

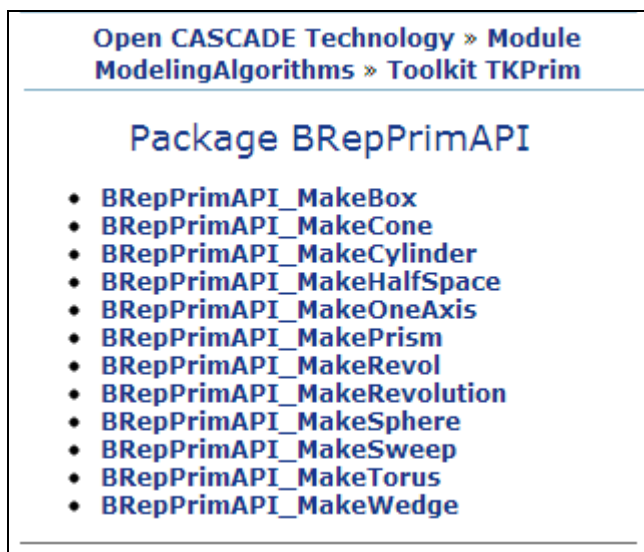
在 Open Cascade 中创建基本几何实体

Construction of Primitives in Open Cascade

eryar@163.com

一、创建基本图元 *Making Primitives*

将用到如下的类创建基本图元，包括长方体(**Box**)、楔形体(**Wedge**)、旋转体(**Revol**)等。这些类提供 *Shell* 和 *Solid* 函数来返回实体 (*solid*) 和壳 (*shell*)。

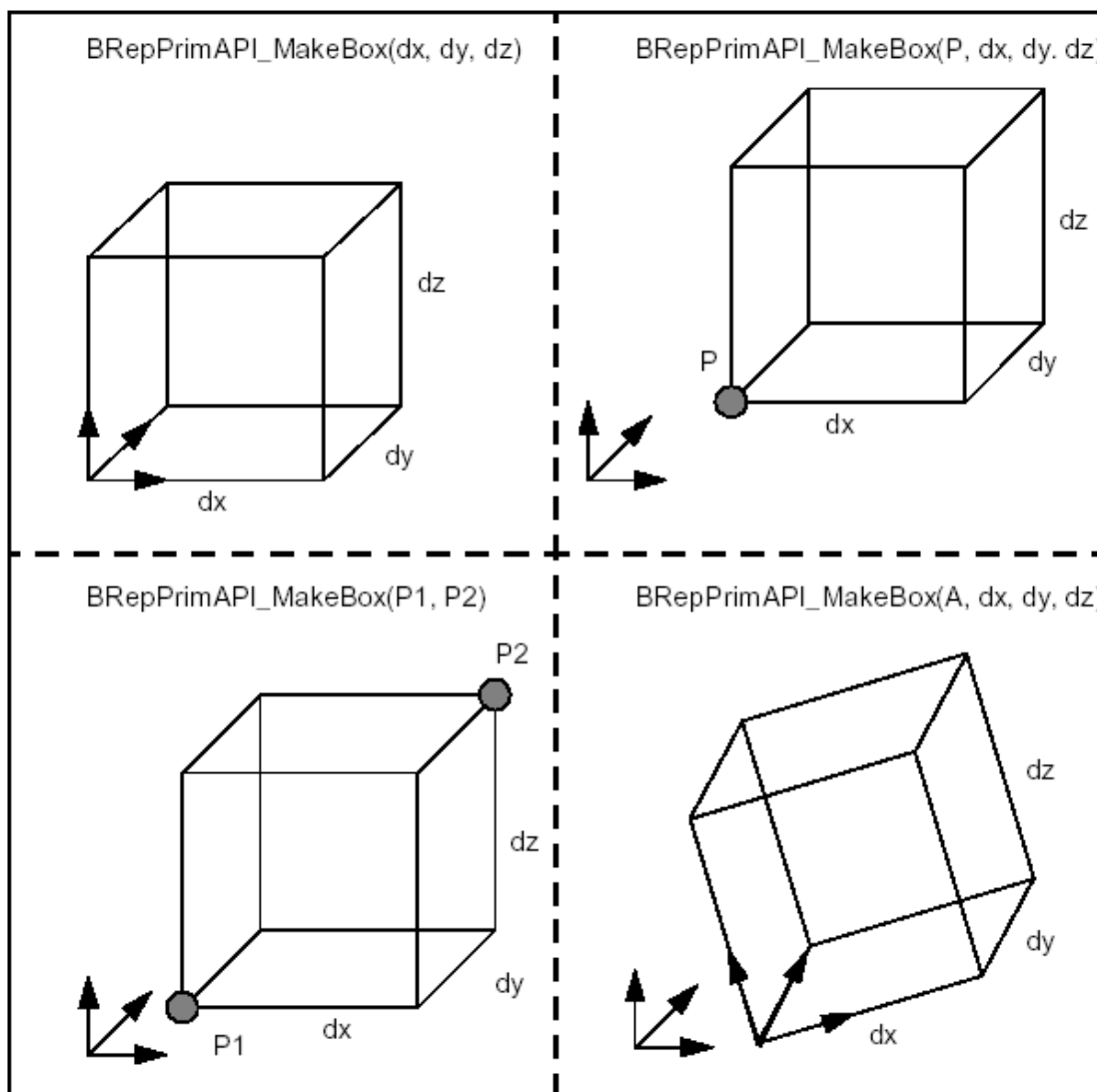


使用方法都是调用这些类的构造函数，输入相应的参数，直接构造出基本图元。构造函数的参数不同，创建的基本几何实体的形状也会不同。

注：所有旋转体的角度值都是弧度值。

1. 创建长方体 *BRepPrimAPI_MakeBox*

使用 *MakeBox* 类创建一个长方体，结果以一个壳或体返回。创建长方体的方式有如下四种：



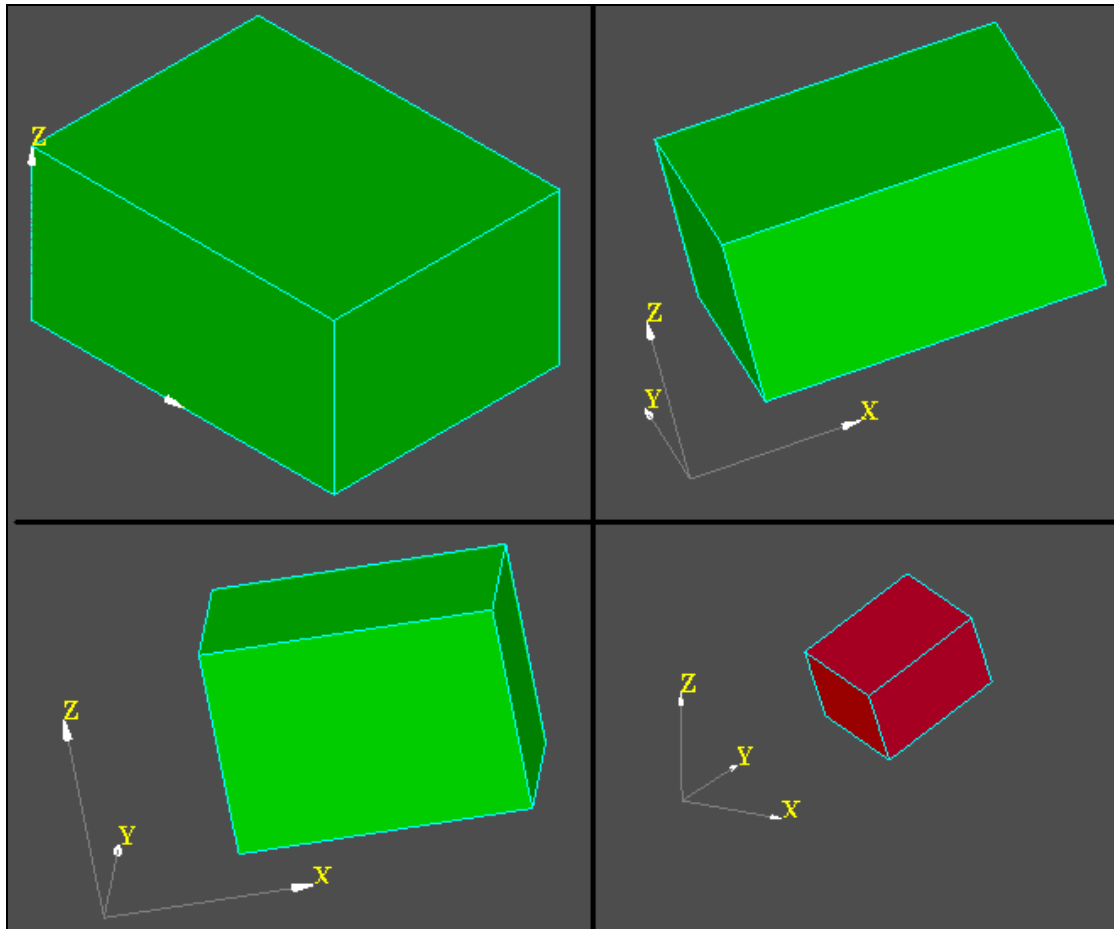
以上创建方式对应类的 *BRepPrimAPI_MakeBox* 的四种构造函数：

```
Standard_EXPORT BRepPrimAPI_MakeBox(const Standard_Real dx, const Standard_Real dy, const Standard_Real dz);
```

```
Standard_EXPORT BRepPrimAPI_MakeBox(const gp_Pnt& P, const Standard_Real dx, const Standard_Real dy, const Standard_Real dz);
```

```
Standard_EXPORT BRepPrimAPI_MakeBox(const gp_Pnt& P1, const gp_Pnt& P2);
```

```
Standard_EXPORT BRepPrimAPI_MakeBox(const gp_Ax2& Axes, const Standard_Real dx, const Standard_Real dy, const Standard_Real dz);
```



