北京航空航天大學
BEIHANG UNIVERSITY

CAGD 大作业程序设计说明

专 业 航空宇航制造工程

学 员 邹 捷

学 号 ZY1807315

老 师 赵 罡

机械工程及自动化学院

2019 年 03 月 03 日

目录

1.程序功能 3
2.程序框架 4
3.程序语言与调试环境 4
4.BspineDrawing 程序主要函数与注释 4
4.1 主要的函数命名与注释 5
5.程序使用说明 9
5.1 在曲线界面绘出 v 向曲线 9
5.2 修改控制点 10
5.3 在曲面界面生成想要的曲面 11
5.4 修改控制网格 12

1. 程序功能

程序名：B-spine 绘图软件

功能介绍：使用鼠标绘出不同参数设置情况下的 b 样条曲线与曲面，并可以进行 u 向与 v 向基函数显示，修改控制顶点等功能。

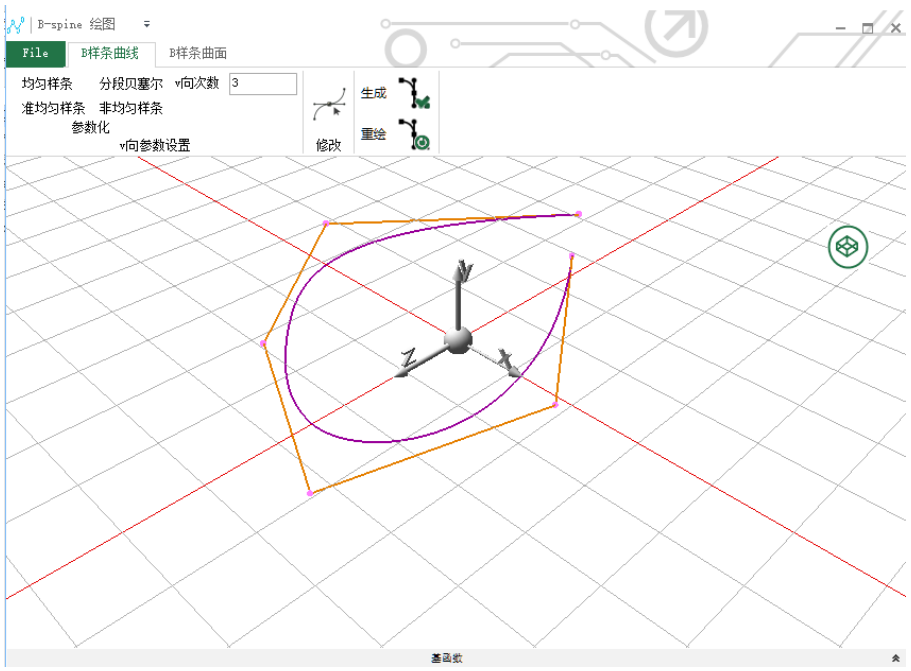


图 1. B 样条曲线绘图

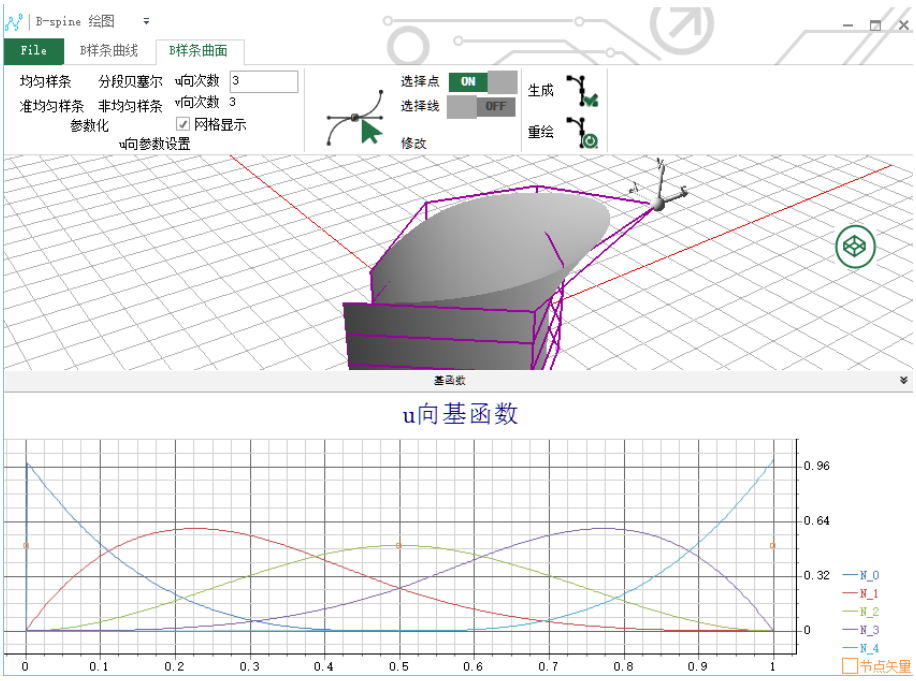


图 2. B 样条曲面绘图

2. 程序框架

该程序外部界面调用 DotNetBar 控件下的界面工具，只有一个窗体 Form1 完成所有的绘图功能，内部三维可视化界面采用 SharpGL 封装的 OpenGL 接口。

