homework4

三次样条函数

三弯矩方程推导

- 点集为$\{(x_1,y_1),...,(x_n,y_n)\}$，共$n$个点。假设经过点$[(x_i,y_i),(x_{i+1},y_{i+1})]$的曲线方程为$S_i$，则：

$$
\begin{aligned}
S_i(x) &= a_i + b_i\left(x-x_i\right) + c_i\left(x-x_i\right)^2 + d_i\left(x-x_i\right)^3 \\
S_i'(x) &= b_i + 2c_i\left(x-x_i\right) + 3d_i\left(x-x_i\right)^2 \\
S_i''(x) &= 2c_i + 6d_i\left(x-x_i\right)
\end{aligned}
$$

- 由于经过(x_i,y_i)，带入可得：

$$a_i = y_i$$

- 由于经过点(x_{i+1},y_{i+1})，C^0连续，令$h=x_{i+1}-x_i$，可得：

$$a_{i+1} = a_i + b_ih_i + c_ih_i^2 + d_ih_i^3$$

- 由于C^1连续，且：

$$
\begin{aligned}
S_i'\left(x_{i+1}\right) &= b_i + 2c_i\left(x_{i+1}-x_i\right) + 3d_i\left(x_{i+1}-x_i\right)^2 = b_i + 2c_ih_i + 3d_ih_i^2 \\
S_{i+1}'\left(x_{i+1}\right) &= b_{i+1} + 2c_i\left(x_{i+1}-x_{i+1}\right) + 3d_i\left(x_{i+1}-x_{i+1}\right)^2 = b_{i+1}
\end{aligned}
$$

  可得

$$b_{i+1} = b_i + 2c_ih_i + 3d_ih_i^2$$

- 由于C^2连续，可得：

$$2c_i + 6h_id_i = 2c_{i+1}$$

- 假设m_i为插值点的二阶导数值，则$m_i=S_i''(x_i)=2c_i$，$m_n=S_{n-1}''(x_n)$，则可得：

$$m_i + 6h_id_i = m_{i+1}$$

  即：

$$d_i = \frac{m_{i+1}-m_i}{6h_i}$$

  进一步，我们可以用m与已知量来表达所有的未知量：

$$b_i = \frac{y_{i+1}-y_i}{h_i} - \frac{m_ih_i}{2} - \frac{(m_{i+1}-m_i)h_i}{6}$$

- a_i,b_i,c_i,d_i带入到$b_{i+1}=b_i+2c_ih_i+3d_ih_i^2$中，可得到只包含$m$的等式：

$$h_i m_i + 2\left(h_i + h_{i+1}\right) m_{i+1} + h_{i+1} m_{i+2} = 6\left[\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}\right]$$

这样的方程一共有$n-2$个，而需要求解的未知数个数为n个。为了能成功解方程，我们还需要别的两个条件，也就是下文说的边界条件。

边界条件

- ▼ 自由端：指定曲线在两个端点处的二阶导数值，当两个端点的二阶导数值为0，称为自然三次样条，自然边界条件。
- ▼ 夹持端：指定曲线在两个端点的一阶导数值
- ▼ 其他：
  - 首末两端为抛物线
  - 周期端
  - 混合边界条件

不同的边界条件可以得到不同的线性方程组。当为自由端时：

$$\begin{bmatrix} 1 & 0 & 0 & & \cdots & & 0 \\ h_1 & 2\left(h_1+h_2\right) & h_2 & 0 & & & \\ 0 & h_2 & 2\left(h_2+h_3\right) & h_3 & & & 0 \\ 0 & 0 & h_3 & 2\left(h_3+h_4\right) & h_4 & & \vdots \\ \vdots & & & & \ddots & \ddots & \\ & \cdots & & 0 & h_{n-2} & 2\left(h_{n-2}+h_{n-1}\right) & h_{n-1} \\ 0 & \cdots & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ \vdots \\ m_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \frac{y_5-y_4}{h_4} - \frac{y_4-y_3}{h_3} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \\ 0 \end{bmatrix}$$

