

## 🔗 8. 组合曲线CurvePath拼接曲线

通过threejs组合曲线 `CurvePath` 对象，你可以把直线、圆弧、贝塞尔等线条拼接为一条曲线。

课件案例源码为大家提供了一个圆弧和直线组合拼接的U形效果。

### 直线线段简介

本节课用到的圆弧前面讲解过，下面给大家说下直线相关API，three.js提供了3D直线 `LineCurve3` 和2D直线 `LineCurve`。

3D直线线段 `LineCurve3`，参数是表示x、y、z坐标的三维向量 `Vector3` 对象。

```
new THREE.LineCurve3(new THREE.Vector3(), new THREE.Vector3());
```

js

2D直线线段 `LineCurve`，参数是表示x、y坐标的二维向量 `Vector2` 对象。

```
new THREE.LineCurve(new THREE.Vector2(), new THREE.Vector2());
```

js

### 整体思路 `CurvePath.curves`

整体思路很简单，`LineCurve` 创建两条直线线段，`ArcCurve` 绘制一段圆弧线，然后把两段直线和一段圆弧线，通过组合曲线的 `CurvePath.curves` 属性拼接起来。

```
const R = 80; // 圆弧半径
const H = 200; // 直线部分高度
// 直线1
const line1 = new THREE.LineCurve(new THREE.Vector2(R, H), new THREE.Vector2(R, 0));
// 圆弧
const arc = new THREE.ArcCurve(0, 0, R, 0, Math.PI, true);
// 直线2
const line2 = new THREE.LineCurve(new THREE.Vector2(-R, 0), new THREE.Vector2(-R, H));
```

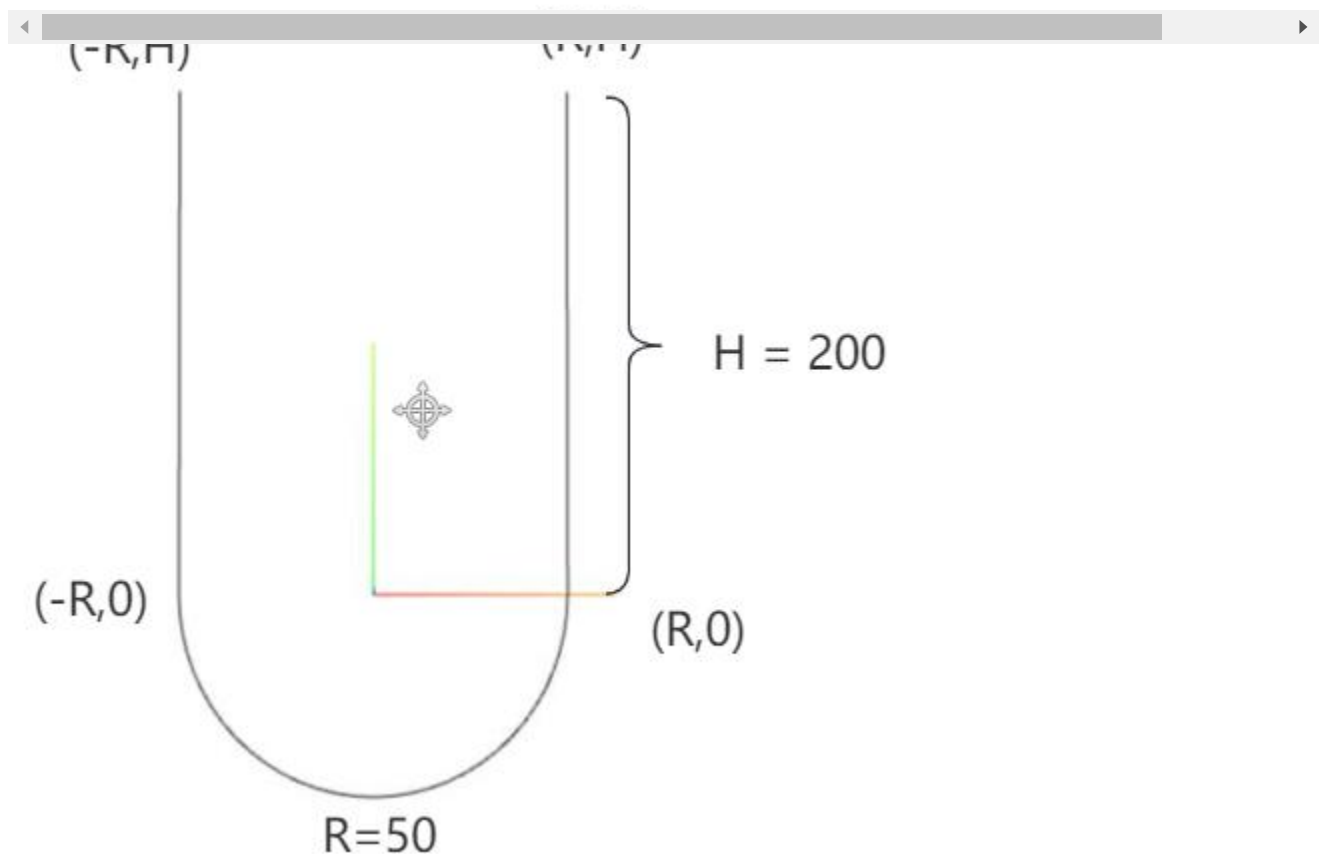
js

```
// CurvePath创建一个组合曲线对象
const CurvePath = new THREE.CurvePath();
//line1, arc, line2拼接出来一个U型轮廓曲线，注意顺序
CurvePath.curves.push(line1, arc, line2);
```

