《计算机辅助几何设计》作业 1

ID 号: 47　　姓名: 陈文博

2020 年 9 月 16 日

作业要求：

Input：已知平面内 n 个点 $p_j(x_j, y_j)$，$j = 1, 2..., n$。

Output：拟合这些点的函数

1　插值型拟合原理

设基函数集合为 $\{\varphi_i(x)\}(i = 1, 2, \cdots)$，插值方程表示为：

$$f(x) = \sum_{i=1}^{n} \alpha_i \varphi_i(x) \tag{1}$$

对输入点 $(x_i, y_i)(i = 1, 2, \cdots, n)$，拟合方程表示为：

$$\begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{2}$$

即：

$$\boldsymbol{A\alpha} = \boldsymbol{y} \tag{3}$$

求解线性方程组即可求得插值拟合参数 $\boldsymbol{\alpha}$:

对于多项式函数插值，基函数取为基函数，即：

$$\varphi_i(x) = x^i, \quad i = 0, 1, 2, \cdots, n-1 \tag{4}$$

对于径向基函数插值，基函数取径向基函数，即：

$$\varphi_i(x) = \frac{1}{|x - p_i|^2 + d}, \quad i = 0, 1, 2, \cdots, n-1 \tag{5}$$

2 思考题

(1) 变量比方程多，如何加约束条件？

可考虑取输入 $\{y_i\}$ 的均值，即：

$$b_0 = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{6}$$

相当于把曲线沿 y 轴方向平移到各个点 y 方向均值为零处，可以实现 y 方向的平移不变性

(2) 常数项 b_0 也可以改为一个低次（比如 2 次或 3 次）的多项式，相应也要加约束条件

拟合方程:

$$f(x) = \sum_{k=0}^{m} a_i x^k + \sum_{i=1}^{n} b_i g_i(x) \tag{7}$$

可以先对多项式部分 $f^{(1)}(x) = p(x) = \sum_{k=0}^{m} a_i x^k$ 做逼近拟合，然后对下面拟合方程进行插值拟合计算:

$$f(x) - p(x) = \sum_{i=1}^{n} b_i g_i(x) \tag{8}$$

从而得到最终的结果

3 结果分析

- 插值拟合将经过所有数据点
- 插值拟合曲线首尾振荡明显，内部较为平滑
- 径向基函数插值相较多项式插值稳定性更高一点
- 径向基函数参数 d 越接近于零，插值曲线越趋于直线

4 程序说明

本实验基于 C++ 和 OpenGL 进行编写，使用开源即时绘制 UI 库 imgui 进行图形界面设计，框架使用了 imgui 的 docking 分支能够更好地进行窗口布局

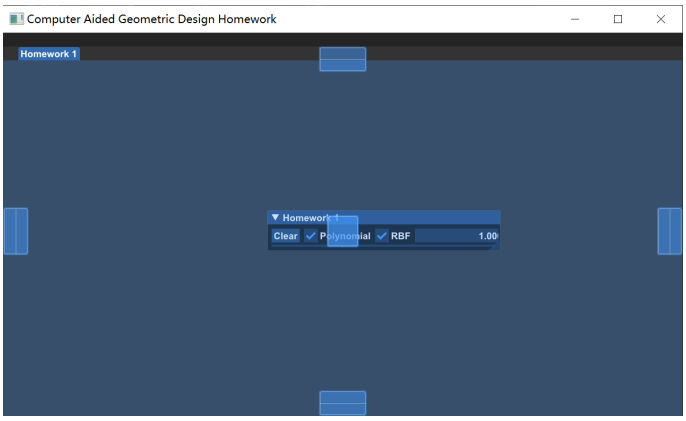

图 1:

