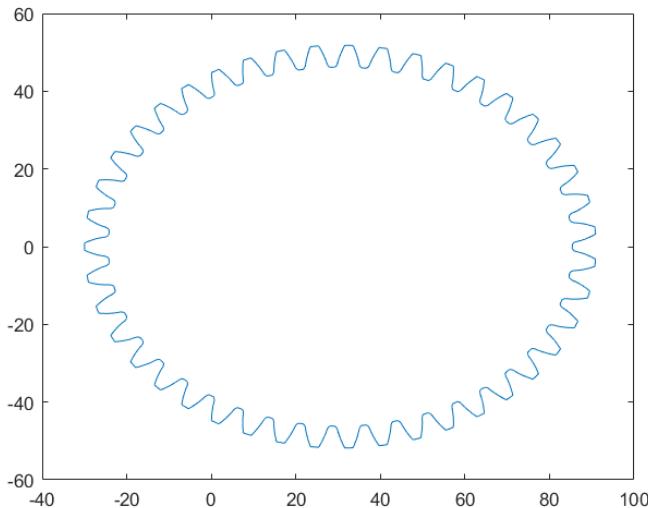


作业二 路径的特征分段和处理

25S065010 吴俊达

一、绘制图形



二、分析讨论如何对这个路径进行曲线插值和近似

首先使用分区域算法（如下节所描述），将路径分为直线、曲线和急转弯。然后，对于每种不同类型的路径点，采用不同的类型进行插值和（或）近似：

1. 长直线段（类型 1）, 直接使用直线方程 $y = \text{slope} \cdot x + \text{intercept}$ 表示，不需要额外的插值处理。
优势在于计算简单，存储高效，渲染快速

2. 曲线段（类型 2）, 比较平滑，可使用曲线插值或近似方法。比如使用 p 次 B 样条曲线进行拟合，当然也可以直接利用多项式基函数（幂级数）来进行拟合；如果点数较多，考虑计算效率，可使用最小二乘曲线拟合（LSCA）算法。

3. 急转段（类型 3）, 采用特殊处理策略。可以采用类似“Blending”的策略，使用带有切线约束的样条（紧压样条，clamped spline），确保过渡自然，避免生硬的转角。

三、完成基础的分区域算法

见压缩包下 hw2.m。

算法流程用自然语言叙述如下：

第一步：寻找尖锐转角

- 算法沿着路径逐点前进，每次观察相邻的三个点
- 计算这三个点形成的两个向量之间的夹角
- 如果夹角超过 45 度（比如一个急转弯），就标记为“尖锐转角”
- 这些尖锐转角点自然地将路径分割成不同的区域