主窗口 Form1 是程序负责操作逻辑和显示类，其中方法分为三个部分：初始化和调整，绘图循环，和事件触发的函数。

曲线曲面计算使用 Bspine3D 类，该类使用两种构造函数分别对应曲线和曲面的情况，对外只有三个只读属性接口，分别对应曲线点，曲面点和基函数的点。

3. 程序语言与调试环境

程序使用 C#语言编写,Windows10 的操作系统平台下使用 visual studio 2012 和 visual studio 2017 均可以调试

4. BspineDrawing 程序主要函数与注释

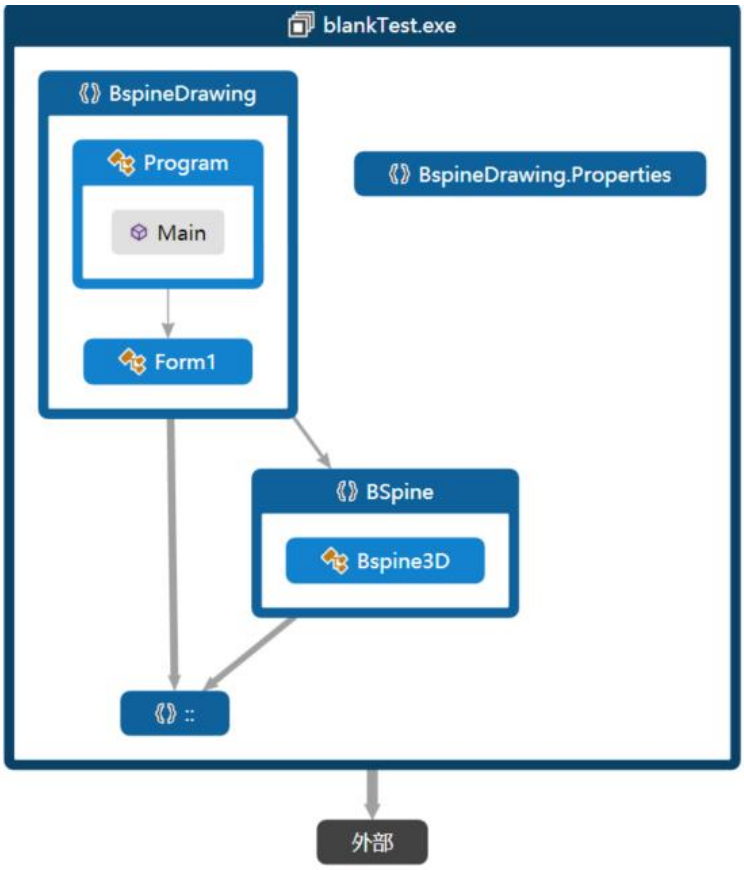


图 3. 主函数调用流程（Main 函数调用 Form1 窗口，Form1 调用生成曲线曲面的类 Bspine3D）

4.1 主要的函数命名与注释

Form1.cs

成员变量:

```
private double rotation = 0.0f;
        private double temprotation = 0.0f;
        private double scale = 1.0f;
        private bool recoverCenter = false;
        private bool viewChanged = false;
        private bool isScale = false;
        private bool isRotate = false;
        //绘图与修改
        private bool isModify = false;
        private bool isDraw = true;
        private bool isStrech = false;
        private bool pointModify = false;
        private bool isAddpoint = false;
        private bool isGenctrlGrid = false;
        private Point selectedPt;
        private int[] selectindex = new int[3];
        private int axisindex = 0;
        private int i_highlight = 0;
        private int i_u_surfacehight = 0;
        private int i_v_surfacehight = 0;
        //切换四个视图
        private bool isFrontView = false;
        private bool isLeftView = false;
        private bool isTopView = false;
        private bool isPerspective = true;
        //光源
        private float[] lightPos = new float[] { -1, 3, 1, 1 };
        private float[] lightSphereColor = new float[] { 1f, 1f, 1f };
        private IList<float[]> lightColor = new List<float[]>();
        private double[] lookatValue = { 2, 2, 2, 0, 0, 0, 0, 1, 0 };
        private IList<double[]> viewDefaultPos = new List<double[]>();
        //控制点与曲面曲线存储区
        private List<Point> ctrlPoints = new List<Point>();
        private Point[] winPoints;
        private Point[] winCoodPos = new Point[3];
        private Point[,] winSurfacePts;
        private Point[] BspineCurve;
        private List<List<Point>> surfaceCtrlPt = new List<List<Point>>();
```

```
