几何建模与处理基础
GAMES102
作业 4：三次插值样条

Qingjun CHANG

2020 年 11 月 8 日

1 目标

- 学习三次样条函数的求解
- 了解曲线设计和编辑工具的原理

2 三次样条函数

问题：对于给定 $\mathbb{R}^2$ 域内一组有序点 $\{\mathbf{Q}_i\}_{i=0}^n$，对应参数为 $\{\mathrm{t}_i\}_{i=0}^n$，找到一组分段三次多项式函数插值每个数据点。

2.1 考虑插值点处 C^2 连续 (平滑顶点)

对于 $n+1$ 个数据点，相邻两点之间用三次多项式表达，则有 n 段三次多项式，用待定系数的方式设第 $i(i=0,1,2,\dots,n)$ 段向量型多项式函数为

$$\mathbf{P}_i(\mathrm{t}) = \boldsymbol{a}_{i,0} + \boldsymbol{a}_{i,1}\mathrm{t} + \boldsymbol{a}_{i,2}\mathrm{t}^2 + \boldsymbol{a}_{i,3}\mathrm{t}^3 \tag{1}$$

从式 (1) 可以看出总共 $4n$ 个未知数，C^2 连续需要满足约束条件：

- 每段多项式在端点处插值

$$\mathbf{P}_i(\mathrm{t}_i) = \mathbf{Q}_i \qquad \mathbf{P}_i(\mathrm{t}_{i+1}) = \mathbf{Q}_{i+1} \qquad i=0,1,2,\dots,n$$

- 中间点一阶导连续

$$\mathbf{P}'_{i-1}(\mathrm{t}_i - 0) = \mathbf{P}'_i(\mathrm{t}_i + 0) \qquad i=1,2,\dots,n-1$$

- 中间点二阶导连续

$$\mathbf{P}''_{i-1}(\mathrm{t}_i - 0) = \mathbf{P}''_i(\mathrm{t}_i + 0) \qquad i=1,2,\dots,n-1$$

这里总共 $4n-2$ 个方程，还需要两个约束条件，这里我们选择首尾两点二阶导等于 0，即自然边界条件：

$$\mathbf{P}''_0(\mathrm{t}_0) = \mathbf{P}''_{n-1}(\mathrm{t}_n) = 0 \qquad i = 1, 2, \dots, n-1$$

参见课程资料 [1] 引入中间变量弯矩，可得每段多项式的表达式：

$$\mathbf{P}_i(\mathrm{t}) = \frac{\mathbf{M}_i}{6h_i}(\mathrm{t}_{i+1}-\mathrm{t})^3 + \frac{\mathbf{M}_{i+1}}{6h_i}(\mathrm{t}-\mathrm{t}_i)^3 + \left(\frac{\mathbf{Q}_{i+1}}{h_i} - \frac{\mathbf{M}_{i+1}h_i}{6}\right)(\mathrm{t}-\mathrm{t}_i) + \left(\frac{\mathbf{Q}_i}{h_i} - \frac{\mathbf{M}_i h_i}{6}\right)(\mathrm{t}_{i+1}-\mathrm{t}) \tag{2}$$

其中 $h_i = \mathrm{t}_{i+1} - \mathrm{t}_i$，$\mathbf{M}$ 是如下三对角线性系统的解，$\mathbf{M}_0 = \mathbf{M}_n = 0$。

$$\begin{pmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_{n-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_{n-1} \end{pmatrix} \tag{3}$$

其中 $u_i = 2(h_i + h_{i-1})$，$\boldsymbol{v}_i = \frac{6}{h_i}(\mathbf{Q}_{i+1} - \mathbf{Q}_i) - \frac{6}{h_{i-1}}(\mathbf{Q}_i - \mathbf{Q}_{i-1})$