与上图对应的创建长方体的代码如下所示：

```
BRepPrimAPI_MakeBox (200. , 150. , 100.);  
BRepPrimAPI_MakeBox (gp_Pnt(60, 60, 0), 200. , 150. , 100.);  
BRepPrimAPI_MakeBox (gp_Pnt(60, 60, 0), gp_Pnt(200. , 150. , 100.));  
BRepPrimAPI_MakeBox (gp_Ax2(gp_Pnt(100. , 80. , 70.), gp_Dir(1. , 2. , 1.)), 80. , 90. , 120.);
```

2. 创建楔形体 *BRepPrimAPI_MakeWedge*

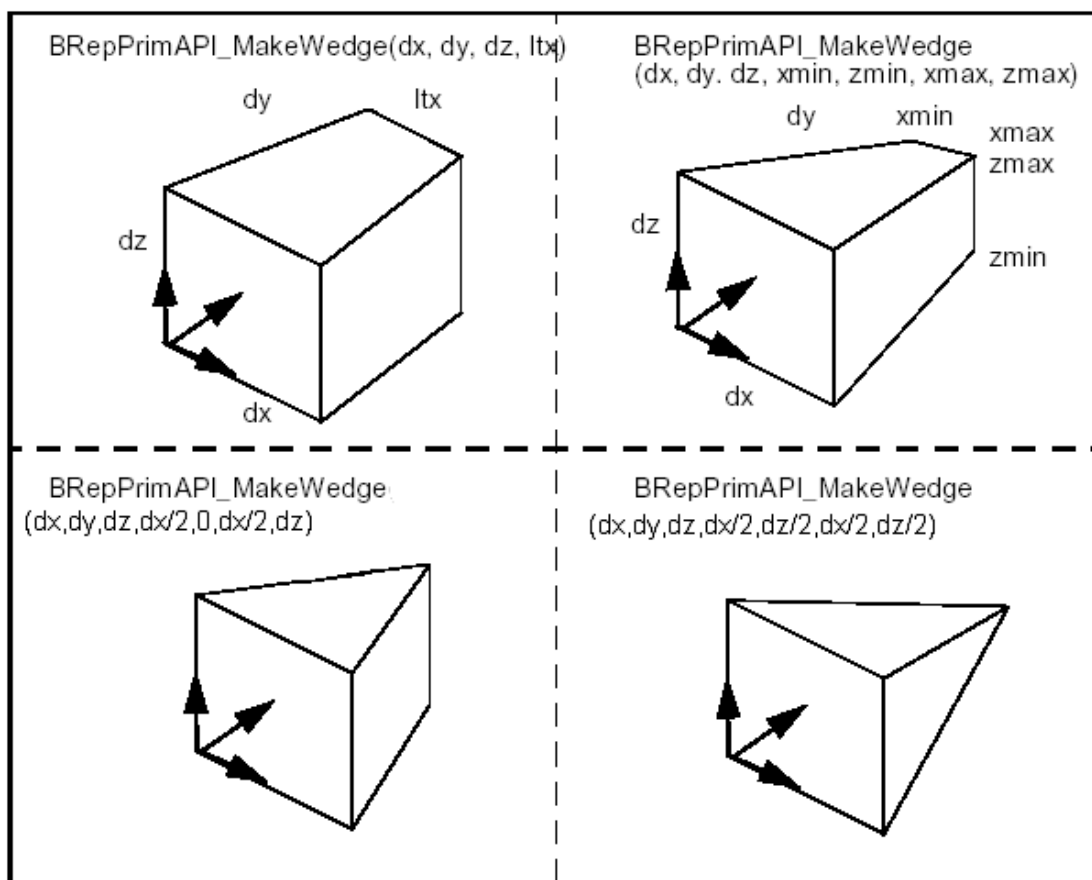
使用类 *BRepPrimAPI_MakeWedge* 来创建楔形体。楔形体就是带斜面的长方体，即带角度的长方体。楔形体的创建方式与长方体的创建方式相似。下图所示为构造楔形体的两种方式。一种是加入 *ltx* 尺寸来描述 *dy* 上的面，另一种是加入 *xmin*, *xmax*, *zmin*, *zmax* 来描述 *dy* 上的面。

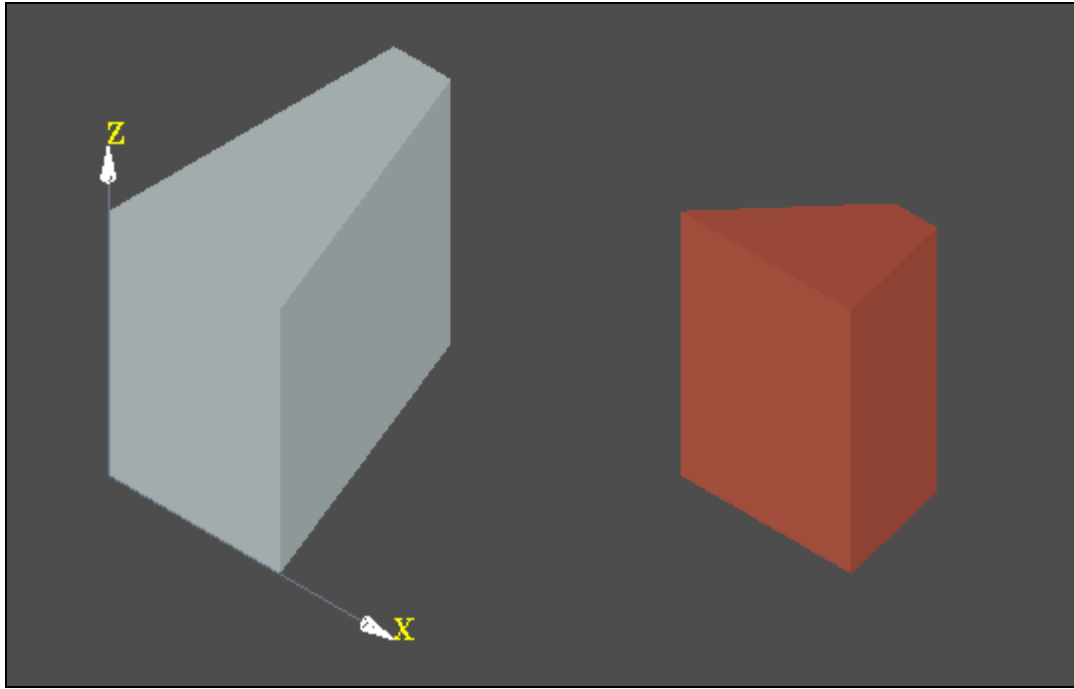
第一种方式是第二种方式在满足以下条件下的特例：

$$xmin = 0, xmax = ltx, zmin = 0, zmax = dz$$

创建一个金字塔的楔形体可以用下面的方式：

$$xmin = xmax = dx / 2, zmin = zmax = dz / 2$$





生成上图所示的程序如下所示:

```
TopoDS_Shape S1 = BRepPrimAPI_MakeWedge(60. , 100. , 80. , 20. );  
TopoDS_Shape S2 =  
BRepPrimAPI_MakeWedge(gp_Ax2(gp_Pnt(100. , 100. , 0. ), gp_Dir(0. , 0. , 1. )),  
60. , 50. , 80. , 25. , - 10. , 40. , 70. );
```