将第一行第一列以及最后一行最后一列去掉，并不会影响到中间m的求解，也就是老师给的补充材料中的方程组。

当使用夹持端时，线性方程组左侧变为：

$$\begin{bmatrix} 2h_1 & h_1 & 0 & \cdots & \cdots & 0 \\ h_1 & 2\left(h_1+h_2\right) & h_2 & 0 & & \vdots \\ 0 & h_2 & 2\left(h_2+h_3\right) & h_3 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_{n-2} & 2\left(h_{n-2}+h_{n-1}\right) & h_{n-1} \\ 0 & \cdots & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

可用**追赶法**求解这样的方程组。

追赶法（Thomas Algorithm）

现在有方程$Ax=f$：

$$A=\begin{pmatrix} b_1 & c_1 & \cdots & & \\ a_2 & b_2 & c_2 & & \\ \vdots & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}, x=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}, f=\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}$$

并且满足：

- $|b_1|>|c_1|>0$
- $|b_i|\geq|a_i|+|c_i|$
- $|b_n|>|a_n|>0$

A称为对角占有的三对角线矩阵，且$\det A\neq 0$ 。

对于这样的矩阵，可以进行唯一的克劳特分解(Crout decomposition)：

$$A=\begin{pmatrix} p_1 & & \cdots & & \\ a_2 & p_2 & & & \\ \vdots & \ddots & \ddots & \ddots & \\ & & a_{n-1} & p_{n-2} & \\ & & & a_n & p_n \end{pmatrix}\cdot\begin{pmatrix} 1 & q_1 & \cdots & & \\ & 1 & q_2 & & \\ \vdots & \ddots & \ddots & \ddots & \\ & & & 1 & q_{n-1} \\ & & & & 1 \end{pmatrix}=PQ$$

其中：

$$\begin{cases} p_1=b_1 \\ q_i=c_i/p_i \quad (i=1,2,\cdots,n-1) \\ p_i=b_i-a_iq_{i-1} \quad (i=2,3,\cdots,n) \end{cases}$$

这时候，原方程组变成了$PQx=f$，可以转化成解两个方程组：$Py=f$与$Qx=y$。

1. $Py=f$的解为：

$$\begin{cases} y_1=f_1/p_1 \\ y_i=(f_i-a_i\cdot y_{i-1})/p_i \quad (i=2,3,\cdots,n) \end{cases}$$

2. $Qx=y$的解为：

$$\begin{cases} x_n=y_n \\ x_i=y_i-q_i\cdot x_{i+1} \quad i=n-1,\cdots,2,1 \end{cases}$$

注：网上查阅到的资料是c_1与a_n需要大于零，但是我自己实际推导中，它们可以等于0，并不会导致结果有错误，但是为了严谨我依然保留了大于零的写法。

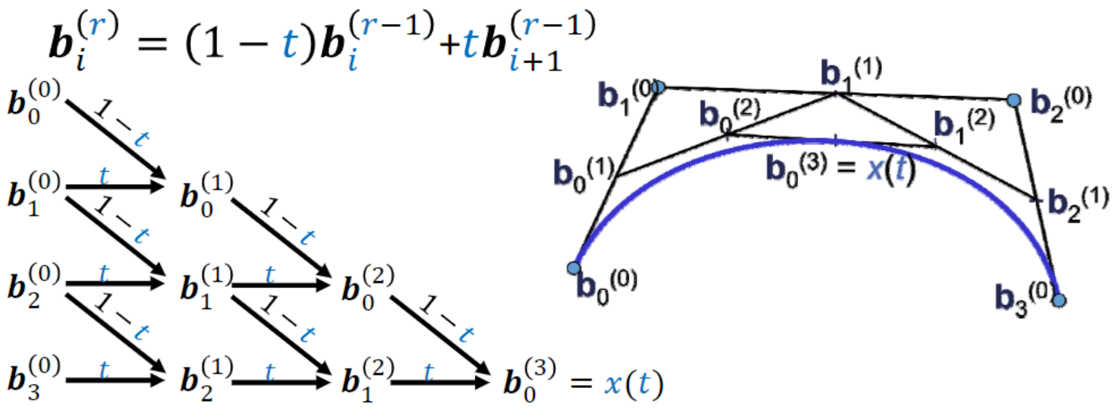
Bezier曲线

Bezier曲线的各种性质非常好，递推公式也是一大串，但是它的实现却是出乎意料的简单。这也许就是数学的魅力。使用de Casteljau算法可以轻易画出bezier曲线。

