

实验报告：交互式 Bézier 曲线编辑

计算机辅助几何设计第二次作业

15 刘行

PB22000150

2025 年 10 月 2 日

目录

1	引言	2
1.1	背景	2
1.2	实验目标	2
2	算法描述	3
2.1	de Casteljau 算法	3
2.2	Bernstein 基函数方法	3
3	程序说明	4
3.1	编程平台	4
3.2	程序结构	4
3.3	程序使用说明	4
4	实验结果	5
4.1	基本结果	5
4.2	不同阶数曲线	5
4.3	算法精度验证	6
5	讨论与分析	7
5.1	算法比较	7
5.1.1	de Casteljau 算法的优势	7
5.1.2	Bernstein 基函数方法的优势	7
5.2	交互设计考虑	7
6	结论	8

Chapter 1

引言

1.1 背景

Bézier 曲线是计算机图形学和计算机辅助设计中最重要曲线表示方法之一, 由法国工程师 Pierre Bézier 在 1960 年代提出. 它通过控制多边形来定义光滑曲线, 广泛应用于汽车设计, 动画制作和字体设计等领域.

Bézier 曲线的两种主要计算方法包括: 递归的 de Casteljau 算法和基于 Bernstein 基函数的代数方法. 理解这两种算法的原理和实现对于掌握计算机辅助几何设计至关重要.

1.2 实验目标

- 实现 Bézier 曲线的两种计算算法: de Casteljau 算法和 Bernstein 基函数方法
- 开发交互式界面, 通过控制多边形编辑曲线
- 比较两种算法的计算效果和数值精度
- 验证两种算法在理论上的等价性

Chapter 2

算法描述

2.1 de Casteljau 算法

de Casteljau 算法是一种递归的几何构造方法, 通过线性插值逐步构建曲线. 对于 n 个控制点 P_0, P_1, \dots, P_n 和参数 $t \in [0, 1]$, 算法定义如下:

$$P_i^0(t) = P_i, \quad i = 0, 1, \dots, n$$

$$P_i^r(t) = (1-t)P_i^{r-1}(t) + tP_{i+1}^{r-1}(t), \quad r = 1, 2, \dots, n, \quad i = 0, 1, \dots, n-r$$

最终点 $P_0^n(t)$ 即为曲线上对应于参数 t 的点.

2.2 Bernstein 基函数方法

Bernstein 基函数方法使用代数公式直接计算曲线点. n 次 Bézier 曲线定义为:

$$B(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad t \in [0, 1]$$

其中 $\binom{n}{i}$ 是二项式系数, $\binom{n}{i} t^i (1-t)^{n-i}$ 称为 Bernstein 基函数.

Chapter 3

程序说明

3.1 编程平台

本实验使用 MATLAB R2024a 完成, 不需要额外的工具箱.

3.2 程序结构

程序包含以下主要部分:

- 主界面: 创建图形窗口和用户界面控件
- 算法选择复选框: 允许用户选择显示哪种算法计算的曲线
- 控制多边形交互: 使用 MATLAB 的 `drawpolyline` 函数实现控制点的交互编辑
- 曲线更新机制: 通过事件监听器实时更新曲线显示
- 算法实现函数: 分别实现 de Casteljau 算法和 Bernstein 基函数方法

3.3 程序使用说明

1. 运行程序后, 在图形窗口中点击创建控制点, 形成控制多边形
2. 通过右下角的复选框选择要显示的算法曲线
3. 绿色曲线为 de Casteljau 算法结果, 蓝色曲线为 Bernstein 基函数方法结果
4. 拖动控制点可以实时观察曲线的变化
5. 可以同时显示两种算法的曲线进行比较

