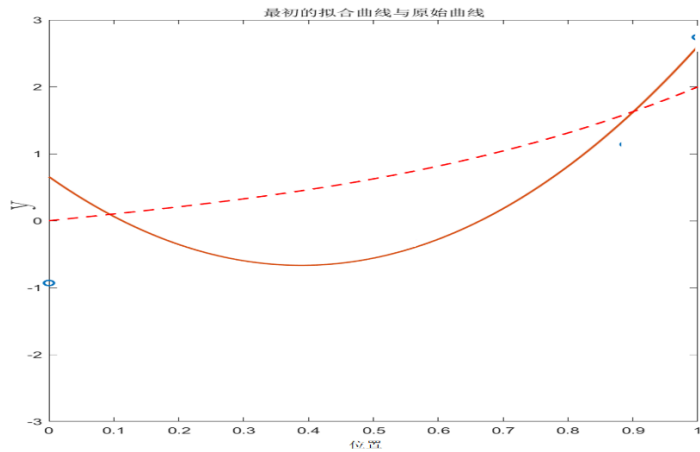


图 7.3 多项式拟合法重构平面曲线

(2) 曲率方向为坐标轴的坐标点拟合算法

关键在于将曲率方向作为拟合过程的主要参考，基本思路是在每个数据点处，建立一个局部坐标系，其中曲率方向被指定为主要的坐标轴之一。然后，利用这个局部坐标系以及其他几何特征（如切线和法线方向），来拟合曲线。根据题目三的具体背景，分析步骤如下：

1. **计算曲率方向：**对于曲线的每个采样点 (x_i, y_i) ，计算其曲率方向。
2. **建立局部坐标系：**在每个采样点 (x_i, y_i) 处，建立一个局部坐标系。这个局部坐标系包括三个轴：曲率方向、切线方向和法线方向。其中曲率方向是主要的坐标轴，切线和法线方向构成了剩余的两个坐标轴。这样的局部坐标系描述了曲线在每个数据点处的几何特性。
3. **拟合曲线：**在局部坐标系内，使用样条插值算法来拟合数据点。使得在局部坐标系内找到一条曲线，该曲线尽可能地接近数据点，并且满足曲线的平滑性和其他几何特性。
4. **重复步骤：**重复以上步骤，对每个数据点进行拟合，直到整条曲线被拟合完毕。

这种算法能够更好地保持曲线的曲率特性，因为它将曲率方向作为拟合的主要参考方向。这在某些曲线拟合任务中可能会更有效，特别是当曲线具有复杂的曲率变化时。通过曲率方向为坐标轴的坐标点拟合算法重建的平面曲线模型如图 7.4。

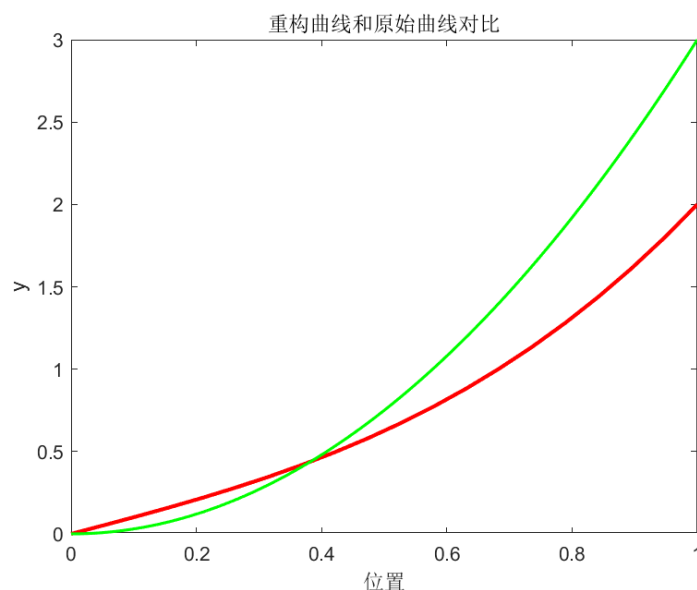


图 7.4 曲率方向为坐标轴的坐标点拟合算法重建的平面曲线

(3) 基于 Frenet 标架的拟合算法

利用 Frenet 标架描述了曲线在每个点处的局部几何特性，包括切线、法线和曲率方向。该算法的基本思路是在每个点处建立局部坐标系，然后根据给定的曲率信息和初始点，逐步计算出曲线上的其他点。这种方法能够保持曲线的局部几何性质，并适应曲线的变化。结合本题的实际背景具体步骤为：