3. *BRepPrimAPI_MakeOneAxis*

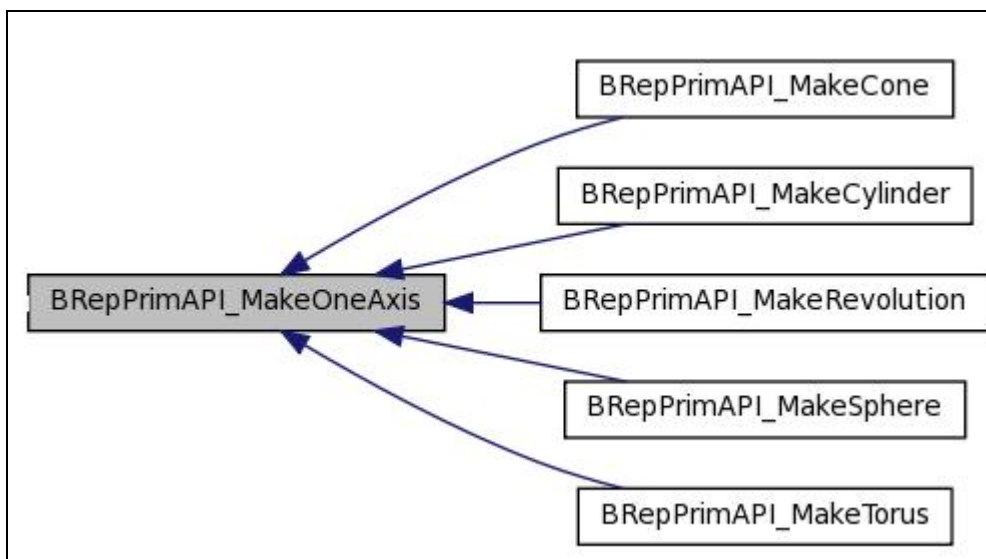
类 *BRepPrimAPI_MakeOneAxis* 是所有旋转体图元的基类。旋转体通过将曲线绕一个轴旋转得到。它们有圆柱体、圆锥体、球体、圆环体等。

这些旋转体对于不同的图元有不同的构造参数，但是它们也有共同的参数：

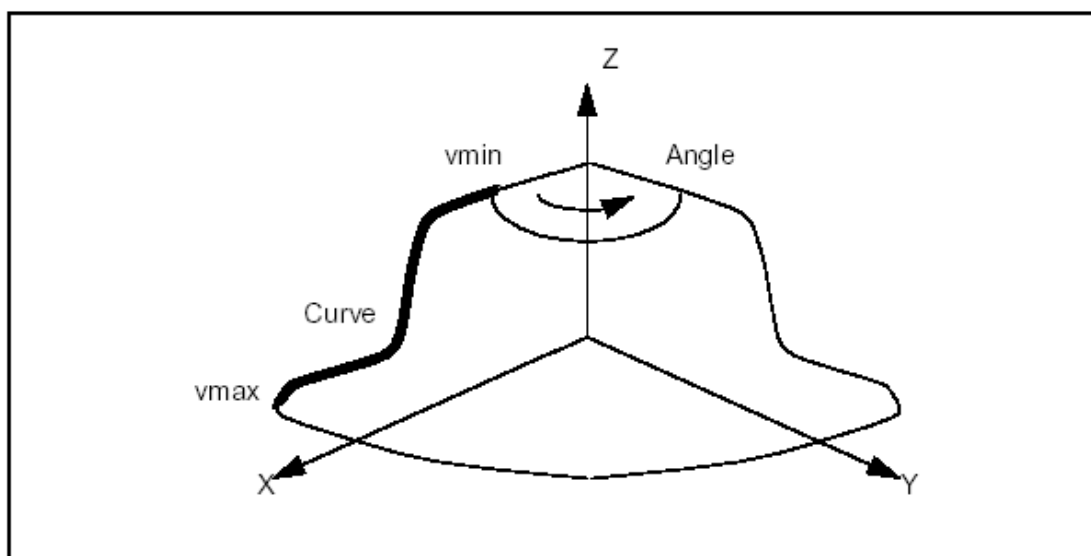
- l 一个坐标系；
- l 一个角度，范围为 $[0, 2*PI]$ ；
- l 曲线的 *vmin*, *vmax* 参数；

OneAxis 构造的结果是体 *Solid*、壳 *Shell* 或面 *Face*。

通常不会直接使用类 *BRepPrimAPI_MakeOneAxis*，而是使用它的派生类。



如下图所示描述了类的参数：



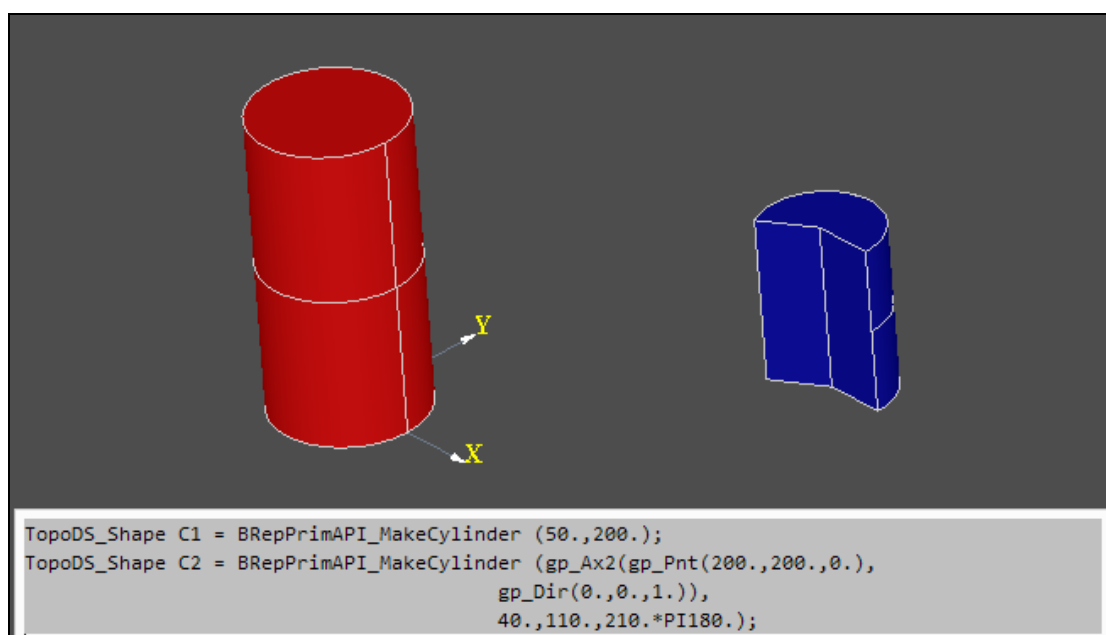
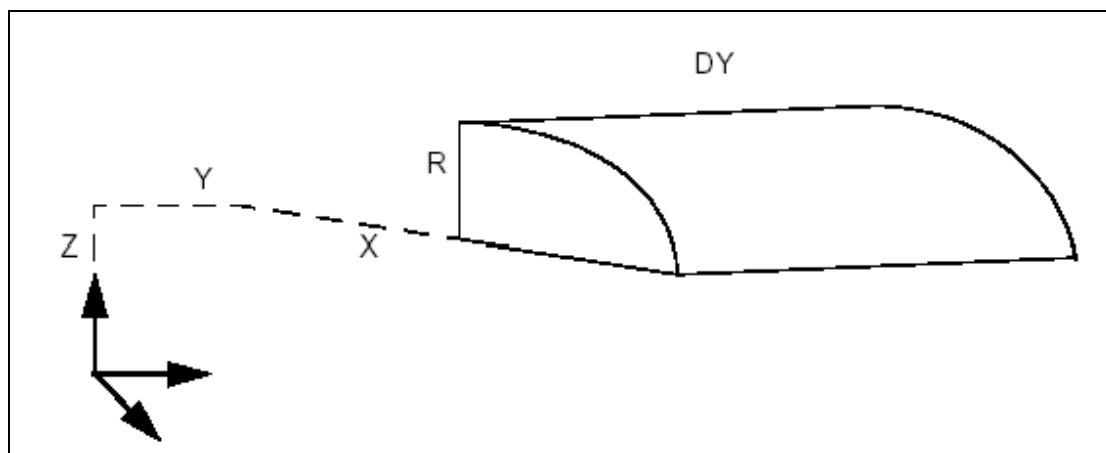
4. 创建圆柱体 *BRepPrimAPI_MakeCylinder*