private List<Point[]> BspineSurface = new List<Point[]>();
//参数化与曲线次数
private int v_paraStyle = 2;
private int u_paraStyle = 2;
private int v_k = 2;
private int u_k = 2;
```

（1）初始化与调整：

```
public Form1()//窗口初始化
private void Form1_Load(object sender, EventArgs e) //窗体加载的初始化设置
private void openGLControl1_OpenGLInitialized(object sender, EventArgs e)//OpenGL界面初始化
private void SetLightColor(OpenGL gl) //设置光照的三种反射
private void openGLControl1_Resize(object sender, EventArgs e)//窗口调整事件发生后图形显示变化
private void SetViewDefaultValue()//设置不同视图的摄像机位置变化
private void UpdateViewforSelct(OpenGL gl)//当视图变化时更新窗口上点的坐标
```

（2）绘图循环函数

```
private void openGLControl1_OpenGLDraw(object sender, PaintEventArgs e)//绘图主循环
private void DrawXoZGrids(OpenGL gl)//画 xoz 平面上的网格
private void DrawCoodinate(OpenGL gl, Point ctrlpt)//画中心坐标架
private void DrawOneCoodinate(OpenGL gl, float xPos, float yPos, float zPos, bool isLine)//画出坐标架
private void DrawSphere(OpenGL gl, double radius, int segx, int segy, bool isLines)//画出球体
private void DrawCylinder(OpenGL gl, double height, double radius, string x)//画圆柱体默认
private void CatchPos(OpenGL gl)//获得当前鼠标的对应世界坐标系的位置
private void Pointtrans(ref Point pos, double[] wc_in_Z0)//点的转换
private Point LinePlanelInsection(Point p0, Point vdir, double[] planel = null)//计算直线和平面的交点，po 为某点的位置，vdir 是直线方向，planel 表示要相交平面 double[4]
private void DrawPoint(OpenGL gl) //画控制点
private void DrawCurve(OpenGL gl)//画出样条曲线
private void drawLine(OpenGL gl, float[] pointcolor, Point[] line)//画基本线元的函数
private void DrawCtrlGrid(OpenGL gl)//画出控制网格
private void DrawSurface(OpenGL gl)//画出曲面
private void DrawSmoothPatch(OpenGL gl)//画出光滑的小面片
private void DrawHighLight(OpenGL gl)//高亮显示鼠标附近的控制点
```

（3）事件触发函数

1） 视图切换

```
private void RadialMenu1qianshiview_Click(object sender, EventArgs e)//前视图
```

```
private void RadialMenu1fushishiview_Click(object sender, EventArgs e)//俯视图
private void RadialMenu1toushiview_Click(object sender, EventArgs e)//透视图
private void RadialMenu1zuoshiview_Click(object sender, EventArgs e)//左视图
```

