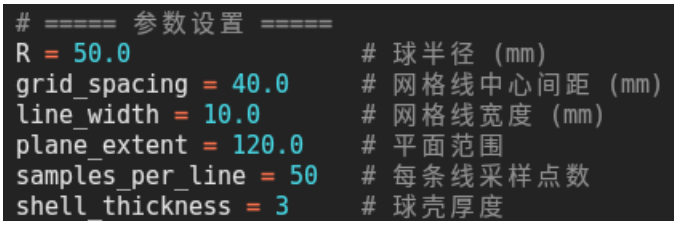
平面网格的球极投影模型构建

核心：构建球面坐标与平面网格坐标间的对应关系；

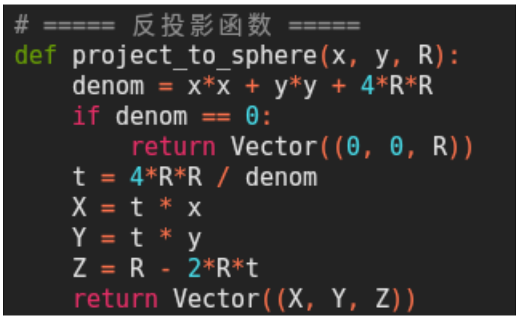
选择合适的平面网格坐标带入关系。

模型参数选择：



```
# ===== 参数设置 =====
R = 50.0                  # 球半径 (mm)
grid_spacing = 40.0       # 网格线中心间距 (mm)
line_width = 10.0         # 网格线宽度 (mm)
plane_extent = 120.0      # 平面范围
samples_per_line = 50     # 每条线采样点数
shell_thickness = 3       # 球壳厚度
```

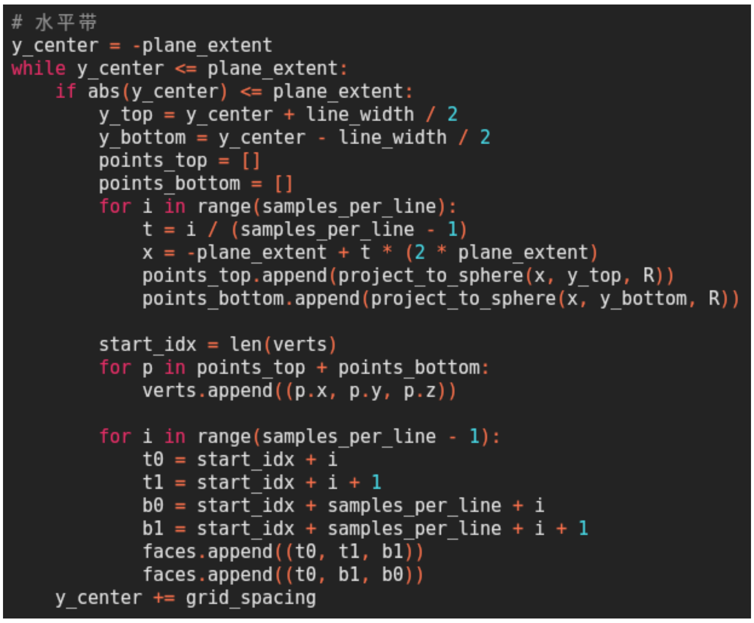
映射函数构建（该模型选取平面 $z = -R$ 为投影平面）：



```
# ===== 反投影函数 =====
def project_to_sphere(x, y, R):
    denom = x*x + y*y + 4*R*R
    if denom == 0:
        return Vector((0, 0, R))
    t = 4*R*R / denom
    X = t * x
    Y = t * y
    Z = R - 2*R*t
    return Vector((X, Y, Z))
```

对模型效果观察反思，投影平面的选取应当和模型间隔一定距离来使得投影效果更好。

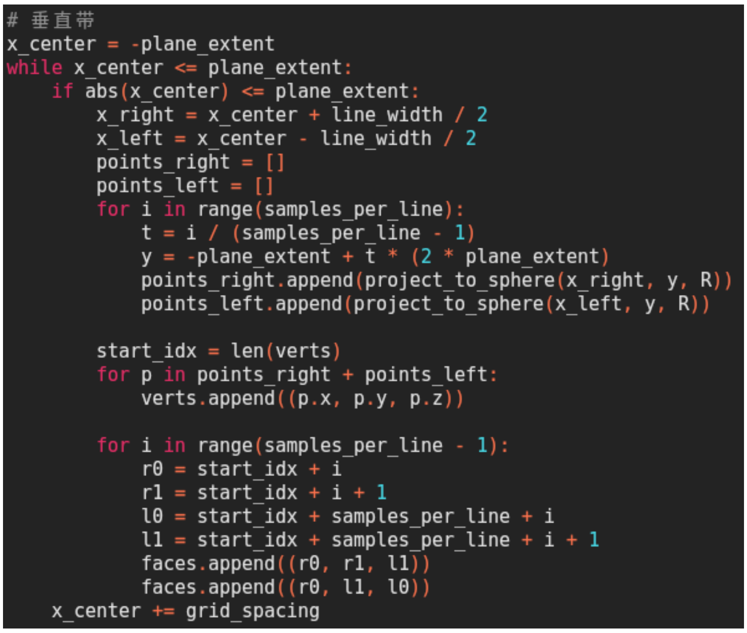
平面网格选取：



```
# 水平带
y_center = -plane_extent
while y_center <= plane_extent:
    if abs(y_center) <= plane_extent:
        y_top = y_center + line_width / 2
        y_bottom = y_center - line_width / 2
        points_top = []
        points_bottom = []
        for i in range(samples_per_line):
            t = i / (samples_per_line - 1)
            x = -plane_extent + t * (2 * plane_extent)
            points_top.append(project_to_sphere(x, y_top, R))
            points_bottom.append(project_to_sphere(x, y_bottom, R))

        start_idx = len(verts)
        for p in points_top + points_bottom:
            verts.append((p.x, p.y, p.z))

        for i in range(samples_per_line - 1):
            t0 = start_idx + i
            t1 = start_idx + i + 1
            b0 = start_idx + samples_per_line + i
            b1 = start_idx + samples_per_line + i + 1
            faces.append((t0, t1, b1))
            faces.append((t0, b1, b0))
    y_center += grid_spacing
```



```
# 垂直带
x_center = -plane_extent
while x_center <= plane_extent:
    if abs(x_center) <= plane_extent:
        x_right = x_center + line_width / 2
        x_left = x_center - line_width / 2
        points_right = []
        points_left = []
        for i in range(samples_per_line):
            t = i / (samples_per_line - 1)
            y = -plane_extent + t * (2 * plane_extent)
            points_right.append(project_to_sphere(x_right, y, R))
            points_left.append(project_to_sphere(x_left, y, R))

        start_idx = len(verts)
        for p in points_right + points_left:
            verts.append((p.x, p.y, p.z))

        for i in range(samples_per_line - 1):
            r0 = start_idx + i
            r1 = start_idx + i + 1
            l0 = start_idx + samples_per_line + i
            l1 = start_idx + samples_per_line + i + 1
            faces.append((r0, r1, l1))
            faces.append((r0, l1, l0))
    x_center += grid_spacing
```

分别选取水平与垂直条带，该方式存在优化空间：水平与垂直重叠部分的打印结果会略微加厚

模型预览：