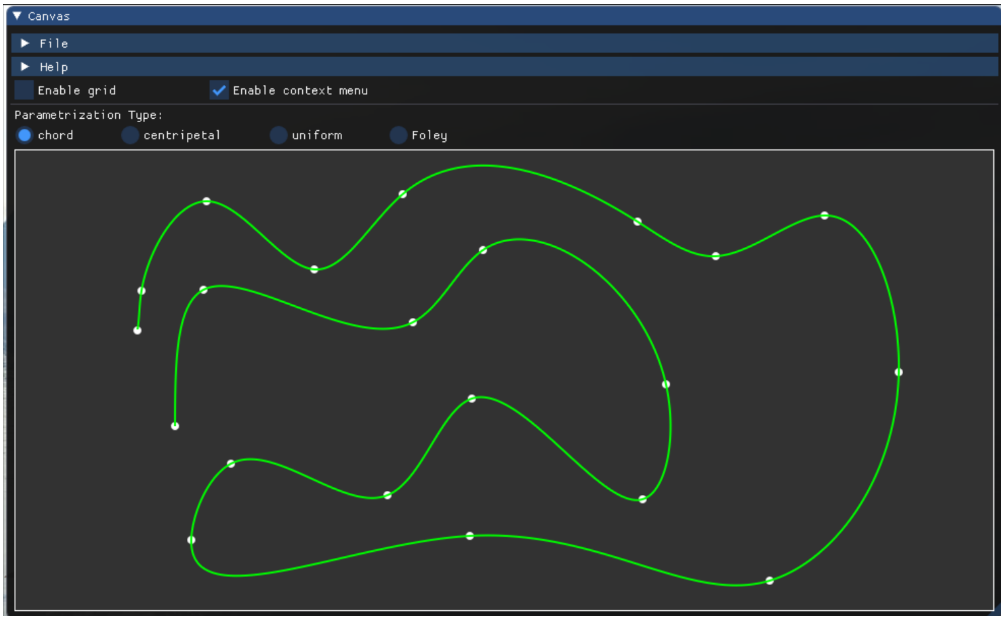

图 1: 平滑顶点绘制 (全局 C^2 连续)

2.2 考虑插值点处 C^0 连续 (角部顶点)

对于每一段给定插值点处的左导数 f'_- 和右导数 f'_+，加上插值该段的两点，4 个未知数 4 个方程可以求解处该段的三次多项式函数：

例如在 $[\mathrm{t}_i, \mathrm{t}_{i+1}]$ 区间：

$$\begin{cases} \boldsymbol{a}_{i,0} + \boldsymbol{a}_{i,1}\mathrm{t}_i + \boldsymbol{a}_{i,2}\mathrm{t}_i^2 + \boldsymbol{a}_{i,2}\mathrm{t}_i^3 = \mathbf{Q}_i \\ \boldsymbol{a}_{i,0} + \boldsymbol{a}_{i,1}\mathrm{t}_{i+1} + \boldsymbol{a}_{i,2}\mathrm{t}_{i+1}^2 + \boldsymbol{a}_{i,2}\mathrm{t}_{i+1}^3 = \mathbf{Q}_{i+1} \\ \boldsymbol{a}_{i,1} + 2\boldsymbol{a}_{i,2}\mathrm{t}_i + 3\boldsymbol{a}_{i,3}\mathrm{t}_i^2 = \mathbf{P}'_i(\mathrm{t}_i + 0) \\ \boldsymbol{a}_{i,1} + 2\boldsymbol{a}_{i,2}\mathrm{t}_{i+1} + 3\boldsymbol{a}_{i,3}\mathrm{t}_{i+1}^2 = \mathbf{P}'_{i+1}(\mathrm{t}_i - 0) \end{cases} \tag{4}$$

求解该式可得到每段的三次多项式。

图 2: 角部顶点绘制 (型值点处为 C^0 连续，其他点 C^2 连续)

2.3 考虑插值点处 C^1 连续 (直线点)

在角部顶点基础上保证一点的左导数与右导数方向在同一直线即可。即方程 (4) 中 $\mathbf{P}_i'(\mathrm{t}_i + 0) = \lambda \mathbf{P}_{i+1}'(\mathrm{t}_i - 0)$