2) 键盘控制摄像机漫游

```
private void openGLControl_KeyDown(object sender, KeyEventArgs e)//WASDQE 控制六个方向
```

3) 图形生成和修改

```
private void BtnClearScreen_Click(object sender, EventArgs e)//曲线重绘按下
private void BtnGenCurve_Click(object sender, EventArgs e)//曲线生成
private void GenChartofbasefunc(int num, List<Point[]> data)//生成基函数
private void btnClearSurface_Click(object sender, EventArgs e)//曲面重绘
private void btnGenSurface_Click(object sender, EventArgs e)//曲面生成
private bool SetParameteOK( int paraStyle, int num_ctrlPts,int flag)//参数化是否设置正确
```

4) 鼠标操作

```
private void openGLControl_MouseDown(object sender, MouseEventArgs e) //鼠标按下
private void openGLControl_MouseMove (object sender, MouseEventArgs e) //鼠标移动
private void openGLControl_MouseUp(object sender, MouseEventArgs e) //鼠标放开
```

5) 修改模式下的一些函数

```
private void switchToPointSelectBtn_MouseDown(object sender, MouseEventArgs e)//点修改模式
private void checkBoxModify_Click(object sender, EventArgs e)//进入修改模式
private void switchToLineSelectBtn_MouseDown(object sender, MouseEventArgs e)//进入线修改
private void ChooseCoodinatetoStrech()//选择座标架哪个方向进行拉伸
private void ModifyCtrlPoints( double delta_mouse)//修改控制点
private void btnCurveModify_Click(object sender, EventArgs e)//曲线修改模式
```

6) 参数化的设置

```
private void btnvUniform_para_Click(object sender, EventArgs e)//v 向设置一般均匀
private void btnv_QriUniform_para_Click(object sender, EventArgs e)//v 向准均匀
private void btnv_SegBezier_para_Click(object sender, EventArgs e)//v 向分段贝塞尔
private void btnv_NoneUniform_para_Click(object sender, EventArgs e)//v 向非均匀
private void btnu_Uniform_para_Click(object sender, EventArgs e)//u 向一般均匀
private void btnu_SegBezier_para_Click(object sender, EventArgs e)//u 向分段贝塞尔
private void btnu_QriUniform_para_Click(object sender, EventArgs e) //u 向准均匀
private void btnu_NoneUniform_para_Click(object sender, EventArgs e)//u 向非均匀
```