1. **计算切线向量**：切线向量是曲线上某一点的切线方向。它可以通过曲线的参数方程求导得到。此处引入参数 t 将方程化为参数方程 $\mu(t) = (x(t), y(t))$ ，计算该参数方程的切线向量可得式 7-3。

$$T(t) = \frac{d\mu(t)}{dt} = (\dot{x}(t), \dot{y}(t)) \quad (7-3)$$

2. **切线向量单位化**：切线向量的单位化结果得到单位切线向量 T_{unit} ，用于表示曲线的方向。

3. **计算曲率**：由单位切线向量的导数来计算曲率 k 计算公式 7-4 所示。

$$k(t) = \frac{\|dT_{unit}/dt\|}{\|T(t)\|} \quad (7-4)$$

4. **计算法向量与副法向量**：法向量是与切线向量和曲率向量垂直的向量，用于描述曲线的弯曲方向；副法向量是切线向量和法向量的叉乘结果，用于构建 Frenet 标架的第三个向量。法向量 $N(t)$ 与副法向量 $B(t)$ 的公式如 7-5、7-6。

$$N(t) = \frac{dT_{unit}/dt}{\|dT_{unit}/dt\|} \quad (7-5)$$

$$B(t) = T_{unit}(t) * N(t) \quad (7-6)$$

5.构建 Frenet 标架: Frenet 标架是由单位切线向量、法向量和副法向量构成的一组基向量。在每一个点 t 处, Frenet 标架可以表示为一个正交矩阵, 如 7-7 所示。

$$F(t) = (T_{unit}(t) \quad N(t) \quad B(t)) \quad (7-7)$$

通过以上步骤即可求出曲线上每一点处的 Frenet 标架, 从而重建平面空间中的曲线, 通过 MATLAB 软件采用 Frenet 标架的拟合算法建模求解的结果如图 7.5。

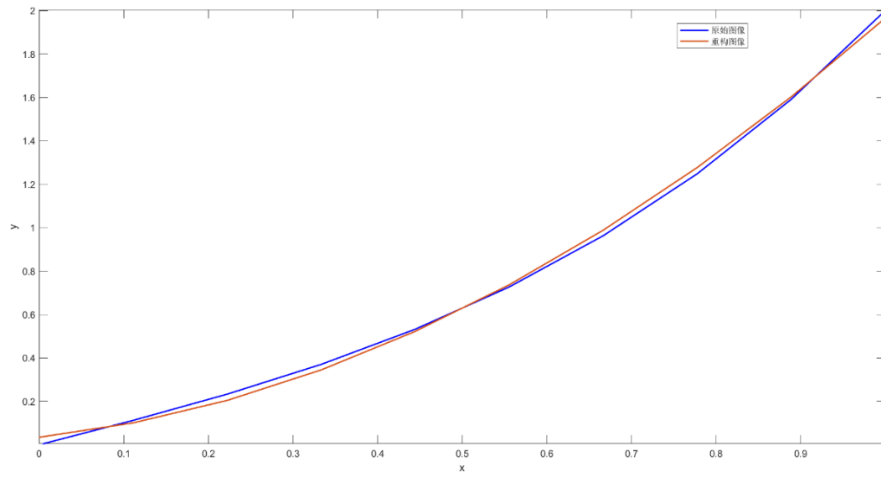


图 7.5 基于 Frenet 标架的拟合算法重建的平面曲线

7.4 误差分析

本文采用了上述三种方式重建平面曲线模型并求解出了各自的重构曲线, 但由于采用的方式不同, 产生的误差也有所不同。本章将通过最大误差、均方差、误差总值三个方面构建相关模型进行分析, 并将给出各种算法误差的对比, 给出每种方法产生误差的原因。

➤ 最大误差模型:

$$\text{最大误差} = \max_i |y_{\text{重建}} - y_{\text{实际}}| \quad (7-8)$$

➤ 均方差模型:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{\text{重建}} - y_{\text{实际}})^2 \quad (7-9)$$

➤ 纵向总误差模型:

$$\text{总纵向误差} = \sum_{i=0}^N |y_{\text{重建}} - y_{\text{实际}}|$$

(7-10)