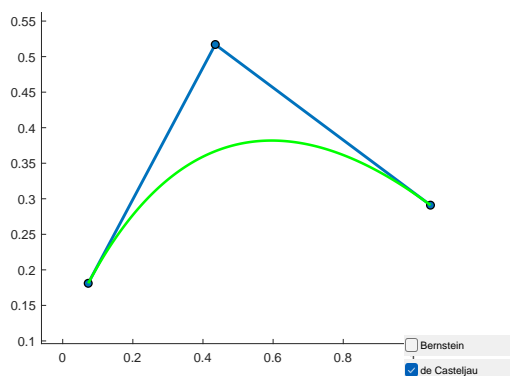
Chapter 4

实验结果

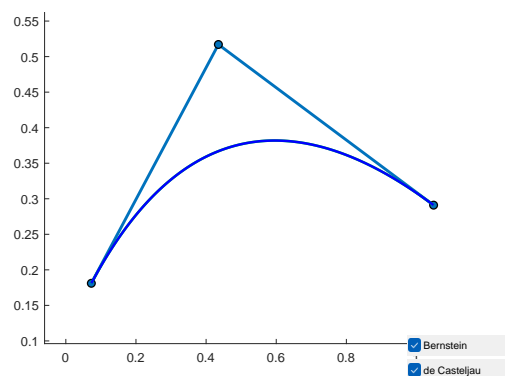
4.1 基本结果

通过实验, 我们成功实现了交互式 Bézier 曲线编辑系统. 程序能够:

- 实时响应用户对控制点的编辑操作
- 同时显示两种算法计算的曲线
- 处理任意数量 (≥ 2) 的控制点



(a) 初始状态: 3 个控制点, de Casteljau 算法

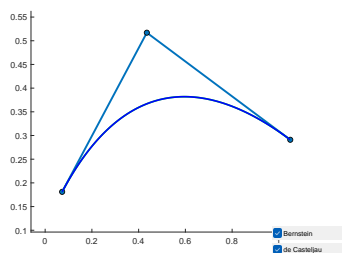


(b) 双算法对比: 两条曲线完全重合

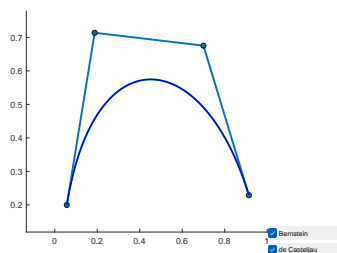
图 4.1: Bézier 曲线基本显示效果

4.2 不同阶数曲线

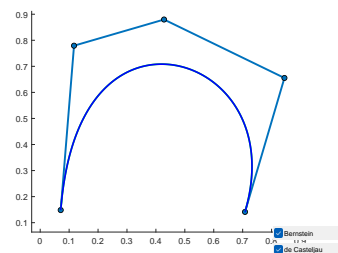
实验测试了不同数量控制点的情况:



(a) 3 个控制点 (二次曲线)



(b) 4 个控制点 (三次曲线)



(c) 5 个控制点 (四次曲线)

图 4.2: 不同阶数的 Bézier 曲线

4.3 算法精度验证

通过数值计算验证两种算法的等价性:

```
control_points = [0,0; 1,2; 3,1; 4,3];
t = 0:0.1:1;
curve1 = bezier_decasteljau(control_points, t);
curve2 = bezier_bernstein(control_points, t);
max_difference = max(abs(curve1(:) - curve2(:)));
```

测试结果显示, 两种算法计算结果的差异在 10^{-16} 数量级, 这可以归因于浮点数计算精度误差, 验证了两种算法在理论上的等价性.

Chapter 5

讨论与分析

5.1 算法比较

5.1.1 de Casteljau 算法的优势

- 几何直观: 算法过程具有清晰的几何意义
- 数值稳定: 递归插值过程数值稳定性好
- 易于分割: 天然支持曲线的分割操作

5.1.2 Bernstein 基函数方法的优势

- 计算效率: 对于固定参数点, 预计算基函数后可快速计算
- 理论分析: 便于进行数学分析和性质证明
- 并行计算: 各点的计算相互独立, 适合并行化

5.2 交互设计考虑

程序采用了以下交互设计策略:

- 实时更新: 控制点移动时立即更新曲线显示
- 视觉区分: 使用不同颜色区分两种算法的曲线
- 界面简洁: 控件放置在角落, 避免遮挡主要绘图区域

Chapter 6

结论

本实验成功实现了交互式 Bézier 曲线编辑系统, 具有以下成果:

- 完整实现了 de Casteljau 算法和 Bernstein 基函数方法
- 开发了友好的用户界面, 支持实时曲线编辑
- 验证了两种算法在数值精度范围内的等价性
- 分析了两种算法各自的优势和适用场景

通过本次实验, 我们深入理解了 Bézier 曲线的数学原理和计算方法, 掌握了计算机辅助几何设计中曲线表示和编辑的基本技术. 这些知识为后续学习更复杂的几何造型技术奠定了坚实基础.