OPEN CASCADE 学习笔记
——曲面建模

著: Roman Lygin

译：George Feng

这是一篇关于开源三维建模软件 OPEN CASCADE 内核的博文：ROMAN LYGIN 是 OPEN CASCADE 的前程序开发员和项目经理，曾经写过许多关于该开源软件开发包的深入文章，可以在他的博客(HTTP://OPENCASCADE.BLOGSPOT.COM)上面找到这些文章。

序

在 OpenCascade 的论坛上知道了 Roman Lygin 在他的博客上写了 Open CASCADE notes 系列文章，但是却无法访问他的博客，幸而百度文库已经收录了 Topology and Geometry 和 Surface Modeling 两篇文章，拜读之后获益良多。如果大家发现文中翻译有错误或不足之处，望不吝赐教，可以发到我的邮箱 fenghongkui@sina.com.cn，十分感谢。

2012 年 6 月 28 日星期四

第 1 节 曲面建模(直纹曲面)

曲面建模是任意一款三维几何建模软件的基本特性。Open CASCADE(OCC)提供了一组基本的曲面(平面、锥面、球面等)，Bezier 和 B-样条曲面，回转曲面、拉伸和偏移曲面(offset surfaces)。将参数空间的底层曲面剪裁可以得到剪裁曲面(trimmed surface)。

Open CASCADE 实现了 STEP 的一个子集(ISO 标准 10303，第 42 节)，该子集用于描述几何和拓扑体，虽然与 ISO 标准 10303 稍有不同。

曲面对象只包含有最终的几何表示，而且不提供任何关于曲面是如何生成的信息。这使得它相对于其他 CAD 核心部分显得很特别，例如 ACIS 使用一种非常著名的过程曲面(procedural surface)，其中不但包含生成曲面的技术，也有可供选择的最终逼近(an optional final approximation)曲面。例如，蒙皮曲面(a skin surface)由一组截曲线(section curves)生成，整个曲面被扫过(skinned)，从而生成逼近 NURBS 的曲面。这要求建模软件支持更多的实体类型，也使得支持这些模型的建模算法变得复杂。这也使我不得不在软件 CAD Exchanger 的转换器(translator)中额外开发一些类来表示所有的这些变体(variety)，并将它们转换到 OCC 中。我可以将 OCC 中导出的 SAT 文件再重新导入到 OCC 中，但是到目前为止还不能支持所有的 SAT 类型。顺便说一下，如果有谁对 ACIS 非常熟悉，就可以对 OCC 和 ACIS 做一个非常有价值对比测试报告。

OCC 采用的方法却不同，在 OCC 中，建模算法与模型本身是分离的，例如在 OCAF 中是使用函数驱动(function drivers)。B-Rep 模型中仅仅包含各种操作最终的结果，从而使得该模型具有更好的兼容性。

另外一点要注意的是，OCC 提供了几何层面(geometry level)的算法(用来处理 Geom_Surface 和 Geom_Curve 对象)，还提供了拓扑层面(TopoDS_Shape 子类)的算法。拓扑层面的算法可能会用到几何层面的算法，但是出现在几何层面的算法并不一定会出现在拓扑层面的算法中，反之亦然。一些算法只在几何层面中出现，而有些算法只在拓扑层面中出现。假如你对此不是很确信，建议读一下这一年早些时候写的关于拓扑和几何的系列文章。

让我们来看看利用 OCC 都能够使用什么建模技术。对于基本曲面生成的技术没有什么好讲的，只看看文档和头文件就足够了。下面讲 OCC 中更为高级的话题。

直纹曲面(Ruled surfaces)

直纹曲面是通过将两条曲线利用直线连接起来生成的(例如连接这两条曲线上的点，在另一种说法中直纹曲面是通过直线沿着两条曲线上的点运动生成的)。平面是直纹曲面的特例(平面可以在两条平行直线的基础上通过直线连接生成)。假如沿着两个平行的圆上的点用直线连接起来，就可以生成圆柱面或者锥面。

图 1 是一般情况下直纹曲面的样子：

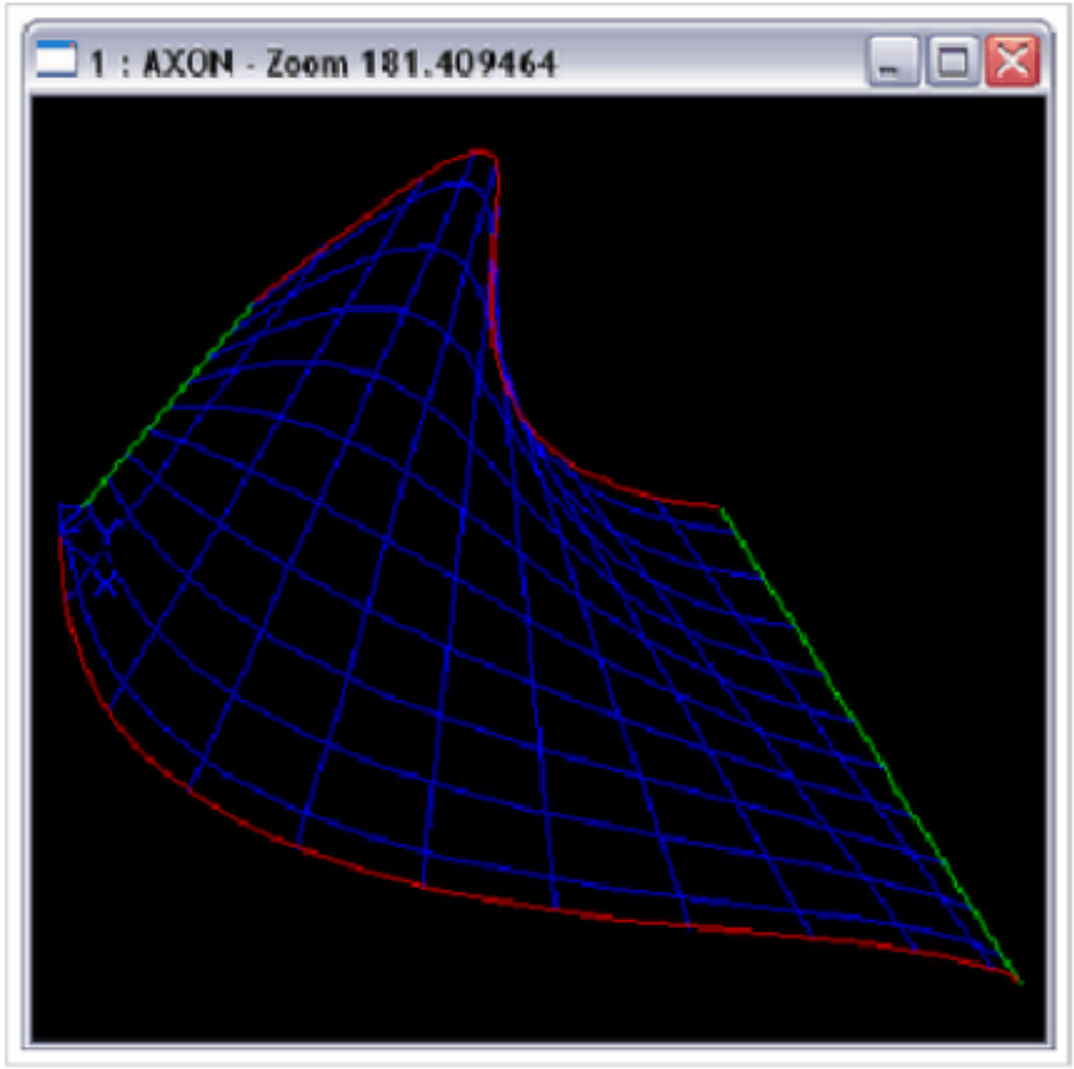

图 1 一般情况下的直纹曲面

去年春天，我们在西班牙，访问巴塞罗那(Barcelona)的 Sagrada Familia 教堂(这个教堂位于市中心，已经建造几十年了)，正好有一位建筑师的建筑技术展览会，他的名字叫 Antoni Gaudi 的，他使用了很多来源于自然的技术。使用的其中一项技术就是直纹曲面，你可以看到这篇文章中的图片，或者读一篇名字是《Gaudi and CAD》的文章。