使用类 *BRepPrimAPI_MakeCylinder* 来创建圆柱体。圆柱可以在默认的坐标系中被创建，也可以指定的坐标系 (*gp_Ax2*) 中创建。创建方式如下：

- ┆ 半径、高度创建一个完整的圆柱；
- ┆ 半径、高度及角度来创建部分圆柱；

如下代码所示为沿 *X*、*Y*、*Z* 偏移一定距离的四分之一的圆柱：

```
Standard_Real X = 20;
Standard_Real Y = 10;
Standard_Real Z = 15;
Standard_Real R = 10;
Standard_Real DY = 30;
// Make the system of coordinates.
gp_Ax2 axes = gp::ZOX();
axes.Translate(gp_Vec(X, Y, Z));
TopoDS_Face F = BRepPrimAPI_MakeCylinder(axes, R, DY, PI/2.);
```



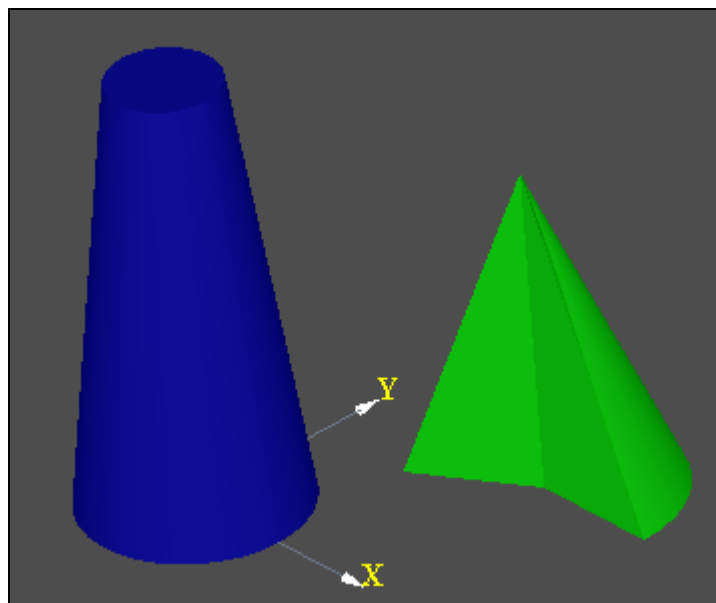
5. 创建圆锥体 *BRepPrimAPI_MakeCone*

使用类 *BRepPrimAPI_MakeCone* 来创建圆锥体。与圆柱体一样，圆锥体可以在默认的坐标系中被创建，也可以指定的坐标系中被创建。创建方式如下：

- I 两个半径和高度来创建一个完整的圆锥。一个半径可以为零，这样就创建了一个尖顶的圆锥；
- I 半径、高度和角度来创建部分圆锥。

如下代码创建的圆锥如下图所示：

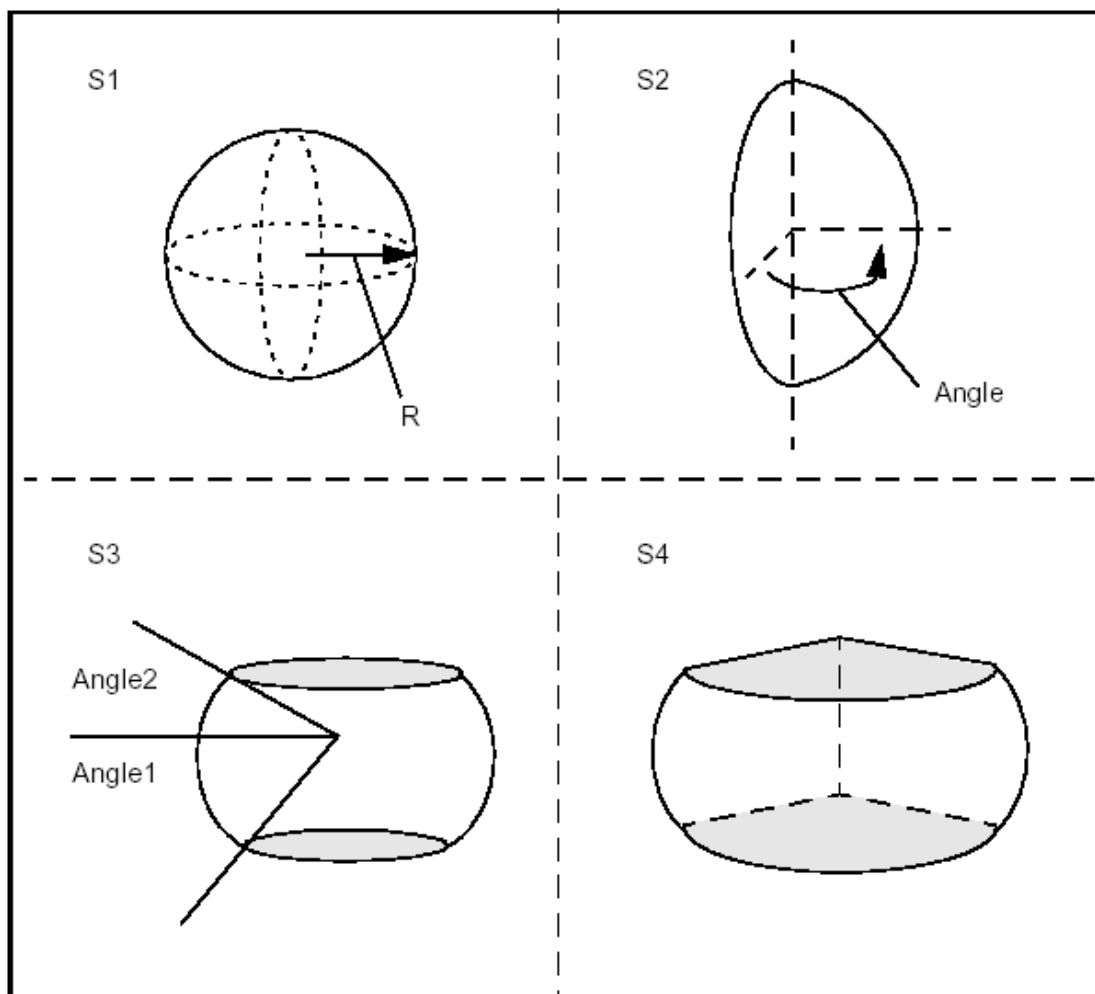
```
TopoDS_Shape C1 = BRepPrimAPI_MakeCone ( 50. , 25. , 200. );  
TopoDS_Shape C2 = BRepPrimAPI_MakeCone(gp_Ax2(gp_Pnt( 100. , 100. , 0. ),  
gp_Dir(0. , 0. , 1. )),  
605. , 0. , 150. , 210. *PI/180);
```