7) 其他

```
private double CalLength( Point P0, int X, int Y)//计算欧式距离函数
```

Bspine3D.cs

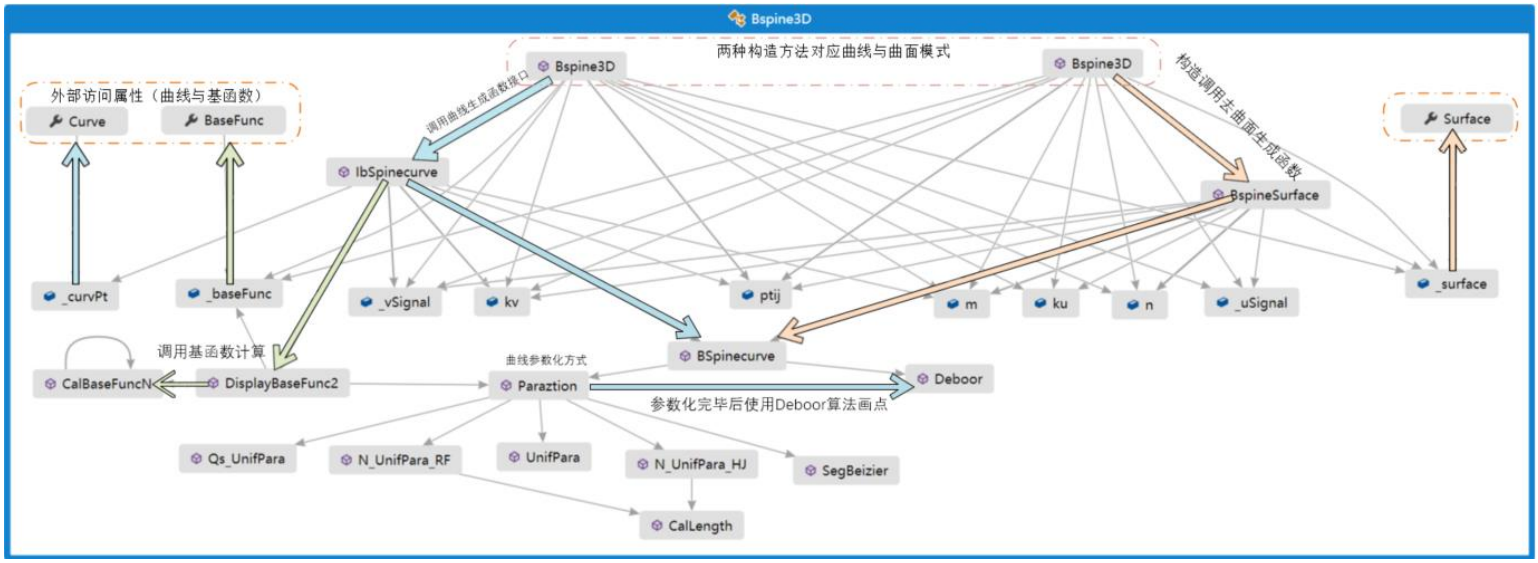


图 4. 函数调用架构图（顶层为构造函数 Bspine3D；构造完成后外部可以访问只读属性 curve，Basefunc 和 Surface；中间层为成员变量；底层为曲线生成方式和参数化方式）

成员变量

```
private readonly Point[,] ptij;           //控制点
private readonly int n, m;                //u向控制点的数量n，v向控制点的数量m
private readonly int ku, kv;              //大写的K代表曲线次数
private readonly int _vSignal;            //v向参数化的方式
private readonly int _uSignal;            //u向参数化的方式
private List<Point[]> _surface= new List<Point[]>();//存储曲面的点的二维矩阵
private Point[] _curvPt;          //存储曲线的点的表
private List<Point[]> _baseFunc = new List<Point[]>();//存储基函数图像的表，每个点列表是一个基函数曲线
```

（1） 构造函数

```
public Bspine3D(Point[,] pij, int m, int n = 1, int ku = 2, int kv = 2, int _vSignal = 1,int _uSignal = 1)//曲面构造函数
public Bspine3D(Point[] pi, int k, int m, int vSignal = 1)//曲线构造函数
```

（2） 参数化函数

```
private float[] Paraztion(int flag, int k, Point[] ctrlPoints)//进行参数化
private void UnifPara(float[] t, int num)//一般均匀
private void Qs_UnifPara(float[] t, int num, int k)//准均匀
private void SegBeizier(float[] t, int num, int k)//分段贝塞尔
private void N_UnifPara_RF(float[] t, int k,Point[] ctrlPoints)// Riesenfield 法
private void N_UnifPara_HJ(float[] t, int k, Point[] ctrlPoints)// Hartley-Judd 法
private float[] CalLength( Point[] ctrlPoints, float[] suml)//计算点的欧式距离
```

（3） 曲线曲面计算函数

```
private Point Deboor(float ti, int k, float[] t, Point[] CtrlPt)//deboor 算法
private double CalBaseFuncN(float ti, int k, int i, float[] t)//计算基函数的 N_ik
public void DisplayBaseFunc2(int k, Point[] ctrlPoint, int sig_para)//计算基函数图像
```

```
private void IbSpinecurve()//b 样条曲线接口
private Point[] BSpinecurve(int order, int n_Segs, Point[] ctrlPoint, int sig_para)//B 样条曲线计算函数
private void BspineSurface() //B 样条曲面计算函数
```