可以用下面的代码生成一个几何层面(geometry level)直纹曲面：

```
Handle(Geom_Curve) aCrv1 = ...;
Handle(Geom_Curve) aCrv2 = ...;
Handle(Geom_Surface) aSurf = GeomFill::Surface (aCrv1, aCrv2);
```

假如要在拓扑层面(topology level)生成直纹曲面，可以利用两条边(edges) 生成面(face)或者使用两个环(wire)生成壳体(shell)。可以使用 BRepFill 来完成这个任务：

```
TopoDS_Edge anEdge1 = ...;
TopoDS_Edge anEdge2 = ...;
TopoDS_Face aFace = BRepFill::Face (anEdge1, anEdge2);

TopoDS_Wire aWire1 = ...;
TopoDS_Wire aWire2 = ...;
TopoDS_Face aShell = BRepFill::Shell (aWire1, aWire2);
```

图 2 是位于直纹曲面上的具有两个面(face)的壳体(shell)。

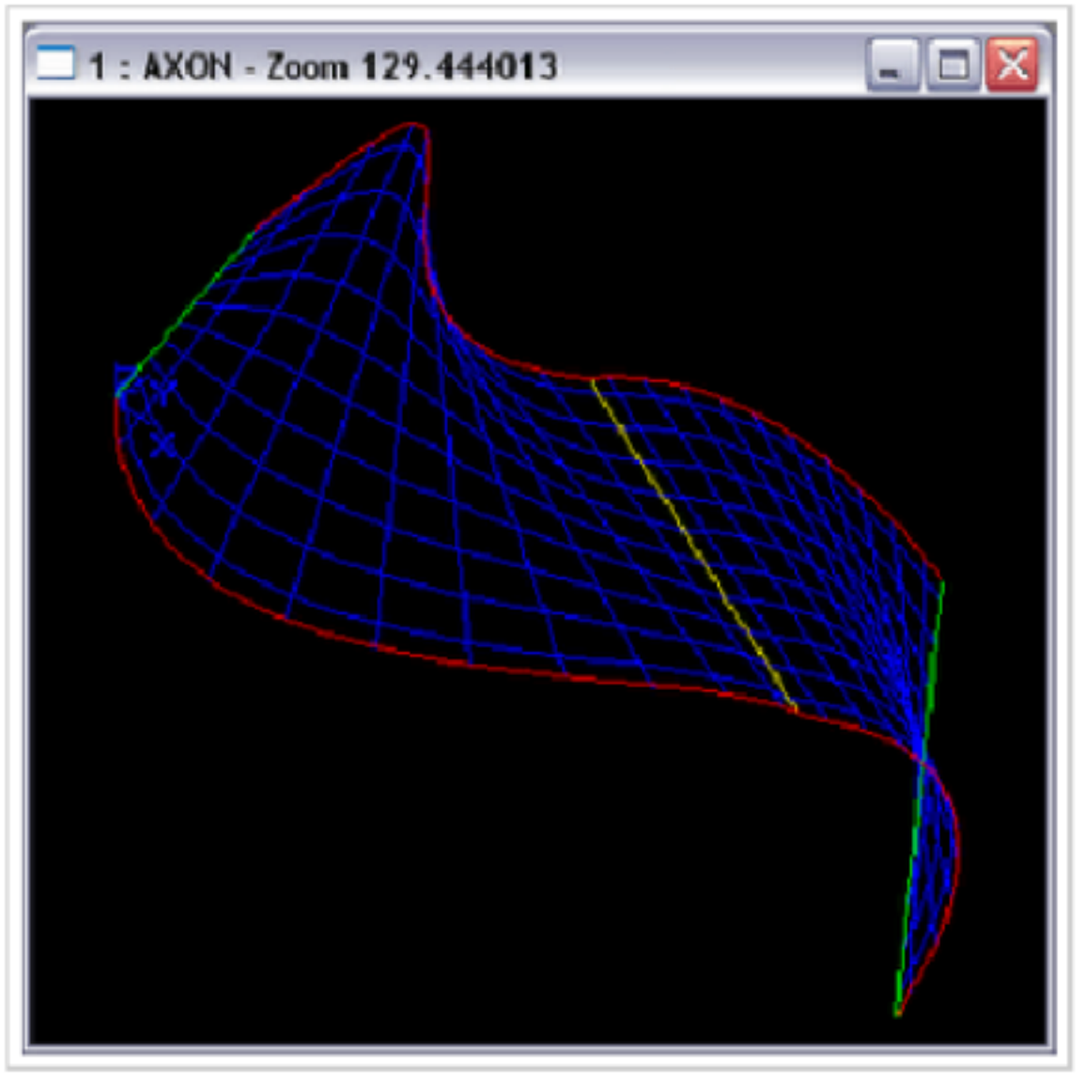

图 2 位于直纹曲面上的具有两个面的壳体

当使用 BRepFill::Shell()时，环(wires)必须包含相同数量的边(edges)。假如不相等，可能需要重新逼近生成(re-approximate)这些边。例如，可以使用 ShapeAlgo_Container::HomoWires() 或者其他相似的算法，或者使用 BRepAdaptor_CompCurve 转换器和 Approx_Curve3d 重新逼近生成环。后者可以将环变成一条 B-样条曲线(B-Spline)，这样就可以使用 GeomFill 生成直纹曲面，也可以将曲线(curve)转换为 TopoDS_Edge 类型的拓扑边从而使用 BRepFill 生成直纹曲面。

待续... ...

第 2 节 扫略曲面

接上节... ...