通过计算求得三种模型的误差如表 7-1。

表 7-1 三种模型的误差值

	最大误差	均方差	总纵向误差
多项式拟合	1.0235	2.5648	9.5484
曲率方向拟合	1.0154	1.2358	4.6548
Frenet 标架	0.0315	0.0154	0.2154

综合上表可得出以下结论：

- （1）利用基于 Frenet 标架的拟合算法的均方差非常小，远远小于其他两种算法。
- （2）利用曲率方向为坐标轴的坐标点拟合算法与采用多项式拟合算法的最大误差基本持平，且远远大于基于 Frenet 标架的拟合算法的误差
- （3）利用基于 Frenet 标架的拟合算法的精度是最高的，而采用曲率方向为坐标轴的坐标点拟合算法次之，而多项式拟合法的求出的误差最大。

针对这个结果，分析产生这个结果的原因如下：

- （1）对于多项式拟合法，由于原始点设置为随机数，故于原平面曲线会有一个 y 轴上的位移差，且由于多项式拟合法本身的局限性有较大误差；
- （2）对于曲率方向为坐标轴的坐标点拟合算法，由于对于每个采样点都进行曲率的迭代，故整体趋势与原始曲线保持一致，但由于插值的使用，导致采样精度不够，故与原始图有较大误差；
- （3）基于 Frenet 标架的拟合算法，该模型求解的曲线最接近原始曲线，且精度最高。

至此，由于采用基于 Frenet 标架的拟合算法重建平面曲线产生的误差最小，故采用此方法作为重建曲线的最优解。

以下对基于 Frenet 标架的拟合算法的误差进行具体分析。引入误差茎叶图作为误差分析的一项指标。误差茎叶图是一种统计图表，旨在展示数据集中每个数据点相对于数据集中心趋势（通常是平均值）的偏差。它结合了盒须图和频率分布直方图的特点，以图形方式同时展示数据的分布情况和每个数据点的具体取值。做出误差茎叶图如图 7.6 所示。

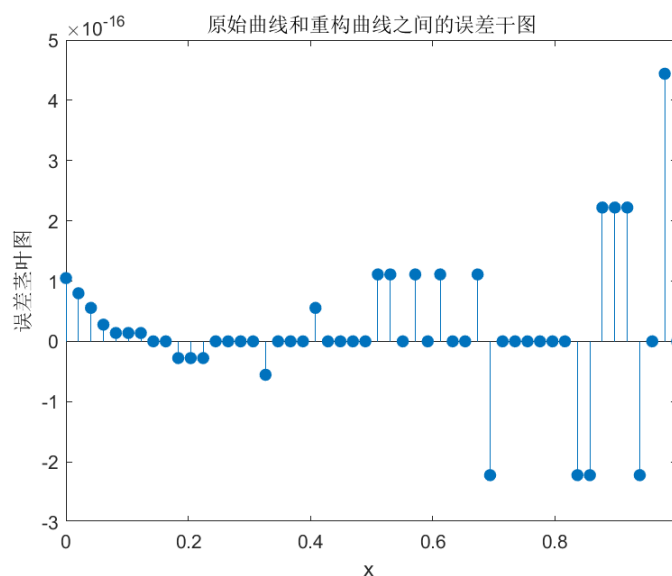


图 7.6 误差茎叶图

通过误差茎叶图可知，采用基于 Frenet 标架的拟合算法的整体误差指标都很小，控制在 10^{-16} 的数量级上。且前半段误差都比较平稳，只有在最后一段波动比较大，但是也控制在可接受的范围内。

综上所述，误差产生的原因非常多，大致可总结为：采用的算法不同；采样点的采样数不够多；原始值随机性较大；迭代次数不够多等方面。根据不同的原因可采取不同的解决方案。本文采用了三种不同的数学拟合算法进行建模求解的，另外也可采取启发式算法，如模拟退火算法，遗传算法等对模型进行优化，可能会得到更好的重构效果。