5. 程序使用说明

5.1 在曲线界面绘出 v 向曲线

a) 选择参数划分方式和次数，默认为准均匀参数划分，k=3

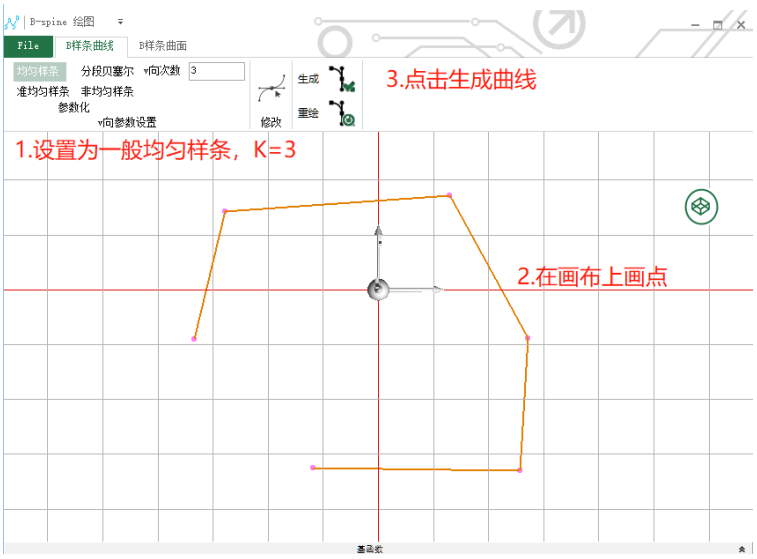


图 5. 画出控制点

b) 可以直接点击图形区域生成控制点，点击生成，打开下方基函数图像显示

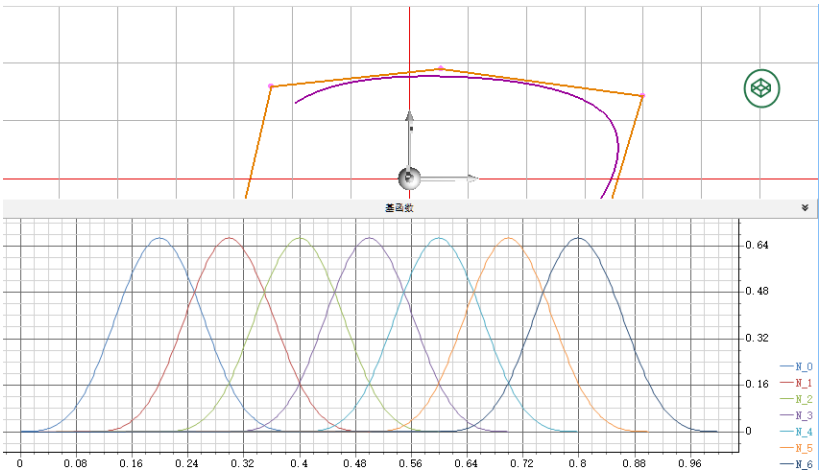


图 6. 点击生成曲线与基函数

5.2 修改控制点

a）点击曲线的修改按键，鼠标移动到控制点附近生成坐标架

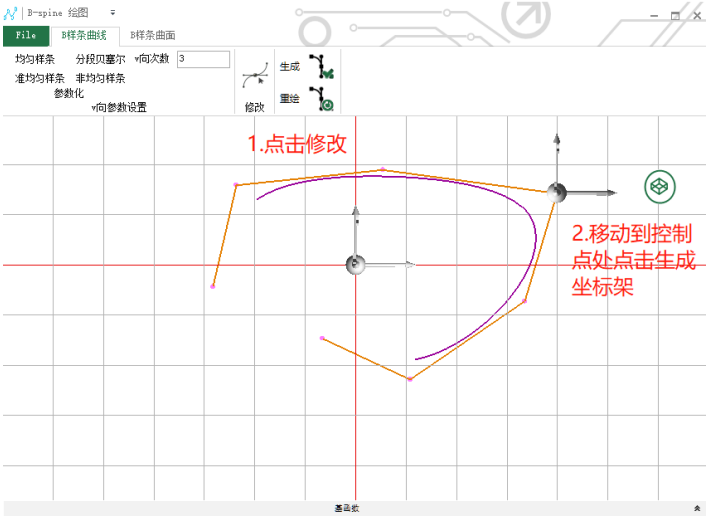


图 7. 修改控制点