扫略曲面(Sweep surfaces)
扫略曲面通过母线(a profile)沿着一根样条曲线移动生成。
图 3 是一个典型的扫略曲面。

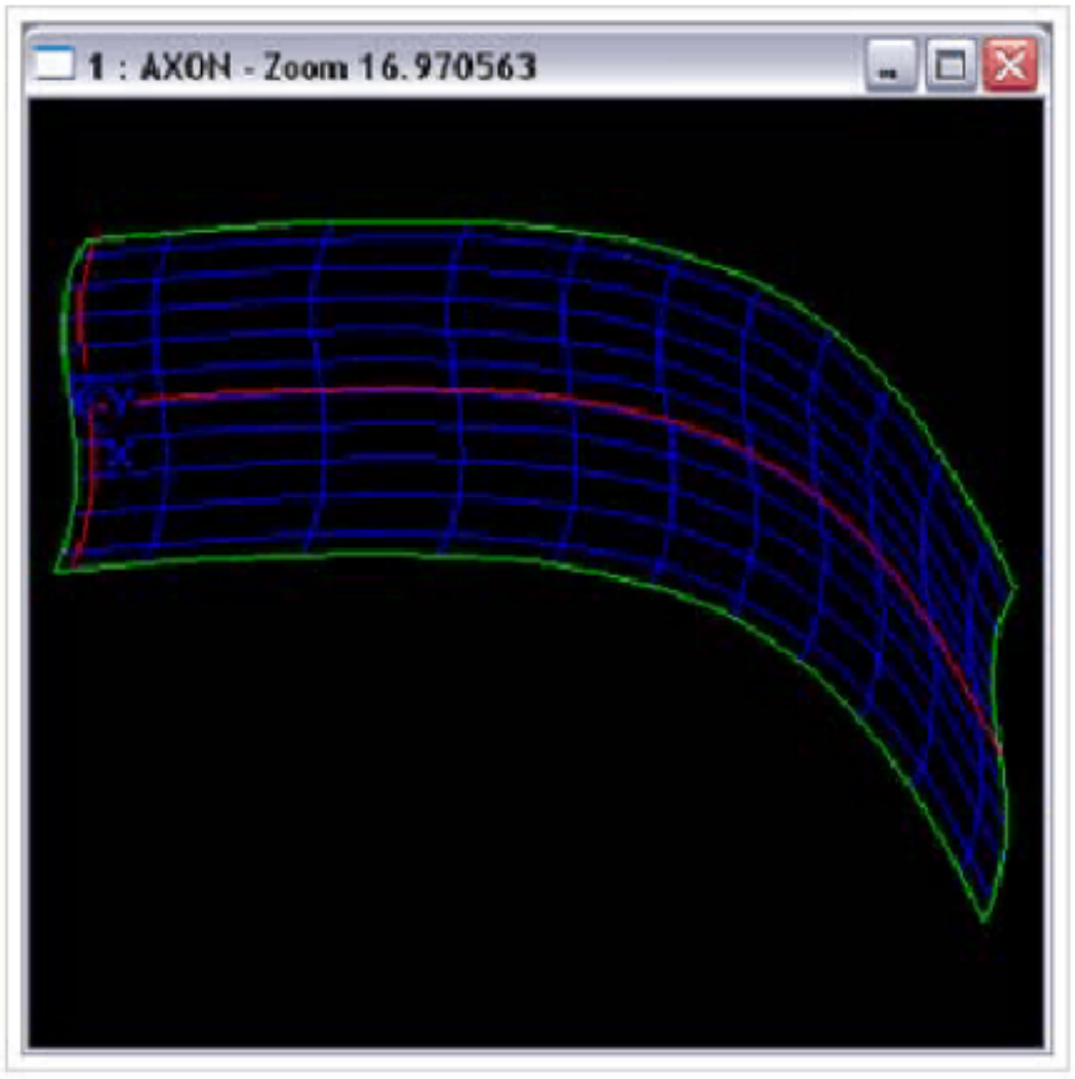

图 3　一个典型的扫略曲面

扫略曲面通过使用 GeomFill_Pipe 生成。也许管子(pipe)这个名字是从母线是闭合曲线时的特例中产生的，此时会生成一个像管子一样的曲面。

```
GeomFill_Pipe Pipe;
GeomFill_Pipe aPipe (aPath, aProfile, GeomFill_IsFixed);
aPipe.GenerateParticularCase(Standard_True);
aPipe.Perform(aTol, Standard_False, GeomAbs_C1, BSplCLib::MaxDegree(),
1000);
const Handle(Geom_Surface)& aSurface = aPipe.Surface();
```

这段代码是从 CAD Exchanger 中摘录的，用来转换 ACIS 中的 sum_spl_sur，其中曲面是由两根曲线来定义。

缺省情况下，扫略曲面会生成 B 样条曲面，可以是有理样条曲面也可以是多项式样条曲面，这依赖于 Perform()方法中的参数。假如你想生成基本曲面(圆环面 torus，圆柱面 cylinder，球面 sphere 等)，当曲线配置允许的情况下(when curves configuration allows)，可以调用函数 GenerateParticularCase()，并将参数设置为 Standard_True。

这个算法可能会返回逼近错误(an approximation error)，可以用函数 ErrorOnSurf() 捕获这个错误。

扫略曲面是通过母线沿着管子样条曲线移动生成的，并在移动过程中根据样条曲线修改母线的朝向。该过程可以通过一个类型为 GeomFill_Trihedron 的参数控制。下面的图 4-6 展示了在具有相同的样条曲线和母线(半圆)的情况下，最终生