算法输入为控制顶点$\boldsymbol{b}_0,\boldsymbol{b}_1,\ldots\boldsymbol{b}_n\in\mathbb{R}^3$，以及$t$，输出为曲线在$t$的值。按照下面的公式，对控制点进行递归组合：

$$\begin{aligned}\boldsymbol{b}_i^0(t)&=\boldsymbol{b}_i, & i&=0,\ldots,n\\ \boldsymbol{b}_i^r(t)&=(1-t)\boldsymbol{b}_i^{r-1}(t)+t\boldsymbol{b}_{i+1}^{r-1}(t) & & \\ r&=1,\ldots,n & i&=0,\ldots,n-r\end{aligned}$$

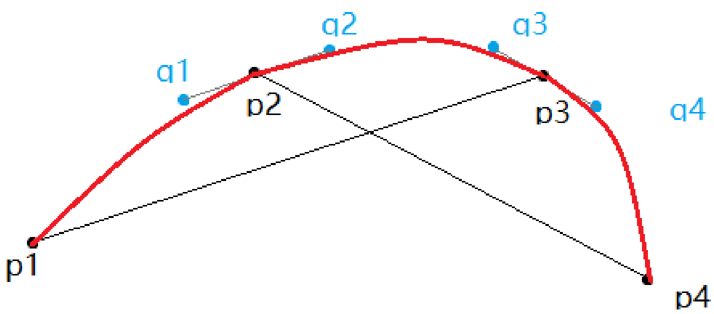
最终，$\boldsymbol{b}_0^n(t)$就是曲线在t的值。



De Casteljau scheme

Bezier插值

Bezier插值也就是分段Bezier曲线，并在端点处保持C^1连续性。利用Bezier曲线的性质非常容易做到这一点。具体做法是，给定n个顶点（$n\geq 3$）。对于顶点$p_i(i\in[1,n-2])$，我们根据点p_i,p_{i+2}的位置来为p_{i+1}构造两个控制顶点，如下图：



上图中，p_1，p_2，p_3，p_4为待插值点，而q_1，q_2，q_3，q_4为增加的额外的控制顶点，分别画了3段bezier曲线。控制顶点的添加规则就是与黑色线段平行，且长度为其$\frac{1}{6}$。

实现

本次作业用taichi来做图形化界面，实现了7种曲线，分别如下：

- 自然边界条件的三次样条曲线

  推导如上。

- 夹持端的三次样条曲线

推导如上。

- 满足C^1的样条曲线，顶点两侧曲线的一阶切线方向大小相同，但并非C^2连续

  这里只需要C^2不连续即可。因此我使用了一个简单的trick，就是让顶点左侧的二阶导数等于右侧二阶导数的k倍，其他的条件保持不变。这样，最后的等式变成：

  $$h_i m_i + 2(k \cdot h_i + h_{i+1}) m_{i+1} + k \cdot h_{i+1} m_{i+2} = 6 \left[\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right]$$

- 满足G^1的样条曲线，顶点两侧曲线一阶切线方向相同，但大小不同

  同上，我做了一个trick，就是让顶点左侧的一阶导数等于右侧一阶导数的k（$k > 0$）倍，其他的条件保持不变。这样左右导数大小不一致，但是x, y方向上的分量都是等量缩放了k倍，最后的等式变成：

  $$h_i m_i + 2(h_i + k \cdot h_{i+1}) m_{i+1} + k \cdot h_{i+1} m_{i+2} = 6 \left[k \cdot \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right]$$

- 满足G^0的样条曲线，顶点两侧曲线一阶切线方向大小都不相同。其实这个很简单，因为x，y分别是两个函数来拟合的，用上述同样的方法，对于x来说，左侧一阶导数是右侧的k_1倍，对于y，左侧是右侧的k_2倍，只要$k_1 \neq k_2$，那么方向与大小都不同了。为了让结果更明显，甚至可以对k取负值（取负值的话，即使两个k相等，依然不是G^1连续的，这时候方向与之前是相反的）。