第二步：分析各段路径性质

对于两个尖锐转角之间的每一段路径：

- 算法尝试用直线来拟合这段路径
- 计算路径上的点到拟合直线的最大垂直距离（弦差）
- 同时测量这段路径的直线长度

第三步：分类决策

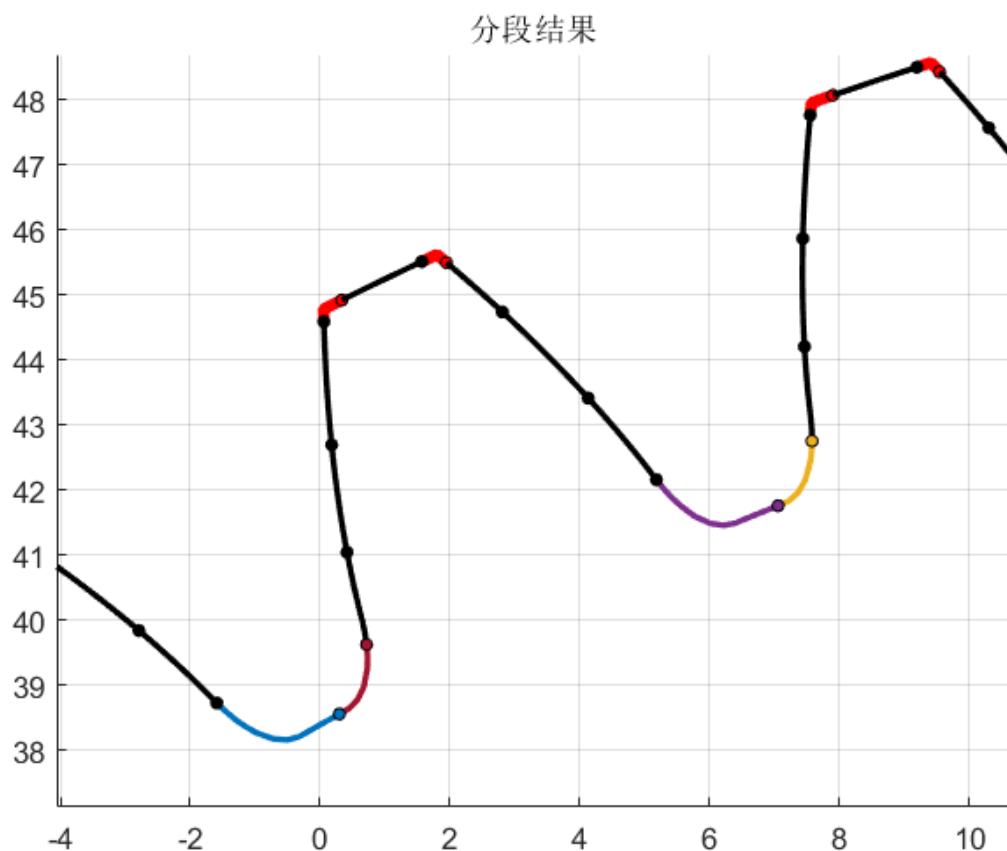
基于上述分析：

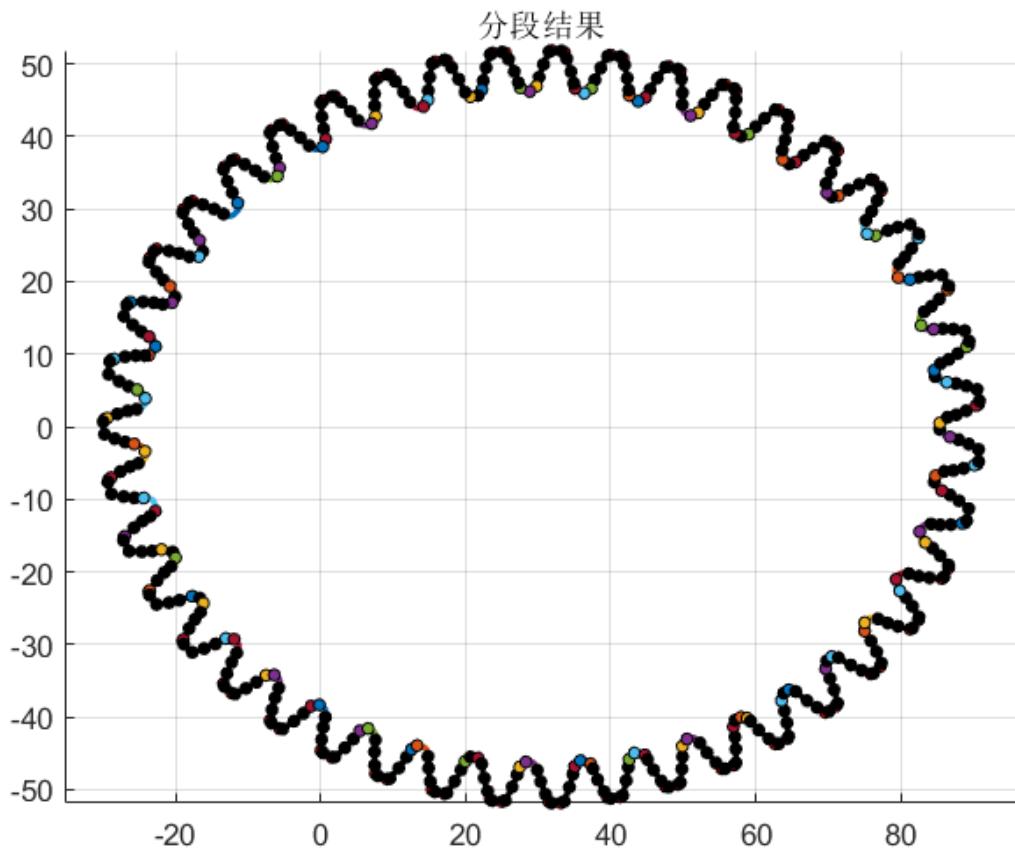
- **长直线段：**如果弦差很小 (<0.02) 且长度足够长 (>1.0)，就判定为直线
- **曲线段：**如果弦差较大，或者长度较短，就判定为曲线
- **急转段：**尖锐转角本身被单独标记

第四步：优化处理

- 合并过短的曲线段，避免分段过于零碎
- 确保每个曲线段有足够的点来进行后续处理

分区结果如下面两张图所示。图中，黑色线是被标记为长直线的段落，红色粗线是尖锐转角，彩色线则是普通的曲线。





值得注意的是，该算法还有可改进的余地。比如，对于每个重复单元，其划分的结果并不总是一样的，这可能是因为算法对每个重复单元之间微小的差异比较敏感（比如说，有的单元被识别成直线的区域，在另一个单元的同个区域中由于路径点偏离稍大，就被截断成曲线了）。对于这个问题，本人尚在思考中。