成的曲面可以不同。

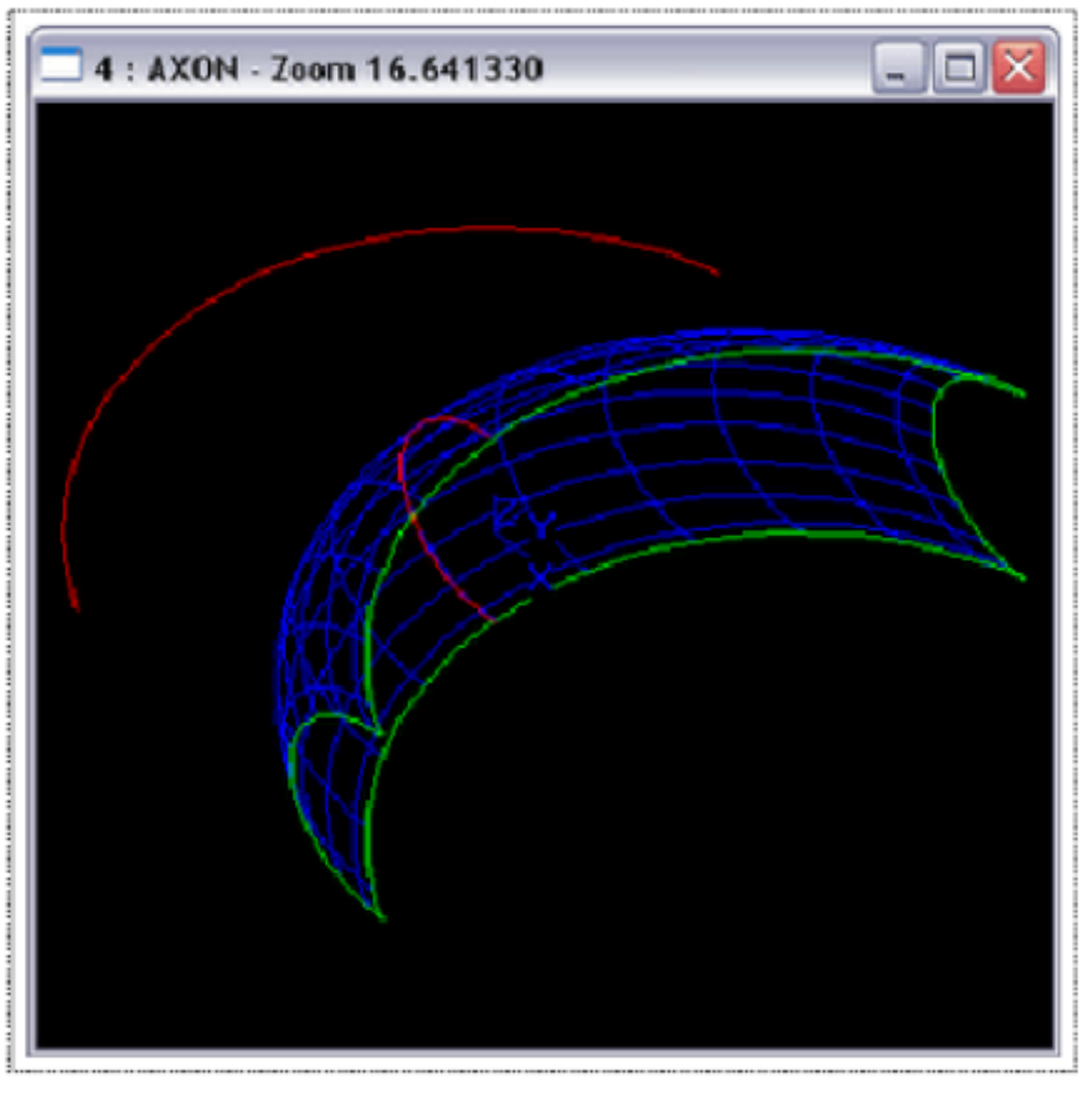

图 4　GeomFill_IsFixed

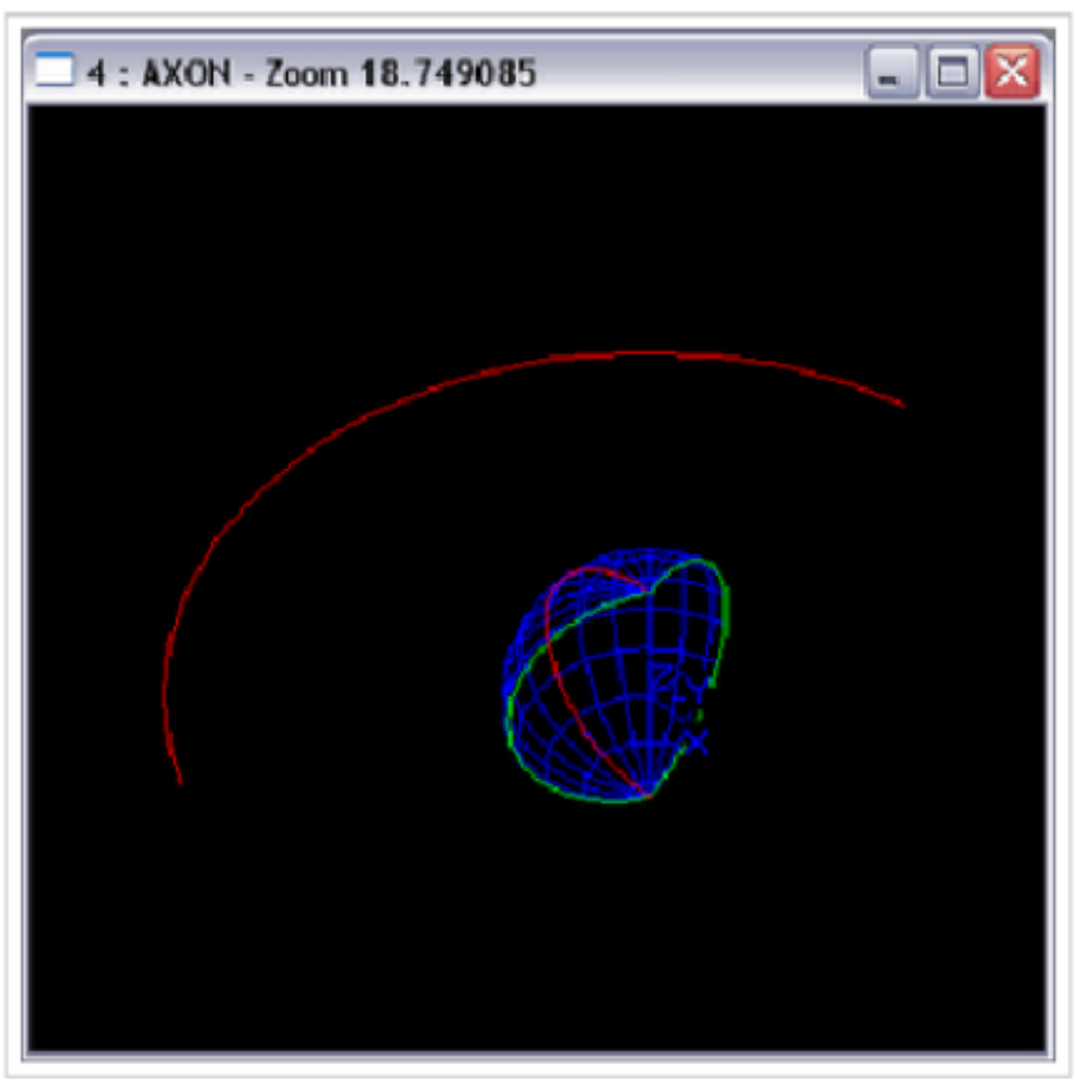

图 5　GeomFill_IsFrenet

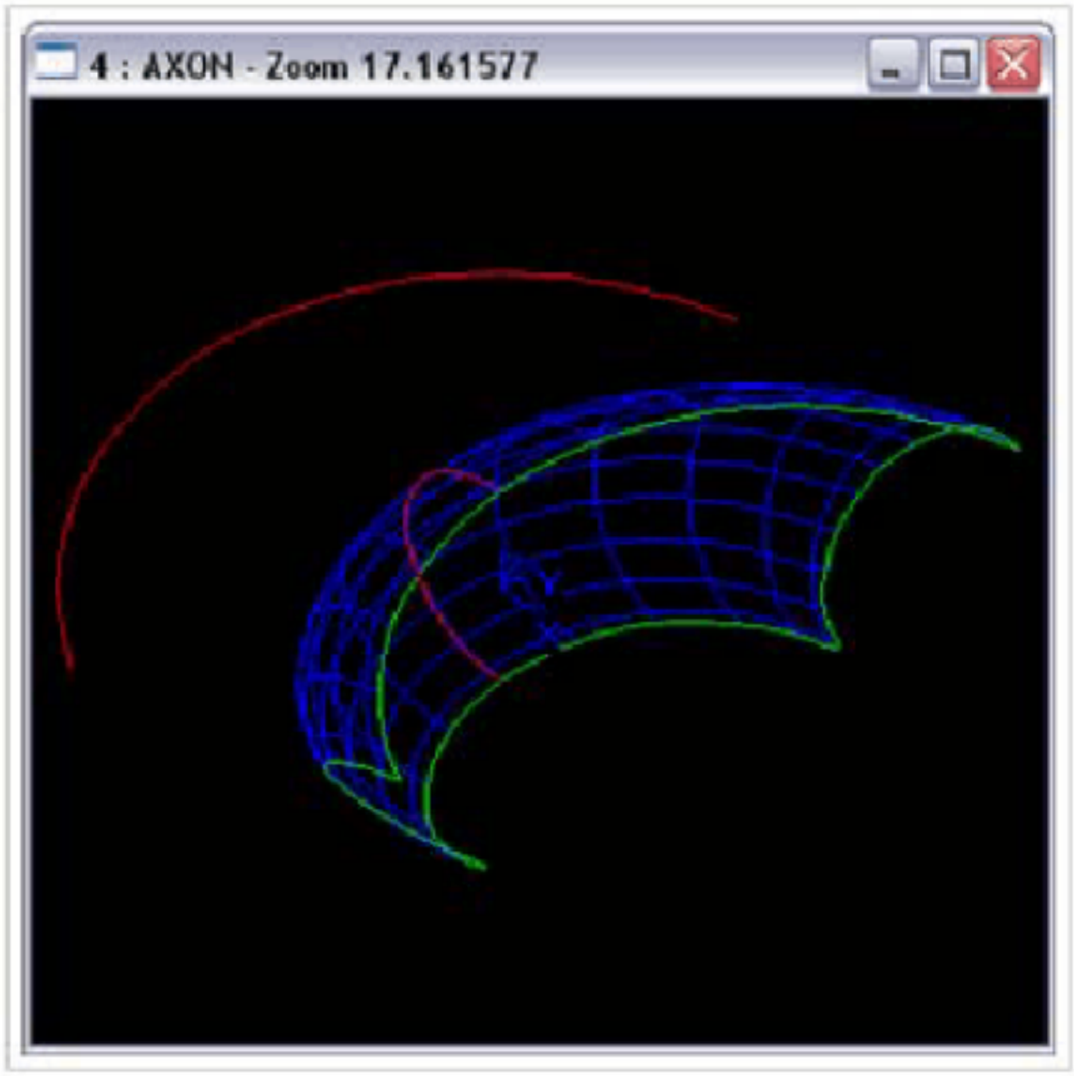

图 6　GeomFill_IsConstantNormal

可以在 DRAW 程序中做一个测试，输入'sweep' 命令，并提供不同的参数选项。

管子 Pipes

GeomFill_Pipe 提供了几种预定义的生成扫略曲面的方法：

1) 具有恒定截面的管子；
2) 具有恒定半径的圆管；
3) 具有恒定半径和两条轨道的圆管。

上面已经讨论过具有恒定截面的管子。下面是两个这种管子的例子：

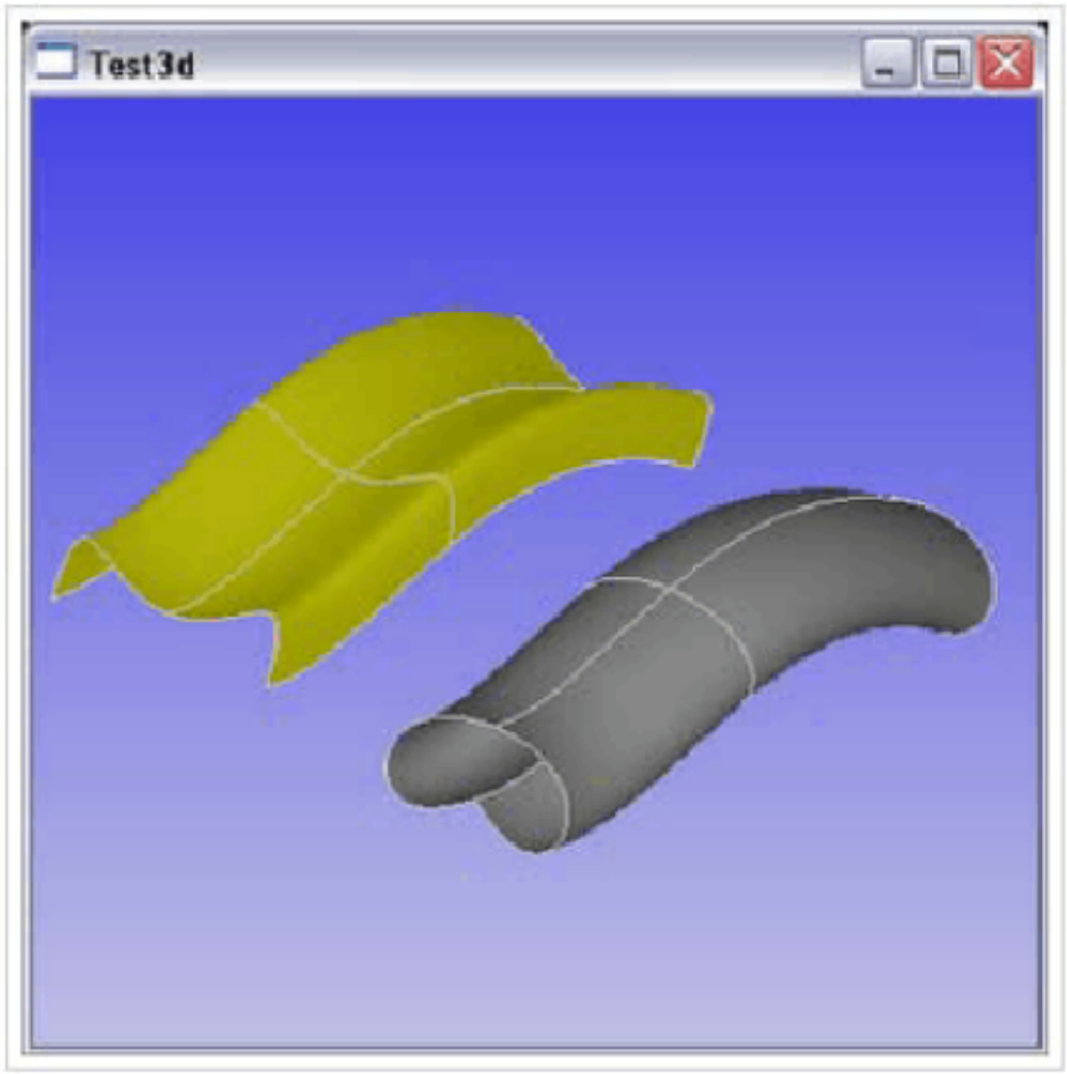

图 7　两个具有恒定截面的管子

待续... ...

第 3 节　管子

接上节... ...

半径为定值的管子