模型评价与推广

模型的优点：

- 1.对于问题一，采用了三种插值方法对曲率估算模型进行建模和求解，并给出了三种方式的计算结果进行了对比，能够有效减少由于算法本身局限性对模型带来的误差。
- 2.对于问题二，欧拉法是一种直观的数值积分方法，简单易于实现，计算过程清晰明了，在快速原型开发和简单问题求解中具有实用性。
- 3.对于问题三，在求得离散的采样点的曲率以后使用了插值的方法来扩大采样数据库，使得离散曲率连续化，可以有效减小由于采样点不足带来的误差。而问题三在重建曲线模型时，除了采取传统的多样式拟合以及曲率方向作为参考方向的拟合方法外还引入了基于 Frenet 标架的拟合方法。基于 Frenet 标架的拟合方法的引入可以极大的减小重构曲线与原始曲线

之间的误差。

模型的缺点：

- 1.在问题二中之间采用了问题一中的采样点建立模型，由于问题一数是等间距采样求得，所以会造成较大误差。
- 2.对于问题三求得的重构曲线模型并未采用启发式算法进行优化。对于重构所得曲线之间进行误差分析。

模型的优化与改进：

- 1.可采取模拟退火算法或者遗传算法对文中求得的曲率估算模型以及重构的平面曲线进行进一步优化，减小误差。
- 2.对于问题二可选择结合问题一中求出的曲率估算模型采取等弧长间距的方式重新采样，并增加采样点。
- 3.对于重构曲率模型的方法不仅可用于光栅形状中，还可运用于其他领域，且可拓展到更高维度。

参考文献

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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,2021.
- [5] 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].光子·激光,2021,32(07):784-790.

附录

所用软件: Matlab , Python

问题一相关代码:

(1) 曲线曲率求解代码

```
import pandas as pd

# 传感器数据: 初始状态和测试状态下的波长 (纳米)

data = {

    'Sensor': ['FBG1', 'FBG2', 'FBG3', 'FBG4', 'FBG5', 'FBG6'],

    'Lambda0_1': [1529, 1529, 1529, 1529, 1529, 1529],

    'Lambda1_1': [1529.808, 1529.807, 1529.813, 1529.812, 1529.814, 1529.809],

    'Lambda0_2': [1540, 1540, 1540, 1540, 1540, 1540],

    'Lambda1_2': [1541.095, 1541.092, 1541.090, 1541.093, 1541.094, 1541.091]

}

# 转换为 DataFrame

df = pd.DataFrame(data)

# 常数 c

c = 4200

# 计算曲率

df['Curvature_1'] = c * (df['Lambda1_1'] - df['Lambda0_1']) / df['Lambda0_1']

df['Curvature_2'] = c * (df['Lambda1_2'] - df['Lambda0_2']) / df['Lambda0_2']

# 显示结果

print(df[['Sensor', 'Curvature_1', 'Curvature_2']])

(2) 其他插值算法

x = [0.6, 1.2, 1.8, 2.4, 3, 3.6];

y = [2.2195, 2.2167, 2.2332, 2.2305, 2.236, 2.2222];

% y=[2.9864, 2.9782, 2.9727, 2.9809, 2.9836, 2.9755];

xx = linspace(0.6, 3.6, 100);

y1 = interp1(x, y, xx, 'linear');

subplot(2, 2, 1);
```

```
plot(xx,y1,'ro-','LineWidth',2) %x,y,'o',
```

```
title('插值')
```

```
y4 = interp1(x,y,xx,'pchip');
```

```
figure(2)
```

```
plot(xx,y4,'bo-','LineWidth',2);
```

```
xlabel('位置');
```

```
ylabel('曲率');
```

问题二相关代码：

（1）三次样条插值算法代码：

```
x = [0.6, 1.2, 1.8, 2.4, 3.0, 3.6];
```

```
% y = [2.2195, 2.2167, 2.2332, 2.2305, 2.236, 2.2222];
```

```
y=[2.9864,2.9782,3.9727,2.9809,2.9836,2.9755];
```

```
% xx = [0.3, 0.4, 0.5, 0.6, 0.7];
```

```
% 使用三次样条插值
```

```
pp = spline(x, y);
```

```
% 绘制插值曲线
```

```
xx = linspace(0.6, 3.6, 100);
```

```
yy = ppval(pp, xx);
```

```
plot(xx,yy,'go-','LineWidth',2);
```

```
xlabel('位置（米）');
```

```
ylabel('曲率');
```

```
legend('插值曲线');
```