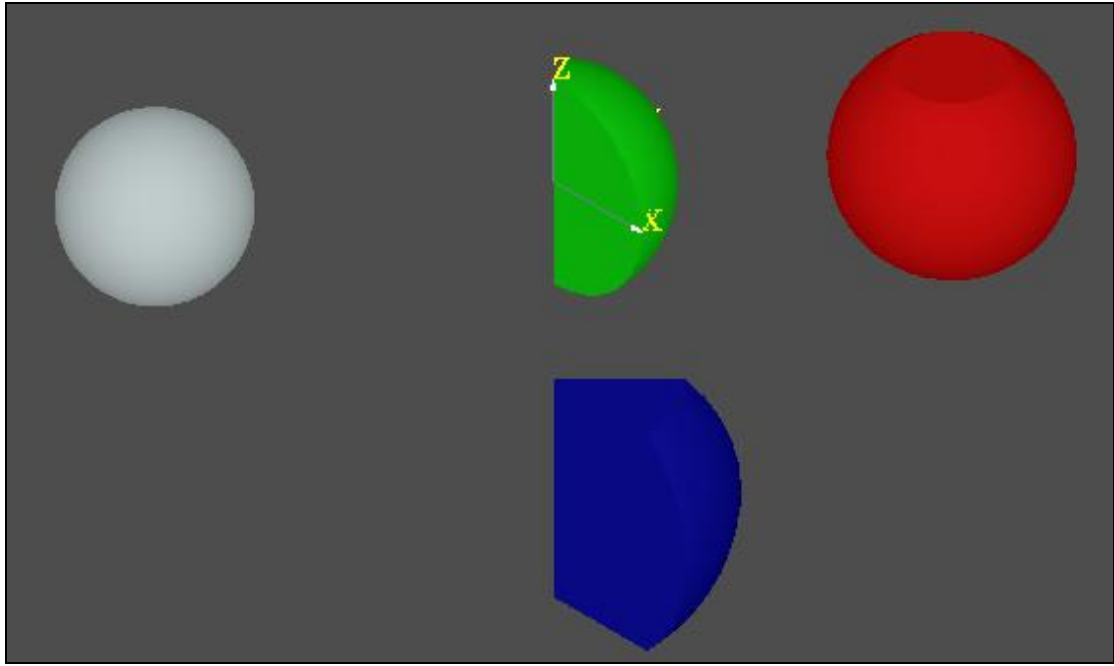
6. 创建球体 *BRepPrimAPI_MakeSphere*

使用类 *BRepPrimAPI_MakeSphere* 来创建球体。与圆柱体一样，球体可以在默认的坐标系中被创建，也可以指定的坐标系中被创建。有四种创建方式，如下图所示：

- I 一个半径值，创建一个完整的球体；
- I 半径和角度，创建部分的球体；
- I 半径和两个角度值，创建纬度上的部分的球体。两个角度 $a1$, $a2$ 必须满足如下条件： $PI/2 \leq a1 < a2 \leq PI/2$
- I 一个半径和三个角度值；

**Example**

```
Standard_Real R = 30, ang =
    PI/2, a1 = -PI/2.3, a2 = PI/4;
TopoDS_Solid S1 = BRepPrimAPI_MakeSphere(R);
TopoDS_Solid S2 = BRepPrimAPI_MakeSphere(R,ang);
TopoDS_Solid S3 = BRepPrimAPI_MakeSphere(R,a1,a2);
TopoDS_Solid S4 = BRepPrimAPI_MakeSphere(R,a1,a2,ang);
```



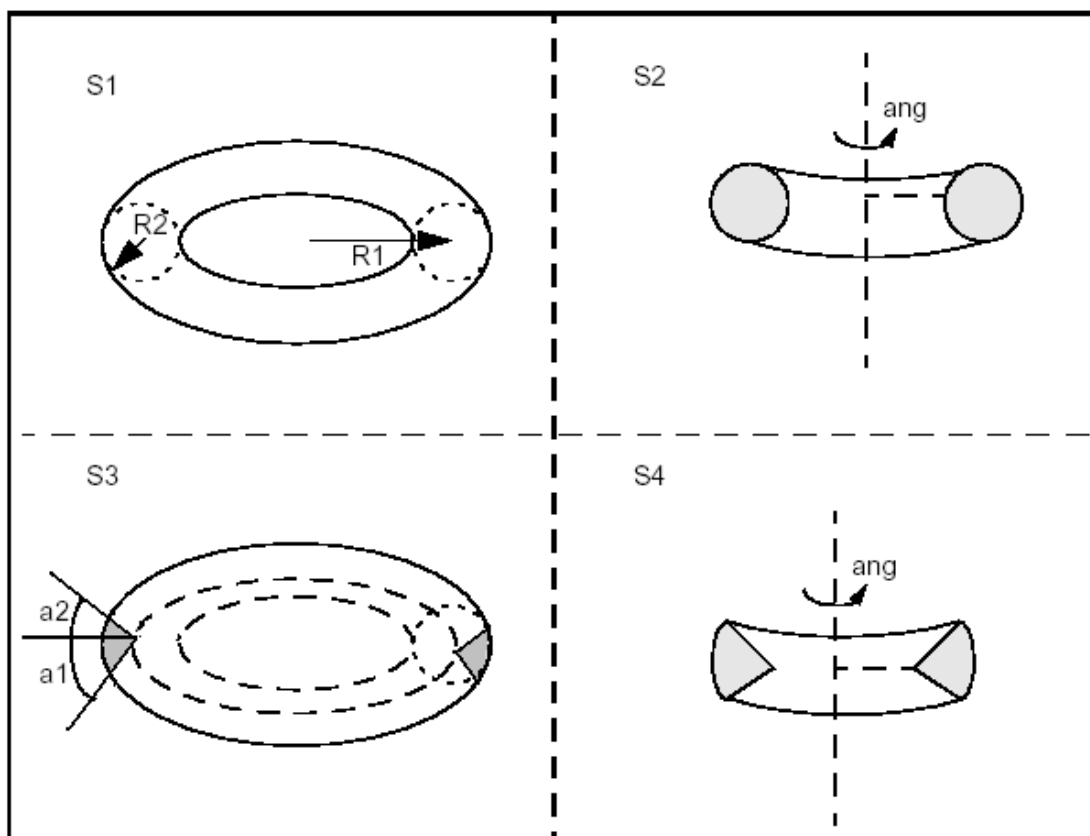
实现上图中球体的代码如下：

```
TopoDS_Shape S1 = BRepPrimAPI_MakeSphere(gp_Pnt(-200., -250., 0.), 80.);  
TopoDS_Shape S2 = BRepPrimAPI_MakeSphere(100., 120.*PI/180);  
TopoDS_Shape S3 = BRepPrimAPI_MakeSphere(gp_Pnt(200., 250., 0.), 100.,  
- 60.*PI/180, 60.*PI/180);  
TopoDS_Shape S4 = BRepPrimAPI_MakeSphere(gp_Pnt(0., 0., -300.), 150.,  
- 45.*PI/180, 45.*PI/180, 45.*PI/180);
```