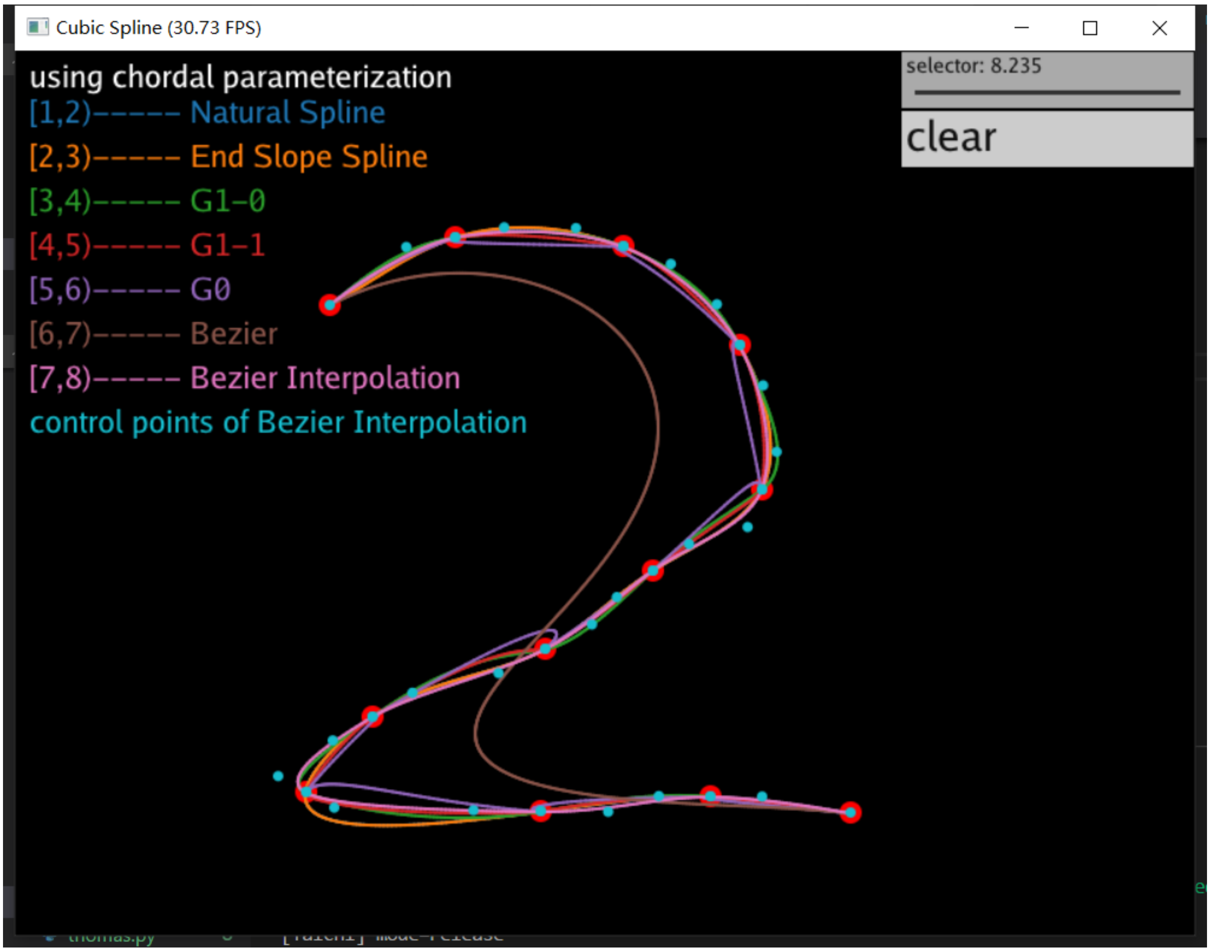
- Bezier曲线

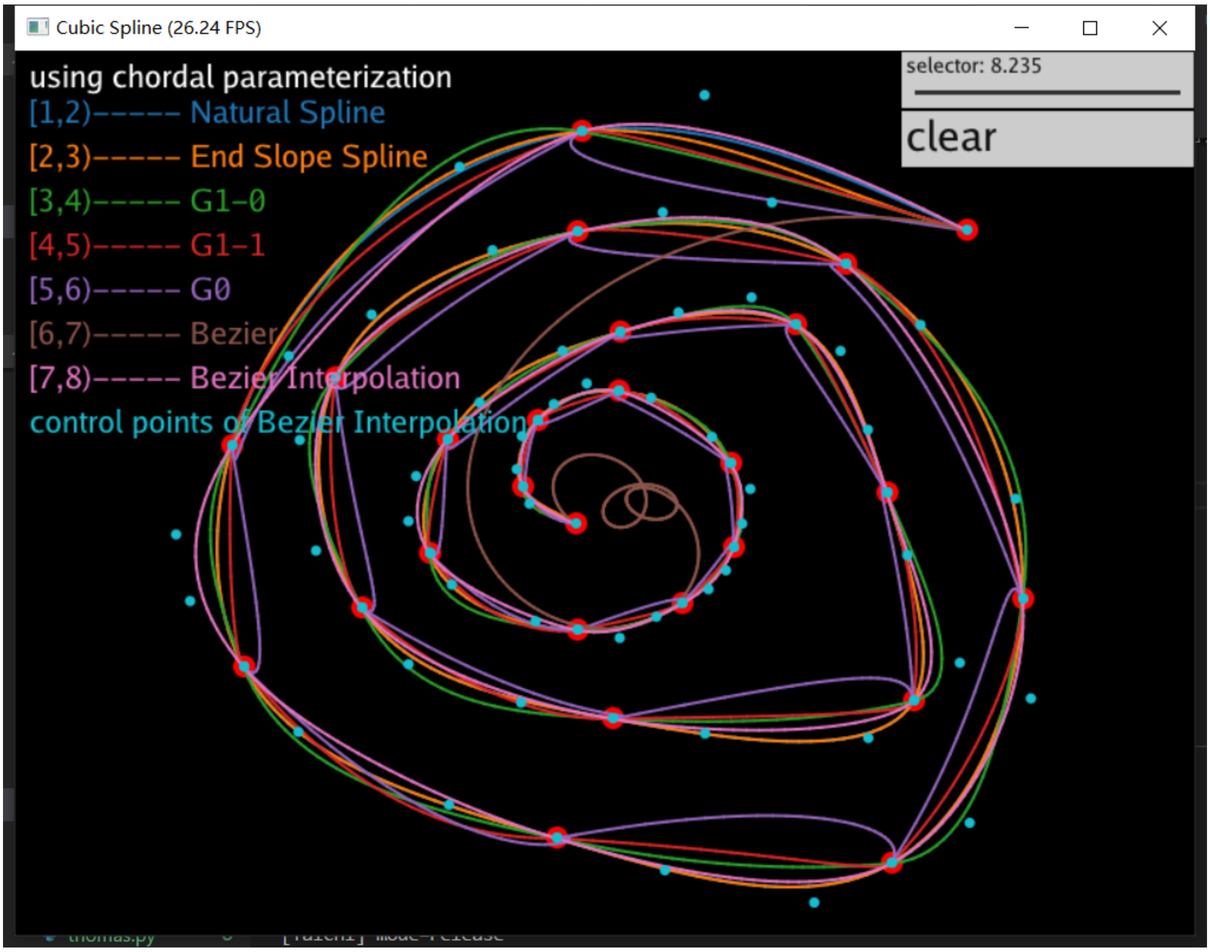
  简单的用de Castelijau算法，递归写法只需要十几行代码就能实现。

- Bezier插值样条曲线

  分段Bezier插值，增加控制顶点。需要注意的是，当给一个参数t时，需要首先判断该t在哪两个点中间，确定位置后，再对这个t进行normalize，生成这一段bezier曲线间的相对参数，利用这个相对参数来确定点的位置。在图形界面中，我会用另外一种颜色，来画出所有的控制顶点。

结果示例如下：


Cubic Spline (30.73 FPS)
selector: 8.235
clear
using chordal parameterization
[1,2)----- Natural Spline
[2,3)----- End Slope Spline
[3,4)----- G1-0
[4,5)----- G1-1
[5,6)----- G0
[6,7)----- Bezier
[7,8)----- Bezier Interpolation
control points of Bezier Interpolation



这次作业中，曲线会根据点的添加实时改变，也可以对已经添加的端点进行移动，这样可以更好地看出样条曲线的局部性质。更多的内容可以查看附带的video demo。

谢谢老师与助教！