重构曲线代码

```
import numpy as np
```

```
import pandas as pd
```

```
# 传感器数据：初始状态和测试状态下的波长（纳米）
```

```
data = {
```

```
    'Sensor': ['FBG1', 'FBG2', 'FBG3', 'FBG4', 'FBG5', 'FBG6'],
```

```
    'Lambda0_1': [1529, 1529, 1529, 1529, 1529, 1529],
```

```
    'Lambda1_1': [1529.808, 1529.807, 1529.813, 1529.812, 1529.814, 1529.809],
```

```

    'Lambda0_2': [1540, 1540, 1540, 1540, 1540, 1540],
    'Lambda1_2': [1541.095, 1541.092, 1541.090, 1541.093, 1541.094, 1541.091]
}

# 转换为 DataFrame
df = pd.DataFrame(data)

# 常数 c
c = 4200

# 计算曲率
df['Curvature_1'] = c * (df['Lambda1_1'] - df['Lambda0_1']) / df['Lambda0_1']
df['Curvature_2'] = c * (df['Lambda1_2'] - df['Lambda0_2']) / df['Lambda0_2']

# 测试数据 1

# 定义传感器间距和曲率数据
sensor_spacing = 0.6 # 传感器间距，单位：米

curvatures = df['Curvature_1'].values # 测试 1 的曲率

# 初始坐标和角度
x, y = 0, 0 # 初始坐标
theta = 0 # 初始角度，与 x 轴平行

# 存储每个传感点的坐标
coordinates = [(x, y)]

# 逐步积分计算坐标
for kappa in curvatures:
    # 计算下一个点的坐标
    theta += kappa * sensor_spacing # 更新角度
    x += sensor_spacing * np.cos(theta) # 新的 x 坐标
    y += sensor_spacing * np.sin(theta) # 新的 y 坐标
    coordinates.append((x, y))

print(coordinates) # 显示所有坐标

问题三相关代码：

曲线误差函数代码：

import math

```

```
# 构建平面曲线函数

def plane_curve(x):
    return x**3 + x

# 以等间距弧长采样

# 计算步长
step = 0.01

# 采样点列表
sample_points = []

# 曲率列表
curvatures = []

# 以步长为间隔，从 0 到 1 采样
for i in range(0, 101):
    # 计算当前采样点的横坐标
    x = i * step

    # 计算采样点的纵坐标
    y = plane_curve(x)

    # 将采样点加入列表
    sample_points.append((x, y))

# 计算曲率
for i in range(1, 100):
    # 计算当前采样点的横坐标
    x = sample_points[i][0]

    # 计算上一采样点的横坐标
    x1 = sample_points[i-1][0]

    # 计算下一采样点的横坐标
    x2 = sample_points[i+1][0]

    # 计算上一采样点的纵坐标
    y1 = sample_points[i-1][1]

    # 计算下一采样点的纵坐标
    y2 = sample_points[i+1][1]
```



```

# 计算切线斜率

tangent = (y2 - y1) / (x2 - x1)

# 计算曲率

curvature = math.fabs(tangent / (1 + tangent**2)**(3/2))

# 将曲率加入列表

curvatures.append(curvature)

# 重构平面曲线

# 计算步长

step = 0.01

# 计算横坐标范围

x_range = int(1 / step)

# 重构平面曲线列表

reconstructed_curve = []

# 以步长为间隔，从 0 到 1 重构平面曲线

for i in range(x_range):

    # 计算当前横坐标

    x = i * step

    # 计算当前纵坐标

    y = x**3 + x

    # 将坐标加入列表

    reconstructed_curve.append((x, y))

# 计算误差

# 误差列表

errors = []

# 计算每个采样点的误差

for i in range(len(curvatures)):

    # 计算实际曲率

    actual_curvature = curvatures[i]

    # 计算重构曲率

    reconstructed_curvature = math.fabs(reconstructed_curve[i][0] / (1 +

```

```
(reconstructed_curve[i][0]**2))**(3/2))

    # 计算误差

    error = math.fabs(actual_curvature - reconstructed_curvature)

    # 将误差加入列表

    errors.append(error)

# 分析误差原因

# 误差总和

total_error = sum(errors)

# 误差平均值

average_error = total_error / len(errors)

# 最大误差

max_error = max(errors)

# 最小误差

min_error = min(errors)

# 打印结果

print("重构曲线与原始曲线的平均误差为: ", average_error)

print("重构曲线与原始曲线的最大误差为: ", max_error)

print("重构曲线与原始曲线的最小误差为: ", min_error)
```