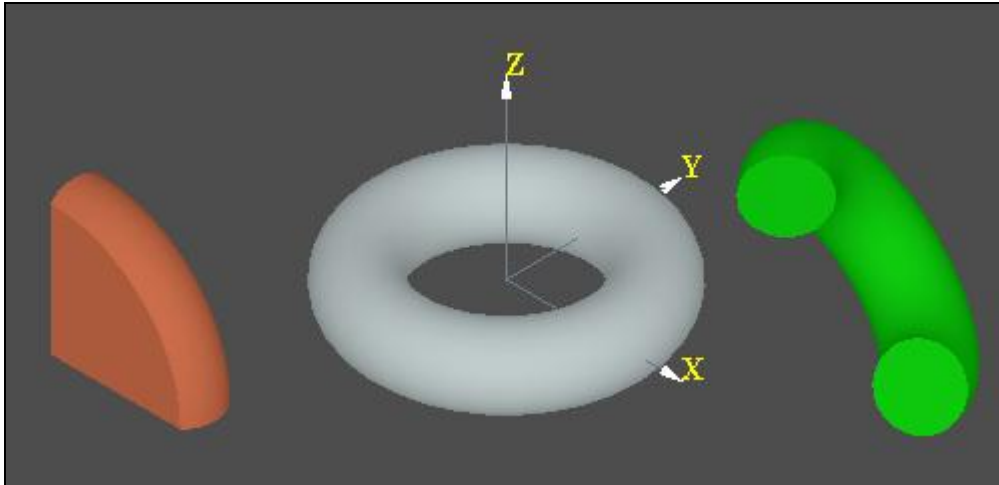
7. 创建圆环体 *BRepPrimAPI_MakeTorus*

使用类 *BRepPrimAPI_MakeTorus* 来创建圆环体。与其它基本图元类似，一个圆环体可以在默认的坐标系中创建，也可以在指定的坐标系中创建。创建与与球体创建方式类似：

- l 两个半径，创建一个完整的圆环体；
- l 两个半径和一个角度值，创建部分的圆环体；
- l 两个半径和两个角度值，创建纬度上的部分圆环体；两个角度必须满足如下条件：
 $0 < a2 - a1 < 2 * PI$ ；
- l 两个半径和三个角度值，创建部分的圆环体；

**Example**

```
Standard_Real R1 = 30, R2 = 10, ang = PI, a1 = 0,
              a2 = PI/2;
TopoDS_Shell S1 = BRepPrimAPI_MakeTorus(R1,R2);
TopoDS_Shell S2 = BRepPrimAPI_MakeTorus(R1,R2,ang);
TopoDS_Shell S3 = BRepPrimAPI_MakeTorus(R1,R2,a1,a2);
TopoDS_Shell S4 =
    BRepPrimAPI_MakeTorus(R1,R2,a1,a2,ang);
```



```
TopoDS_Shape S1 = BRepPrimAPI_MakeTorus(60. , 20. );  
TopoDS_Shape S2 =  
BRepPrimAPI_MakeTorus(gp_Ax2(gp_Pnt(100. , 100. , 0. ), gp_Dir(1. , 1. , 1. )),  
50. , 20. , 210. *PI/180);  
TopoDS_Shape S3 =  
BRepPrimAPI_MakeTorus(gp_Ax2(gp_Pnt(-200. , -150. , -100), gp_Dir(0. , 1. , 0. )),  
60. , 20. , -45. *PI/180, 45. *PI/180, 90. *PI/180);
```

8. 创建旋转体 *BRepPrimAPI_MakeRevolution*

使用类 *BRepPrimAPI_MakeRevolution* 来根据一条曲线和轴创建旋转体。曲线可以是任意的 *Geom_Curve*，有四种构造方式：

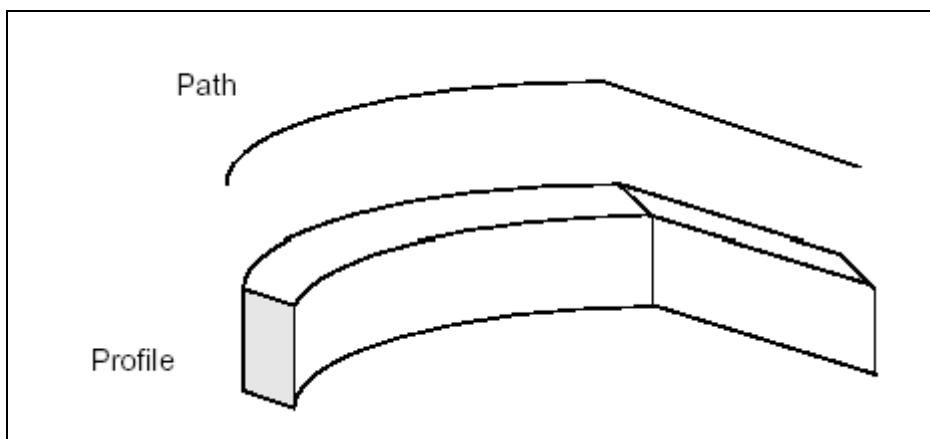
- | *From a curve, use the full curve and make a full rotation;*
- | *From a curve and an angle of rotation;*
- | *From a curve and two parameters to trim the curve. The two parameters must be growing and within the curve range;*
- | *From a curve, two parameters, and an angle. The two parameters must be growing and within the curve range.*

二、创建扫掠体 *Sweeping: Prism, Revolution and Pipe*

扫掠体 (*Sweeps*) 就是一条截面 (*profile*) 沿任意轨道线 (*path*) 扫掠后得到的体。截面可以是任意的拓扑结构。轨道线通常是曲线 (*curve or wire*)。创建扫掠体的规则如下:

- | 点扫掠后得到边; *Vertices generate Edges;*
- | 边扫掠后得到面; *Edges generate Faces;*
- | 线扫掠后得到壳; *Wires generate Shells;*
- | 面扫掠后得到体; *Faces generate Solids;*
- | 壳扫掠后得到组合体; *Shells generate Composite Solids;*
- | *A Compound generates a Compound with the sweep of all its elements;*

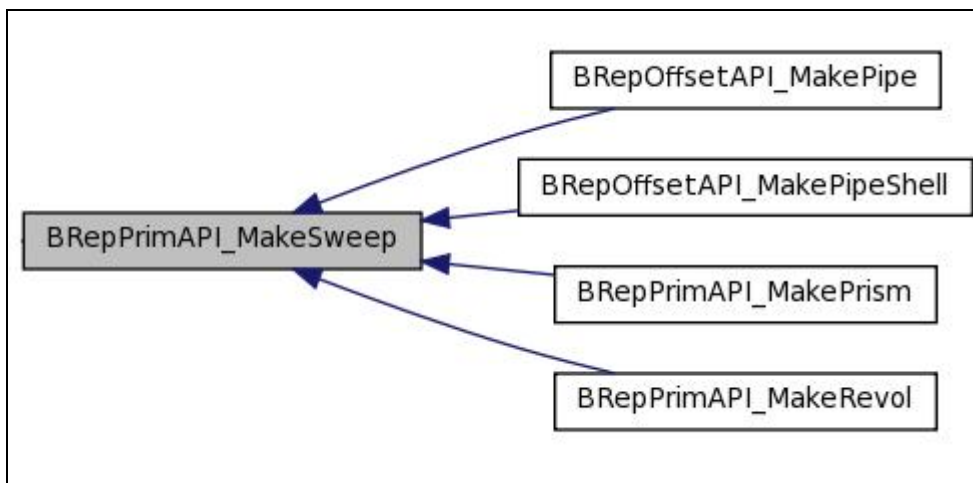
体 (*Solids*) 和组合体 (*Composite Solids*) 的扫掠是禁止的。