截面恒定不变的管子的一个特例是半径为定值的管子。在 ACIS 中，这样的曲面称为 tubes(管子)。

图 8 是一个例子。

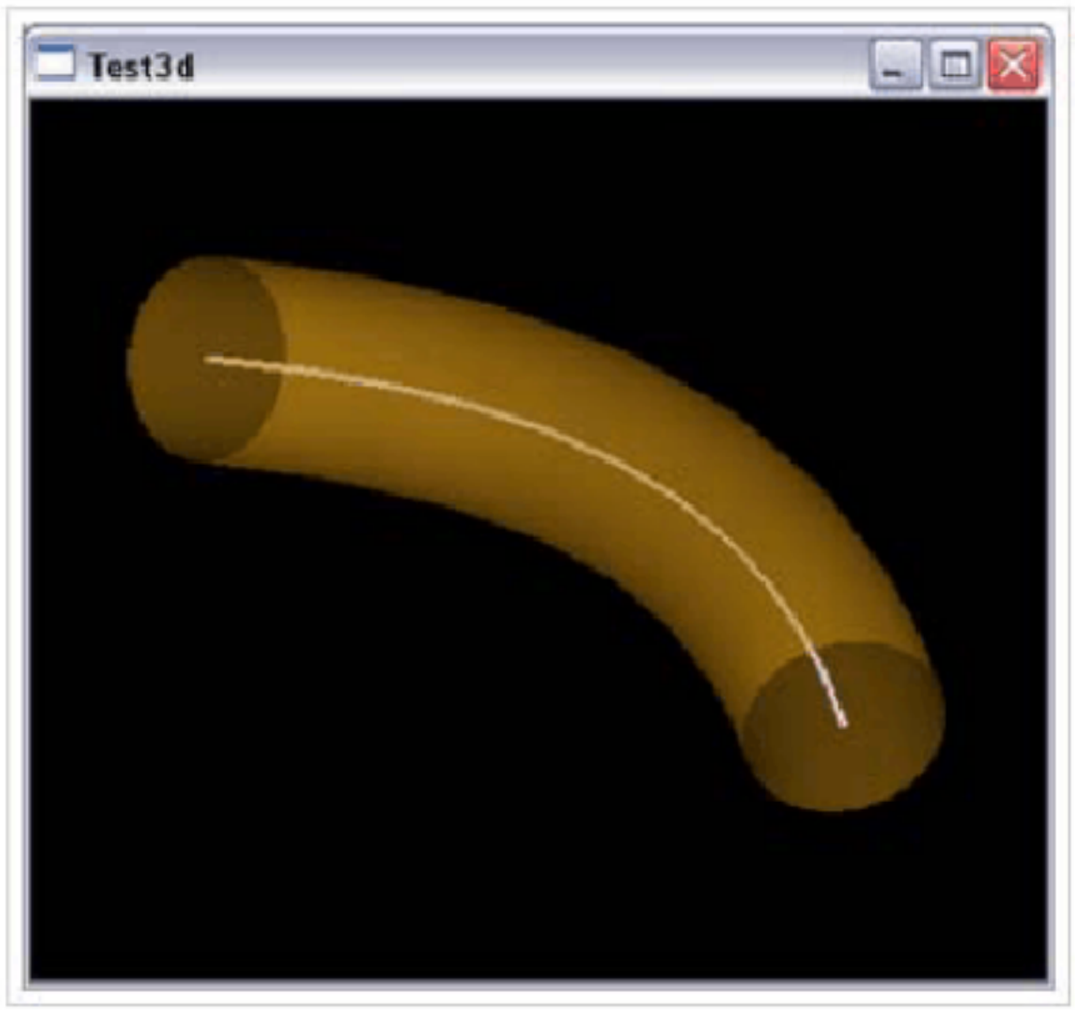

图 8　半径为定值的管子

下面是生成半径为定值的管子的例子：

```
GeomFill_Pipe aTube (thePath, theRadius);
aTube.Perform (aTol, Standard_False, (GeomAbs_Shape)Min (GeomAbs_C1,
thePath->Continuity()), aMaxDeg, aMaxSeg);
```

具有定值半径和两条轨道的管子

为了方便起见，管子算法允许设定两个轨道(rail)。这两个轨道用来限定截面。这个算法可以用来建立滚球曲面的模型(to model so-called rolling ball surfaces)。在 ACIS 中，滚球曲面描述为具有恒定半径的球沿着一定的路径(path)滚动，且始终受到两个平面的作用，从而形成的曲面。球在边界曲面上形成的轨迹称为轨道曲线(spring or rail curves)。