实验效果如下：

多项式插值

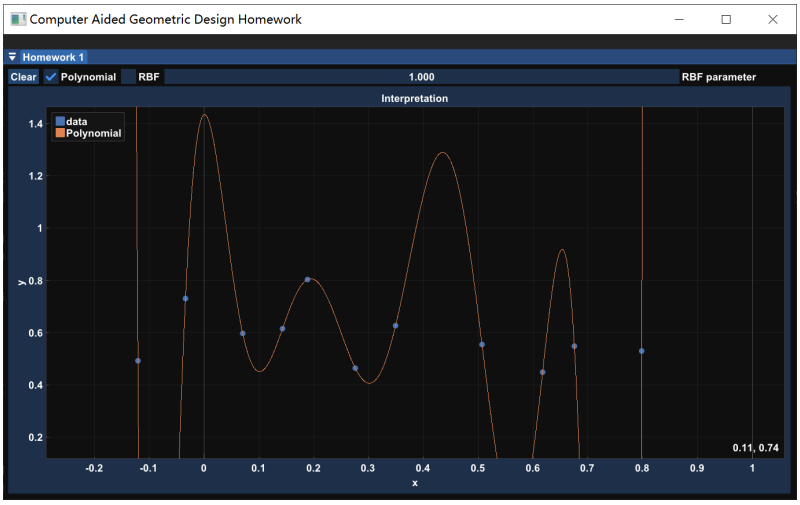

图 2: 多项式插值

径向基函数插值

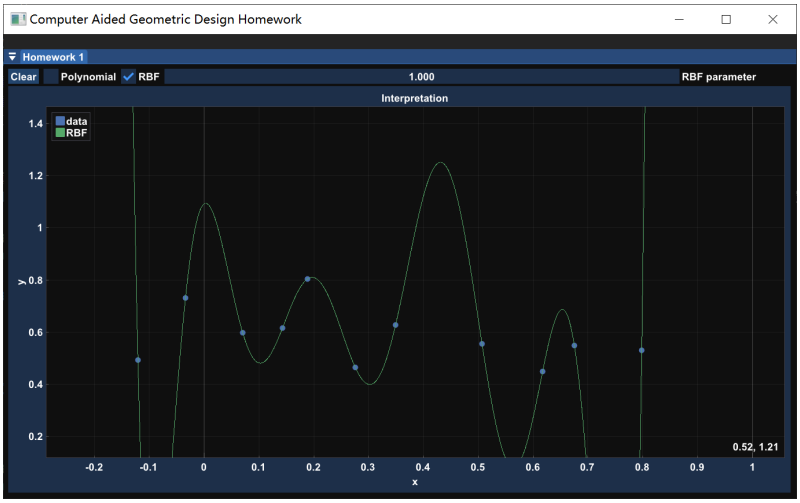

图 3: 径向基函数插值

修改径向基函数参数

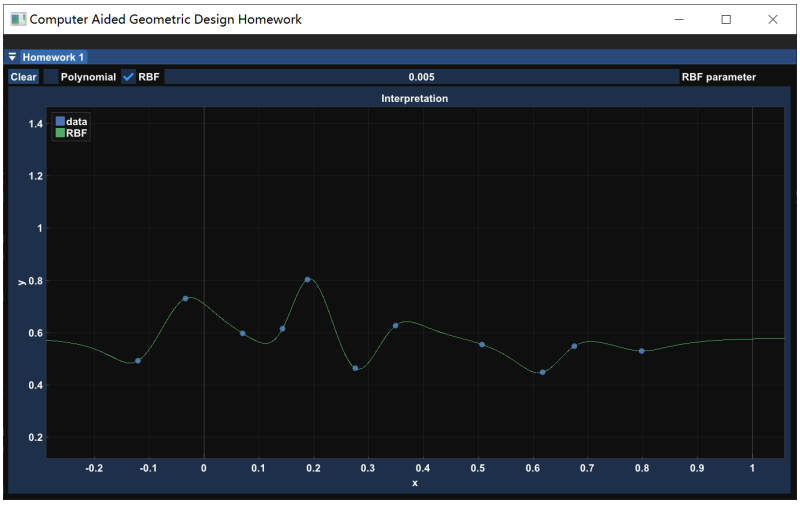

图 4: 径向基函数插值，参数 $d = 0.005$

同时显示

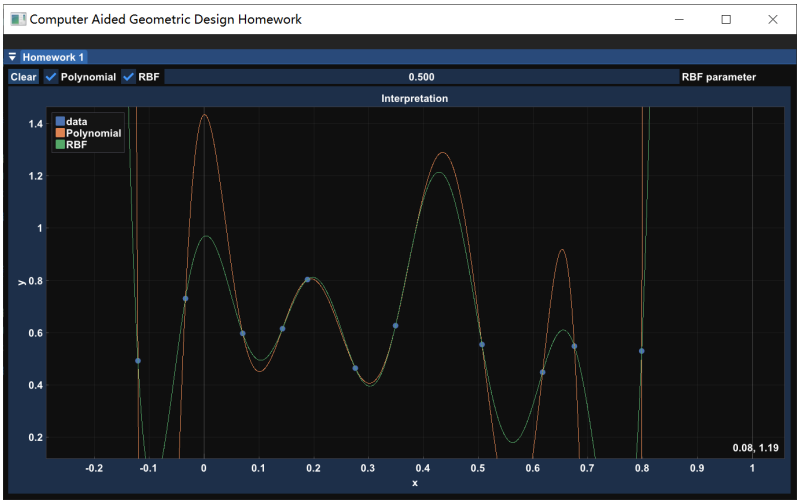

图 5: 多项式 & 径向基

交互说明

- 鼠标左键拖动坐标系可移动坐标系的观察区域
- 鼠标右键单击可打开坐标系设置选项
- 鼠标右键拖动可放大覆盖的区域
- 鼠标滑轮可缩放坐标系
- 鼠标中键在空白区域单击添加坐标点
- 鼠标中键点击已有坐标点进行拖动可编辑坐标点位置
- 鼠标中键点击已有坐标点并按下键盘 delete 键可删除坐标点

程序说明

项目地址：*https://github.com/Chaphlagical/Chaf-Engine/tree/CAGD*

核心算法代码：*src/CAGD/HW1* 文件夹中的 *polynomial.h/.cpp* 和 *RBF.h/.cpp*

线性方程组求解部分使用 Eigen 库与 QR 分解方法进行完成