包 *BRepPrimAPI* 中实现了三种扫掠体: 称线性扫掠体为 *Prism*, 称旋转扫掠体为 *Revol* 和通用的扫掠体叫作 *Pipe*。

1. *BRepPrimAPI_MakeSweep*

类 *BRepPrimAPI_MakeSweep* 是扫掠体的基类。对用户来说, 没有什么实质的用途。



2. 创建拉伸体 *BRepPrimAPI_MakePrism*

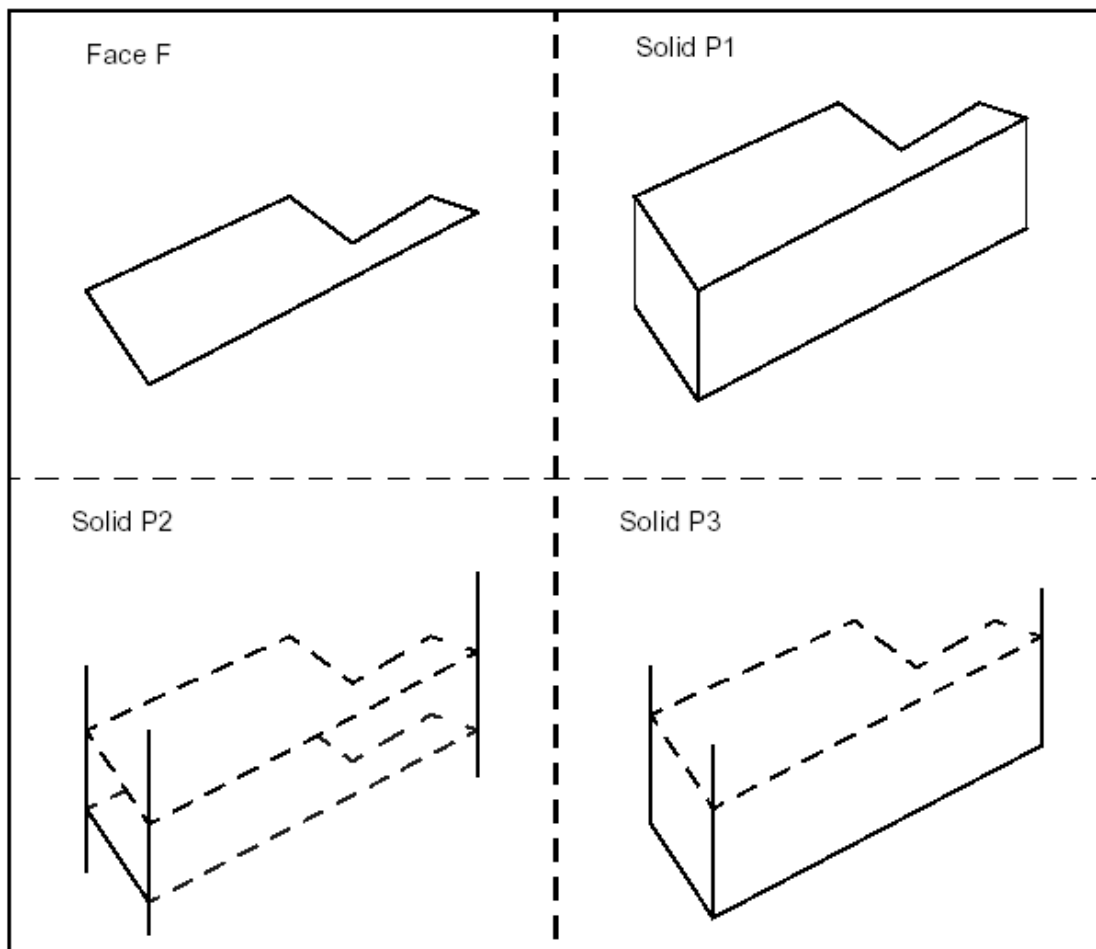
使用类 *BRepPrimAPI_MakePrism* 来创建线性拉伸体 *prism*。拉伸体可从一个形状及一个向量或一个方向来创建。从一个向量可以创建一个有限拉伸体, 而从一个方向可以创建一个无限的或半无限的拉伸体。一个布尔的函数参数用来控制半无限和全无限的拉伸体。所有的构造函数都有一个布尔值来控制构造的拉伸体与原拉伸体是否共享, 默认是共享方式构

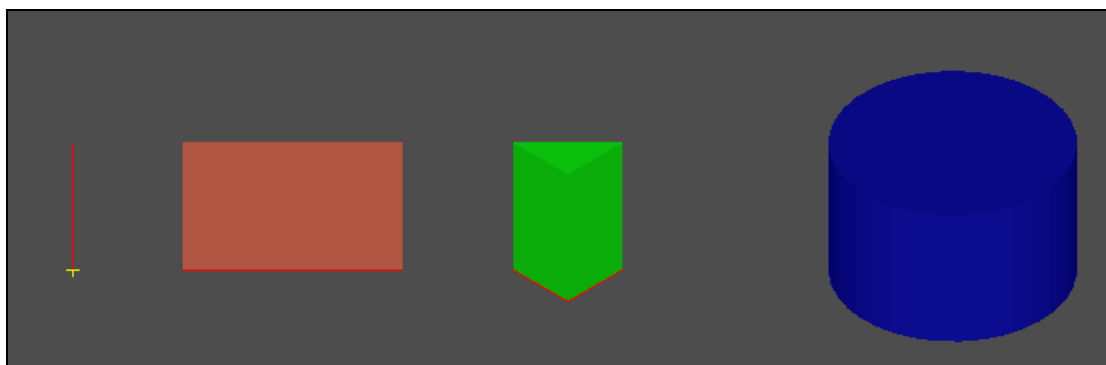
造。如下代码为使用一个面，一个方向和一个长度来创建一个有限的、无限的和半无限的拉伸体。

Example

```

TopoDS_Face F = ..; // The swept face
gp_Dir direc(0,0,1);
Standard_Real l = 10;
// create a vector from the direction and the length
gp_Vec v = direc;
v *= l;
TopoDS_Solid P1 = BRepPrimAPI_MakePrism(F,v);
// finite
TopoDS_Solid P2 = BRepPrimAPI_MakePrism(F,direc);
// infinite
TopoDS_Solid P3 = BRepPrimAPI_MakePrism(F,direc,Standard_False);
// semi-infinite
    
```





生成上图所示形状的代码如下所示:

```
//--- Prism a vertex -> result is an edge ---
TopoDS_Vertex V1 = BRepBuilderAPI_MakeVertex(gp_Pnt(-200., -200., 0.));
TopoDS_Shape S1 = BRepBuilderAPI_MakePrism(V1, gp_Vec(0., 0., 100.));
//--- Prism an edge -> result is a face ---
TopoDS_Edge E = BRepBuilderAPI_MakeEdge(gp_Pnt(-150., -150., 0.), gp_Pnt(-50., -50., 0.));
TopoDS_Shape S2 = BRepPrismAPI_MakePrism(E, gp_Vec(0., 0., 100.));
//--- Prism an wire -> result is a shell ---
TopoDS_Edge E1 = BRepBuilderAPI_MakeEdge(gp_Pnt(0., 0., 0.), gp_Pnt(50., 0., 0.));
TopoDS_Edge E2 = BRepBuilderAPI_MakeEdge(gp_Pnt(50., 0., 0.), gp_Pnt(50., 50., 0.));
TopoDS_Edge E3 = BRepBuilderAPI_MakeEdge(gp_Pnt(50., 50., 0.), gp_Pnt(0., 0., 0.));
TopoDS_Wire W = BRepBuilderAPI_MakeWire(E1, E2, E3);
TopoDS_Shape S3 = BRepPrismAPI_MakePrism(W, gp_Vec(0., 0., 100.));
//--- Prism a face or a shell -> result is a solid ---
gp_Circ c = gp_Circ(gp_Ax2(gp_Pnt(200., 200., 0.), gp_Dir(0., 0., 1.)), 80.);
TopoDS_Edge Ec = BRepBuilderAPI_MakeEdge(c);
TopoDS_Wire Wc = BRepBuilderAPI_MakeWire(Ec);
TopoDS_Face F = BRepBuilderAPI_MakeFace(gp::XOY(), Wc);
TopoDS_Shape S4 = BRepBuilderAPI_MakePrism(F, gp_Vec(0., 0., 100.));
```