b)拖拽某一坐标轴进行该轴方向上的修改，可以双击上方的曲线曲面菜单栏，收起工具栏

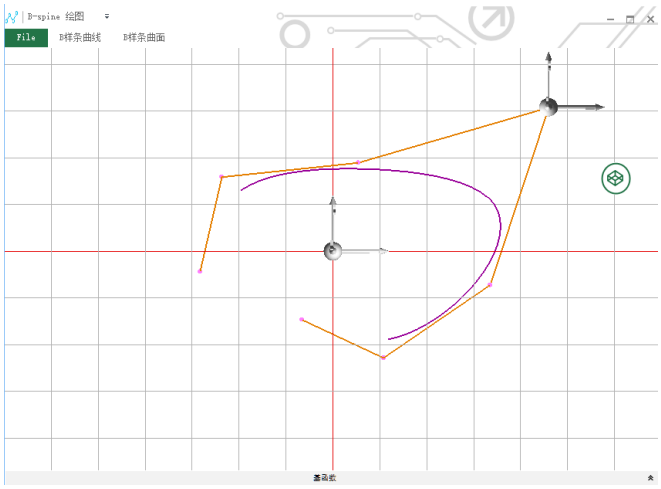


图 8. 拖拽坐标轴以修改控制点

c）键盘上的 QWEASD 按键可以控制视角，进行视角漫游，

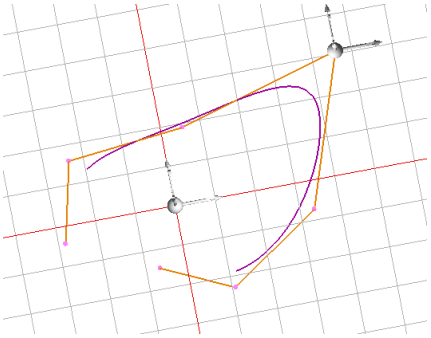


图 9. 变化视角

5.3 在曲面界面生成想要的曲面

确认在曲线界面完成 v 向曲线设置后，可以设置 u 向的曲面设置

a) 点击右边的圆形视图控制按钮，切换到透视图，右键拖动旋转;打开修改模式下的线修改模式

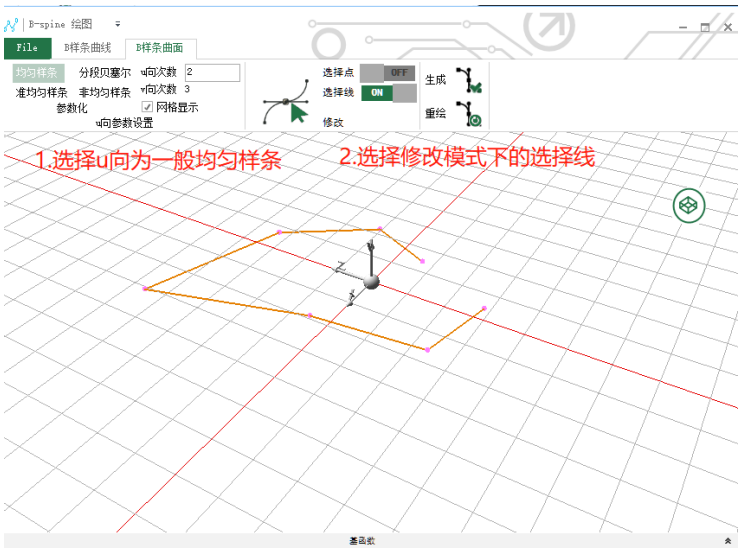