Open CASCADE 算法通过半径、路径和两个轨道曲线，从而生成具有部分圆截面的曲面。截面在与路径和轨道曲线相交且垂直的平面上生成的。

图 9 和图 10 是滚球曲面的两个例子：

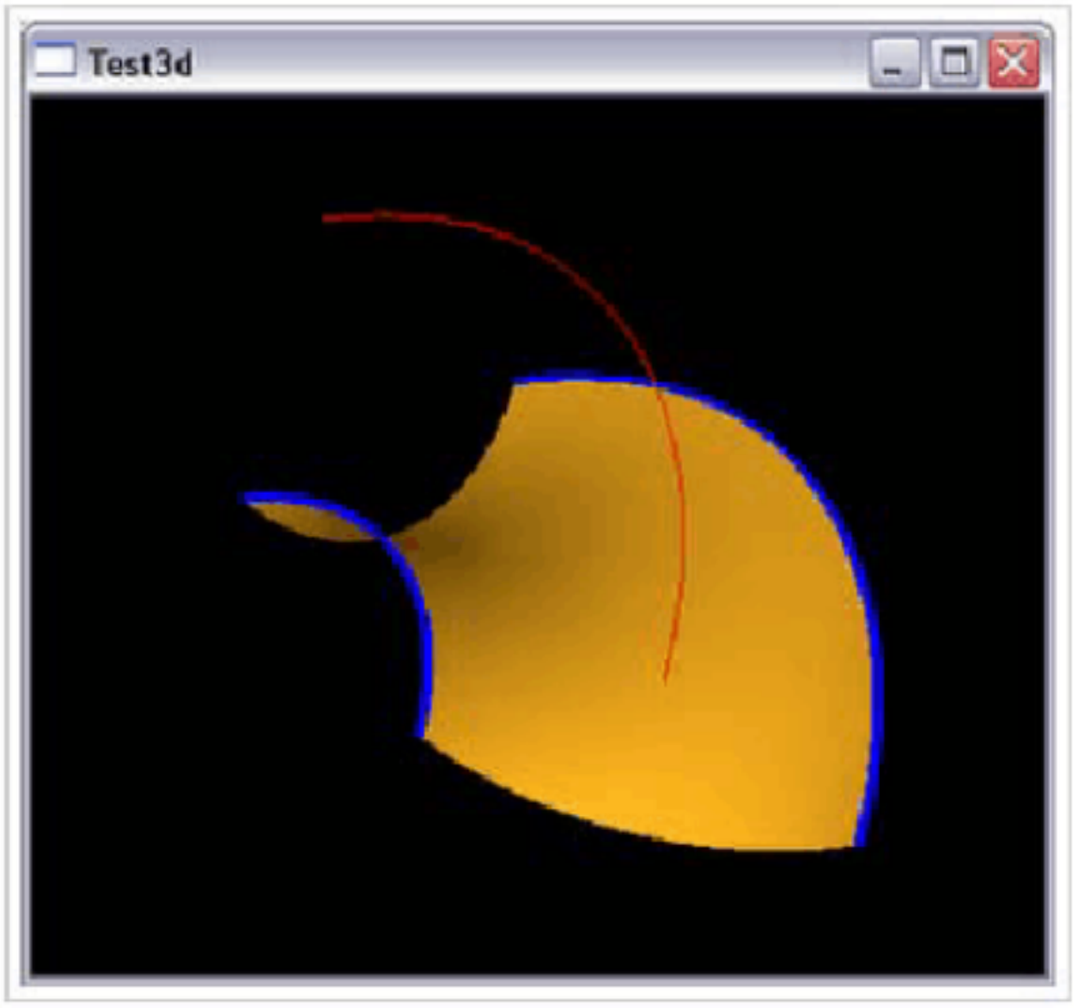

图 9　滚球曲面的两个例子之一

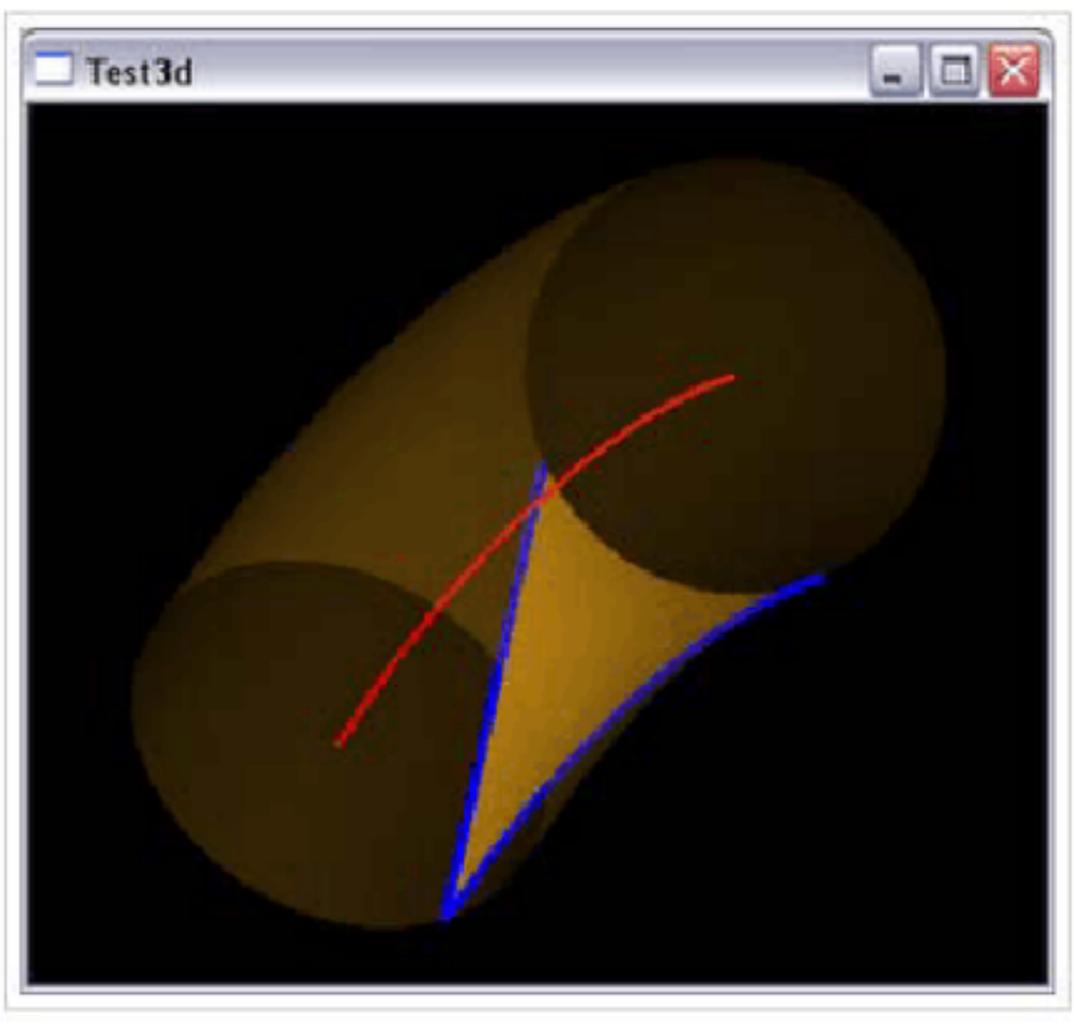

图 10　滚球曲面的两个例子之二

在这两幅图中管子路径都是红色的，轨道曲线是蓝色的。图 10 中包含有一根完整的管子(tube)，一部分管子使用两根轨道曲线生成。管道是滚动的球生成的轨迹。

需要特别注意的是 Open CASCADE 要求轨道曲线与路径的参数一致(follow the path parametrization)。这意味这轨道曲线的参数范围必须至少与路径的参数一样大，且两者之间必须相互保持一致。