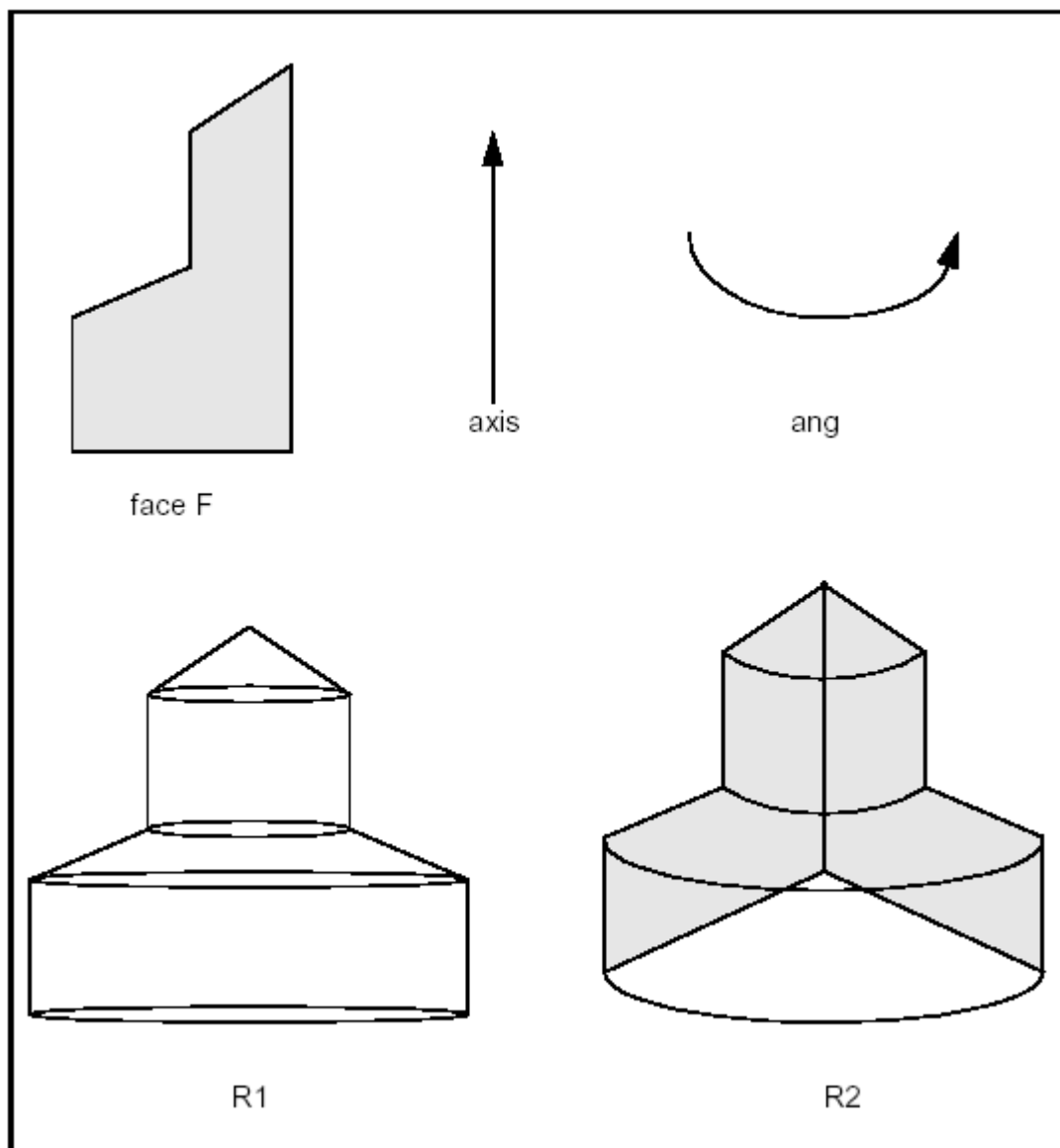

3. 创建旋转体 *BRepPrimAPI_MakeRevol*

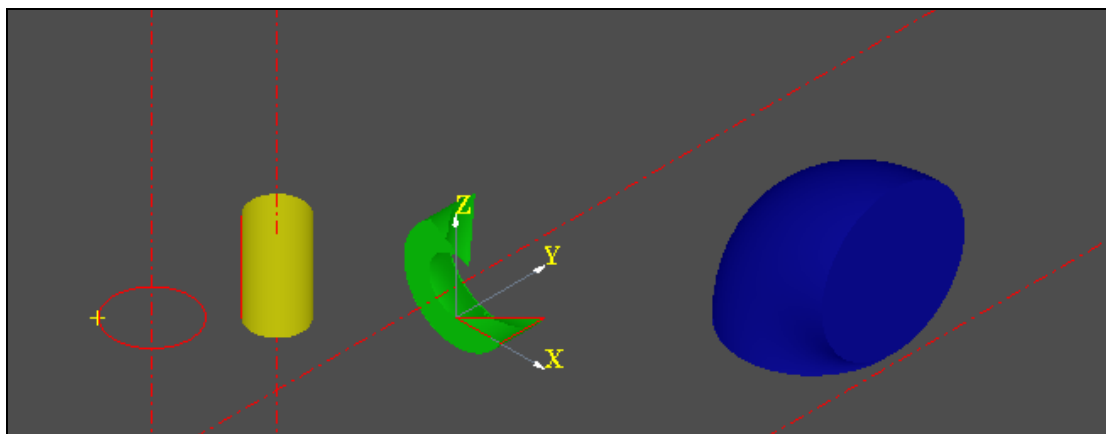
使用类 *BRepPrimAPI_MakeRevol* 来创建旋转扫掠体。旋转扫掠体由一个形状、轴和一个角度构造而得。类 *BRepPrimAPI_MakeRevol* 所有的构造函数的最后一个参数是用来控制构造的形状与原形状的关系，默认为共享的方式。

Example

```
TopoDS_Face F = ...; // the profile
gp_Ax1 axis(gp_Pnt(0,0,0),gp_Dir(0,0,1));

Standard_Real ang = PI/3;
TopoDS_Solid R1 = BRepPrimAPI_MakeRevol(F,axis);
// Full revol
TopoDS_Solid R2 = BRepPrimAPI_MakeRevol(F,axis,ang);
// Partial revol
```





生成上图所示形状的代码如下所示:

```
//--- Revol of a vertex -> result is an edge ---
TopoDS_Vertex V1 = BRepBuilderAPI_MakeVertex(gp_Pnt(-200., -200., 0.));
gp_Ax1 axe = gp_Ax1(gp_Pnt(-170., -170., 0.), gp_Dir(0., 0., 1.));
TopoDS_Shape S1 = BRepPrimAPI_MakeRevol(V1, axe);
//--- Revol of an edge -> result is a face ---
TopoDS_Edge E = BRepBuilderAPI_MakeEdge(gp_Pnt(-120., -120., 0.),
gp_Pnt(-120., -120., 100.));
axe = gp_Ax1(gp_Pnt(-100., -100., 0.), gp_Dir(0., 0., 1.));
TopoDS_Shape S2 = BRepPrimAPI_MakeRevol(E, axe);
//--- Revol of a wire -> result is a shell ---
TopoDS_Edge E1 = BRepBuilderAPI_MakeEdge(gp_Pnt(0., 0., 0.), gp_Pnt(50., 0., 0.));
TopoDS_Edge E2 = BRepBuilderAPI_MakeEdge(gp_Pnt(50., 0., 0.), gp_Pnt(50., 50., 0.));
TopoDS_Edge E3 = BRepBuilderAPI_MakeEdge(gp_Pnt(50., 50., 0.), gp_Pnt(0., 0., 0.));
TopoDS_Wire W = BRepBuilderAPI_MakeWire(E1, E2, E3);
axe = gp_Ax1(gp_Pnt(0., 0., 30.), gp_Dir(0., 1., 0.));
TopoDS_Shape S3 = BRepPrimAPI_MakeRevol(W, axe, 210.*PI/180);
//--- Revol of a face -> result is a solid ---
gp_Circ c = gp_Circ(gp_Ax2(gp_Pnt(200., 200., 0.), gp_Dir(0., 0., 1.)), 80.);
TopoDS_Edge Ec = BRepBuilderAPI_MakeEdge(c);
TopoDS_Wire Wc = BRepBuilderAPI_MakeWire(Ec);
TopoDS_Face F = BRepBuilderAPI_MakeFace(gp_Pln(gp::XOY()), Wc);
axe = gp_Ax1(gp_Pnt(290, 290., 0.), gp_Dir(0., 1., 0.));
TopoDS_Shape S4 = BRepPrimAPI_MakeRevol(F, axe, 90.*PI/180);
```