图 10. 切换视图设置参数

b) 同时按住键盘上的 ctrl 按键和鼠标左键拖动生成曲面网格

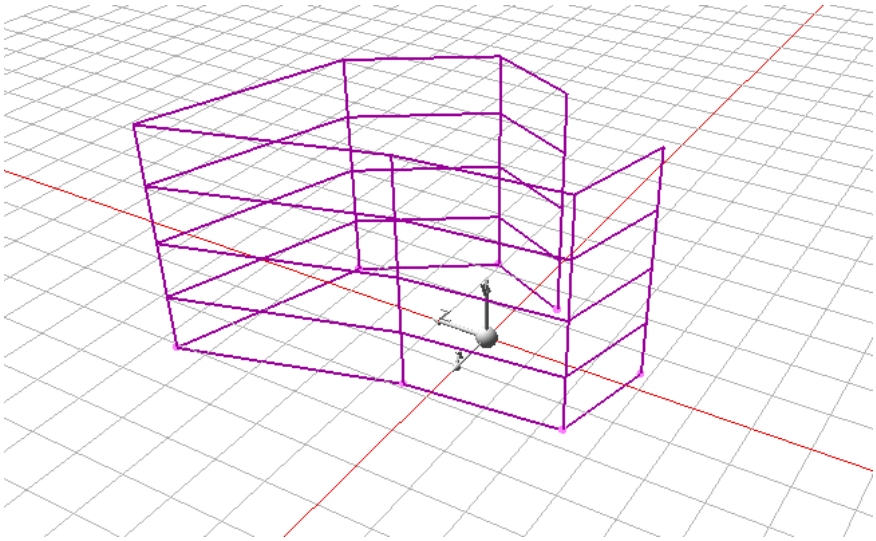


图 11. 拖拽生成控制网格

c) 点击生成生成曲面

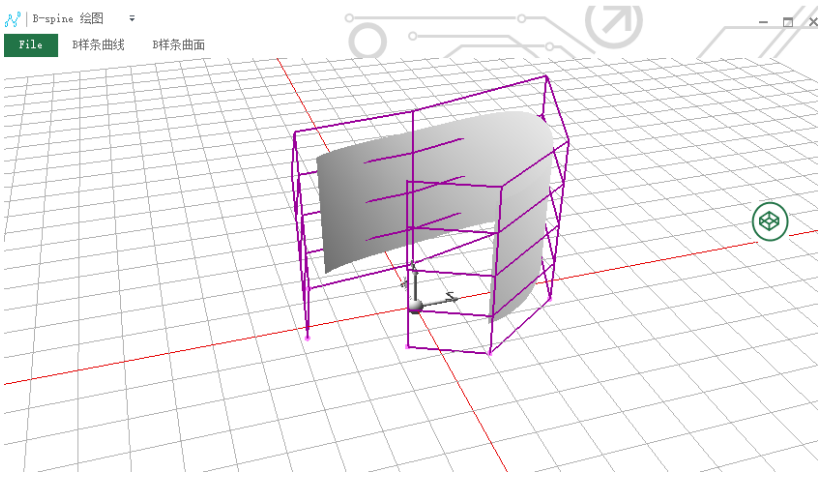
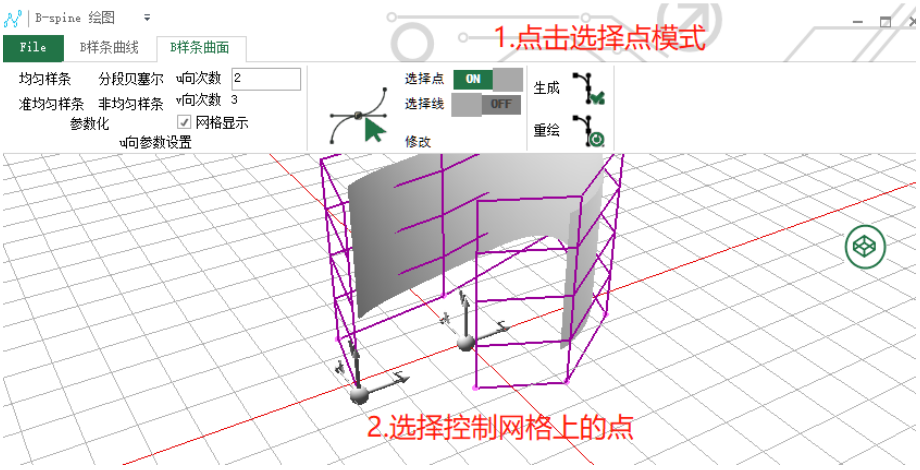


图 11. 生成 B 样条曲面

5.4 修改控制网格

a） 打开修改模式下的点选择模式



b） 按住鼠标中键拖动放大，鼠标点击屏幕上的控制网格的控制点，生成座标架;拖动座标架修改控制网格