具有变半径的管子

除了具有恒定半径的管子之外，也可以生成具有变半径的管子(例如：具有圆形截面的管子)。如图 11 所示。

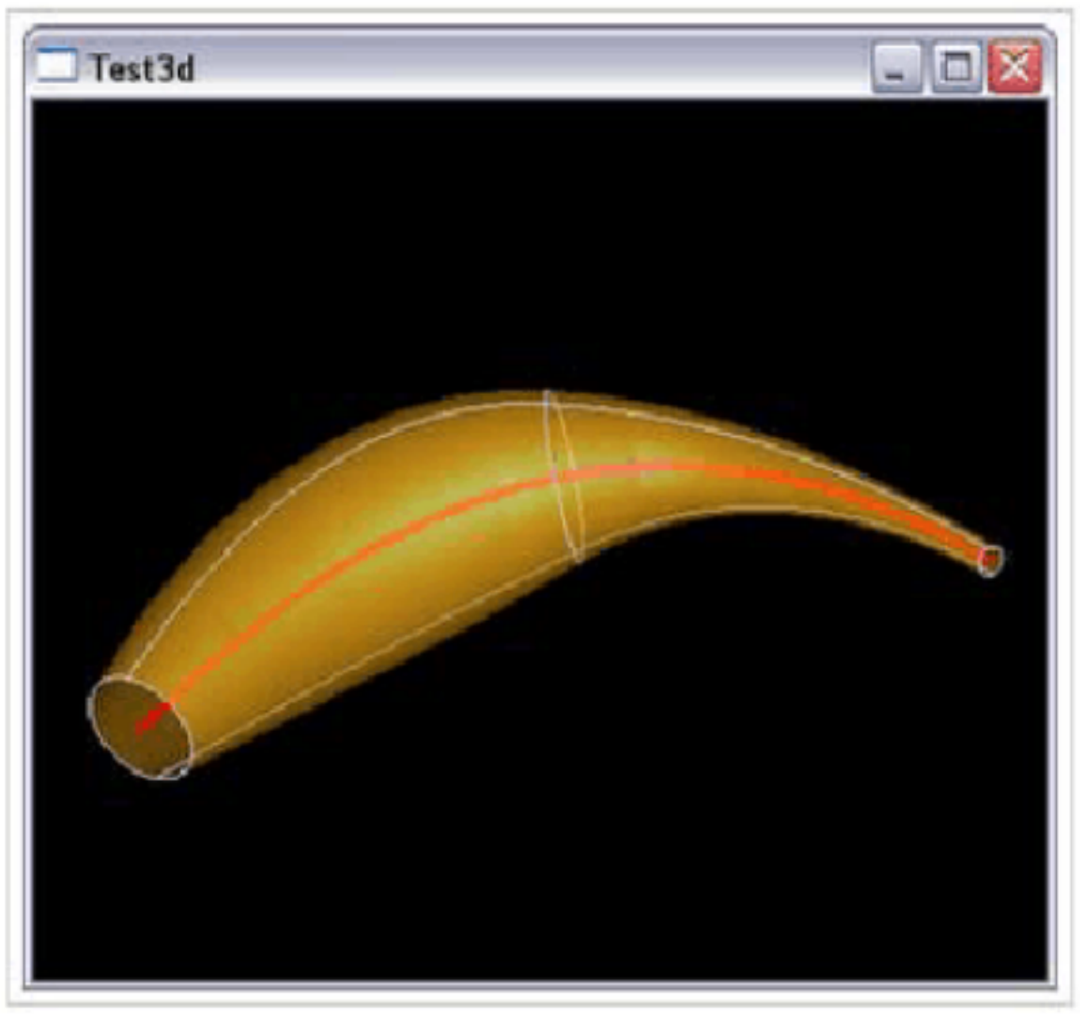

图 11　具有变半径的管子

Open CASCADE 不提供直接的 API 生成这样的曲面，但是可以使用底层的 API 来生成曲面。例如，下面是摘录的部分代码：

```
/*! Set radius evolution function with SetEvol() before calling this method.

If \a theIsPolynomial is true tries to create polynomial B-Spline, otherwise -
rational.

\sa Surface(), Error().
*/
void ACISGGeom_Pipe::Perform (const Standard_Real theTol,
const Standard_Boolean theIsPolynomial,
const GeomAbs_Shape theContinuity,
const Standard_Integer theMaxDegree,
const Standard_Integer theMaxSegment)
{
mySurface.Nullify();
myError = -1.;
```

```
if (myEvol.IsNull())
return;

//circular profile
Handle(Geom_Circle) aCirc = new Geom_Circle (gp::XOY(), 1.);
aCirc->Rotate (gp::OZ(), PI / 2.);

//code inspired by GeomFile_Pipe when using for constant radius and
corrected
//trihedron orientation

//perpendicular section
Handle(GeomFill_SectionLaw) aSec = new GeomFill_EvolvedSection (aCirc,
myEvol);
Handle(GeomFill_LocationLaw) aLoc = new GeomFill_CurveAndTrihedron (
new GeomFill_CorrectedFrenet);
aLoc->SetCurve (myPath);

GeomFill_Sweep Sweep (aLoc, myIsElem);
Sweep.SetTolerance (theTol);
Sweep.Build (aSec, GeomFill_Location, theContinuity, theMaxDegree,
theMaxSegment);
if (Sweep.IsDone()) {
mySurface = Sweep.Surface();
myError = Sweep.ErrorOnSurface();
}
}
```

在该例子中，myEvol 是类型为 Handle_Law_BSpFunc 的对象，使用二维 B 样条构建，该样条描述了半径的变化。

```
/*! Creates an internal Law_BSpFunc object which represents an evolution
function. Uses X
coordinates of the \a theEvol B-Spline curve.

\a theFirst and \a theLast are boundaries of the path curve.
*/
static Handle(Law_BSpFunc) CreateBsFunction (const
Handle(Geom2d_BSplineCurve)& theEvol,
const Standard_Real theFirst,
const Standard_Real theLast)
{
//knots are recalculated from theEvol prorate to [theFirst, theLast] range
Standard_Integer i;
```

```
const Standard_Integer aNbP = theEvol->NbPoles();
TColgp_Array1OfPnt2d aPArrE (1, aNbP);
theEvol->Poles (aPArrE);
TColStd_Array1OfReal aPArr (1, aNbP);
for (i = 1; i <= aNbP; i++)
aPArr(i) = aPArrE(i).X();

const Standard_Integer aNbK = theEvol->NbKnots();
TColStd_Array1OfReal aKArrE (1, aNbK), aKArr (1, aNbK);
theEvol->Knots (aKArrE);
TColStd_Array1OfInteger aMArr (1, aNbK);
theEvol->Multiplicities (aMArr);

const Standard_Real aKF = aKArrE(1), aKL = aKArrE (aNbK);
const Standard_Real aKRatio = (theLast - theFirst) / (aKL - aKF);
for (i = 1; i <= aNbK; i++) {
aKArr(i) = theFirst + (aKArrE(i) - aKF) * aKRatio;
}

Handle(Law_BSpline) aBs;
if (theEvol->IsRational()) {
TColStd_Array1OfReal aWArrE (1, aNbP);
theEvol->Weights (aWArrE);
aBs = new Law_BSpline (aPArr, aWArrE, aKArr, aMArr, theEvol->Degree(),
theEvol->IsPeriodic());
} else {
aBs = new Law_BSpline (aPArr, aKArr, aMArr, theEvol->Degree(),
theEvol->IsPeriodic());
}

Handle(Law_BSpFunc) aFunc = new Law_BSpFunc (aBs, theFirst, theLast);
return aFunc;
}

/*! Uses X coordinates of the \a theEvol B-Spline curve to set evolution
function.
*/
void ACISGGeom_Pipe::SetEvol (const Handle(Geom2d_BSplineCurve)&
theEvol)
{
myEvol = ::CreateBsFunction (theEvol, myPath->FirstParameter(),
myPath->LastParameter());
}
```