## 注意：曲线首尾相接

有一点要注意，组合曲线的坐标顺序和线条组合顺序不能随意写，总的方向，就是先确定整个曲线的起点，然后沿着一个方向依次绘制不同曲线，确保不同曲线首尾相接。

- 直线的起点是直线的第一个参数
- 圆弧线的起点，默认就是从x轴正半轴开始



```
const R = 80; // 圆弧半径
const H = 200; // 直线部分高度
// 直线1
const line1 = new THREE.LineCurve(new THREE.Vector2(R, H), new THREE.Vector2(R, 0));
// 圆弧
const arc = new THREE.ArcCurve(0, 0, R, 0, Math.PI, true);
// 直线2
const line2 = new THREE.LineCurve(new THREE.Vector2(-R, 0), new THREE.Vector2(-R, H));
```

```
// CurvePath创建一个组合曲线对象
const CurvePath = new THREE.CurvePath();
//line1, arc, line2拼接出来一个U型轮廓曲线，注意顺序
CurvePath.curves.push(line1, arc, line2);
```

## 组合曲线 CurvePath 取点

组合曲线 `CurvePath` 和它的父类`Curve`一样具有 `.getPoints()` 和 `.getSpacedPoints()` 取点方法。

```
//组合曲线上获取点
const pointsArr = CurvePath.getPoints(16);
const geometry = new THREE.BufferGeometry();
geometry.setFromPoints(pointsArr); //读取坐标数据赋值给几何体顶点
```

js

## 点模型可视化组合曲线返回的顶点

执行 `.getPoints()` ，直线部分不会像曲线返回中间多余点，只需要起始点即可。

```
// 执行.getPoints(), 直线部分不会像曲线返回中间多余点，只需要起始点即可。
const pointsArr = CurvePath.getPoints(16); //曲线上获取点
const geometry = new THREE.BufferGeometry();
geometry.setFromPoints(pointsArr); //读取坐标数据赋值给几何体顶点

// 可视化curve.getPoints从曲线上获取的点坐标
const material2 = new THREE.PointsMaterial({
  color: 0xff00ff,
  size: 10,
});
//点模型对象
const points = new THREE.Points(geometry, material2);
```

js

## `.getSpacedPoints()` 取点测试

复习前面知识，你可以通过测试曲线 `Curve` 方法 `.getSpacedPoints()` 从组合曲线获取点坐标。

这时候你会发现直线部分会按照等间距方式返回顶点数据，需要把 `.getSpacedPoints()` 的精度参数提升，圆弧部分才会更加圆滑。

```
const pointsArr = CurvePath.getSpacedPoints(16); //圆弧不够光滑
```

js

---

← [7. 样条、贝塞尔曲线应用](#)

[9. 曲线路径管道TubeGeometry](#) →