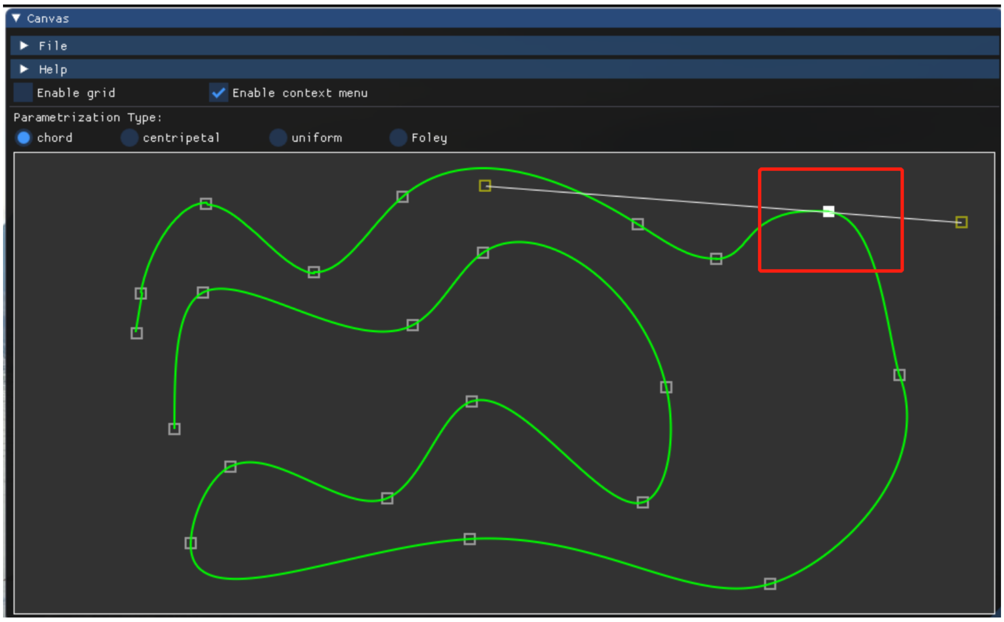

图 3: 角部顶点绘制 (型值点处为 C^1 连续，其他点 C^2 连续)

3 主要代码

——更多示例结果可参见附带的视频文件——

3.1 code

高斯-塞德尔迭代求解三对角：

```cpp
void GaussSeidel(std::vector<float> h, std::vector<float> u,
 std::vector<float> v, std::vector<float>* M) {
  std::vector<float> _M0 = *M;
  while (true) {
    for (int i=0; i<u.size(); ++i) {
      (*M)[i+1]=(v[i]-h[i]*(*M)[i]-h[i+1]*_M0[i+2])/u[i];
    }

  // 迭代停止条件
  if ((Eigen::VectorXf::Map((*M).data(),(*M).size())-Eigen::
   VectorXf::Map(_M0.data(),_M0.size())).norm()<EPSILON) {
    break;
  }
  _M0 = *M;
  }
}
```

求解给定导数后的分段样条：

```cpp
if (validDerivative) {
  float x, y;
  auto A = [=](int i) -> Eigen::Matrix4f {
    Eigen::Matrix4f a;
    a << 1, t[i], t[i] * t[i], t[i] * t[i] * t[i],
      1, t[i+1], t[i+1]*t[i+1], t[i+1]*t[i+1]*t[i+1],
      0, 1, 2 * t[i], 3 * t[i] * t[i],
      0, 1, 2 * t[i+1], 3*t[i+1]*t[i+1];
    return a;
  };
  auto B = [=](int i, int xy) -> Eigen::Vector4f {
    Eigen::Vector4f b;
    b << points[i][xy],
    points[i + 1][xy],
    (*derivative)[i].second[xy],
    (*derivative)[i + 1].first[xy];
    return b;
  };
  // 四维矩阵可用Eigen，不会影响交互延迟
  Eigen::Vector4f ax = A(segment_index).colPivHouseholderQr
   ().solve(B(segment_index, 0));
```

```
    Eigen::Vector4f ay = A(segment_index).colPivHouseholderQr().solve(B(segment_index, 1));
    x = ax[0]+ax[1]*t0+ax[2]*t0*t0+ax[3]*t0*t0*t0;
    y = ay[0]+ay[1]*t0+ay[2]*t0*t0+ay[3]*t0*t0*t0;
    return Ubpa::pointf2(x, y);
}
```

参考文献

[1] GAMES-102 Courses. 三次样条插值函数. [EB/OL]. http://staff.ustc.edu.cn/~lgliu/Courses/GAMES102_2020/documents/GAMES102-suppl-2-CubicSpline.pdf 2020-11-02.