图 12 和图 13 是半径函数(这里是二维 B 样条曲线)和生成的具有变半径的管子曲面：

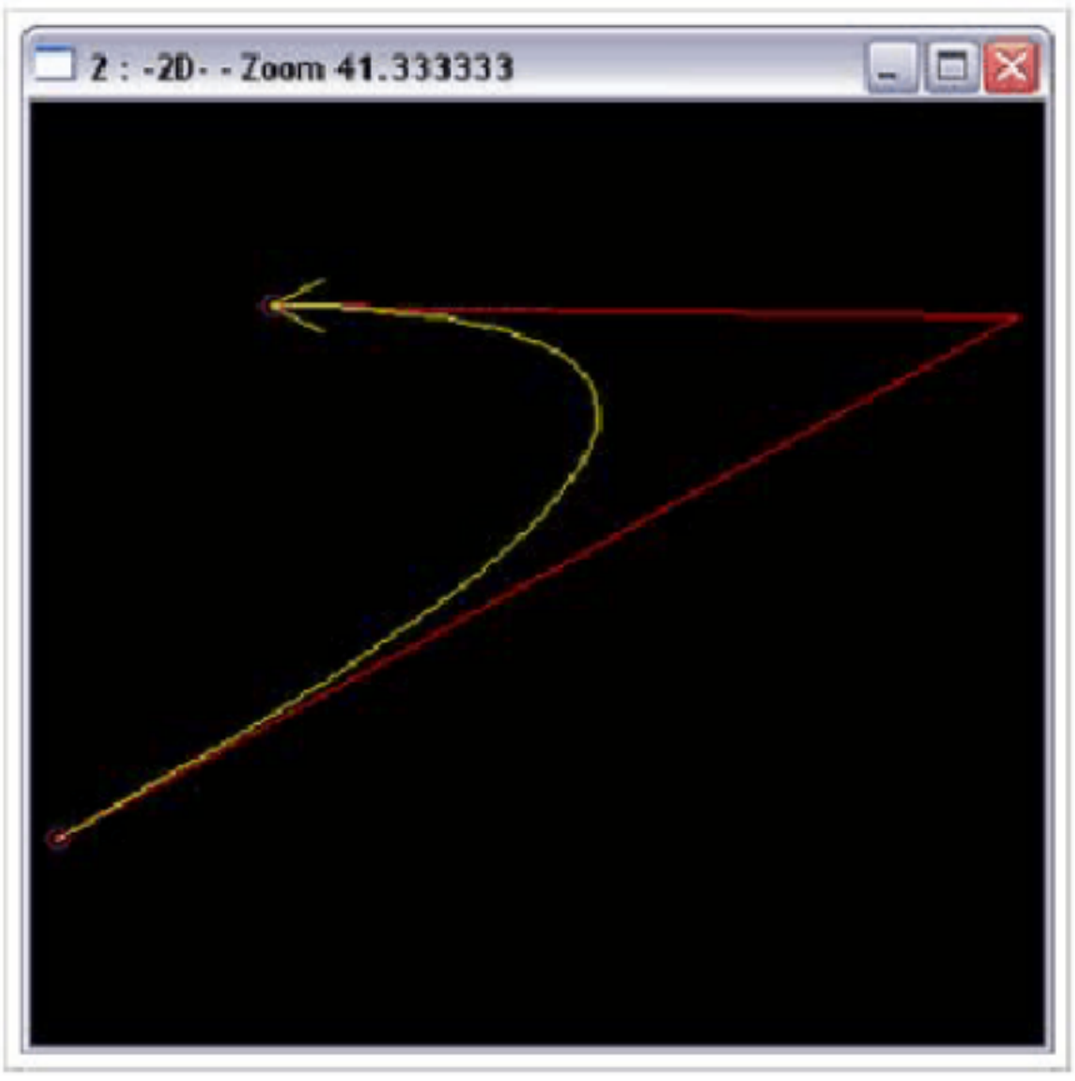

图 12　半径函数曲线

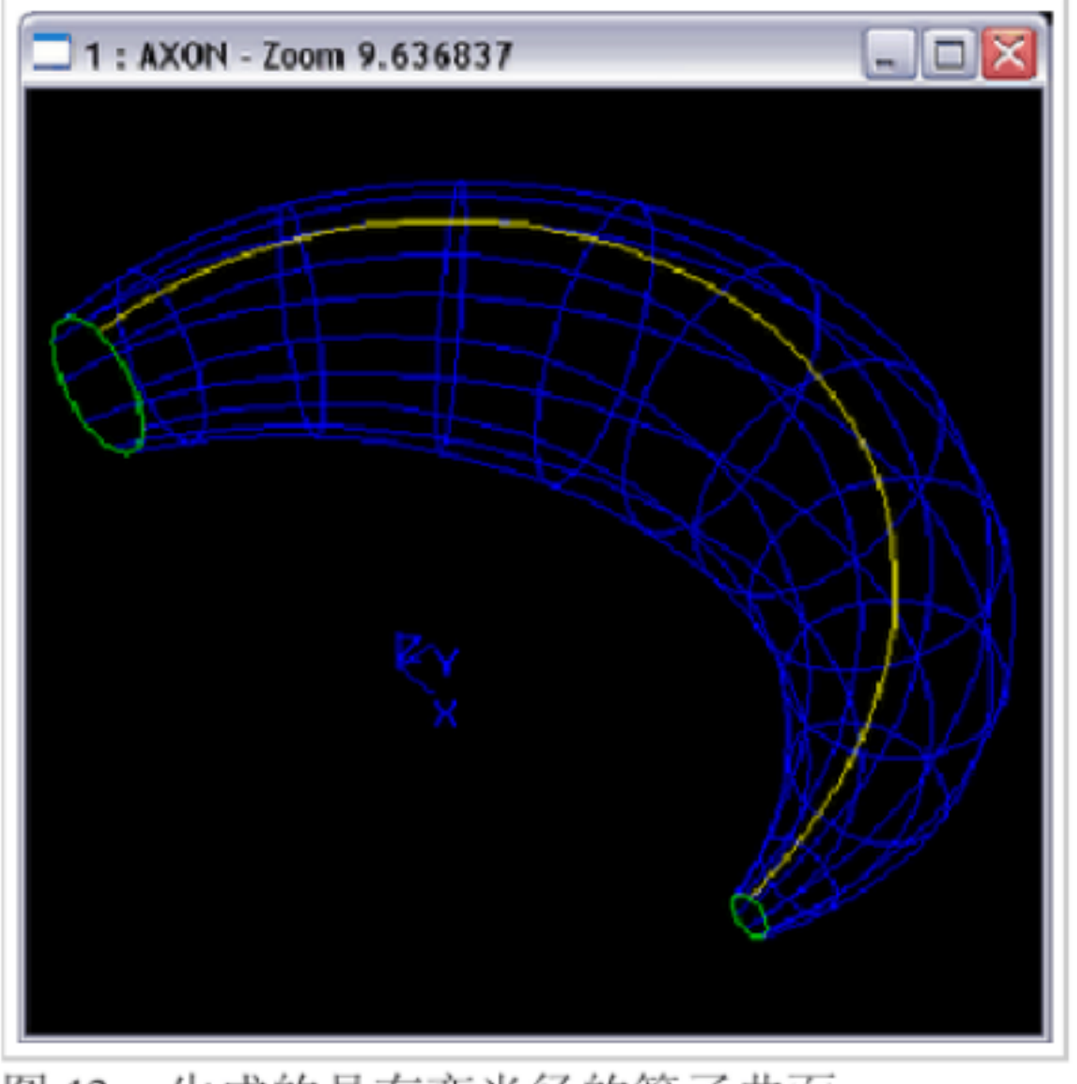

图 13　生成的具有变半径的管子曲面

小节

管子曲面在沿着截面的 U 方向上和沿着路径的 V 方向上是参数化的，曲面的参数化继承了路径的范围，且根据截面范围调整 U 参数。例如管子(tubes)在 U 方向上的参数为 0 到 2*PI。

待续... ...

第 4 节 任意扫略曲面

接上节... ...

创建任意扫略曲面

在前面章节中讨论的 GeomFill 算法在内部使用了 Approx_SweepFunction 的子类，该子类使用了特殊的技术计算横截面。

可以使用自己的子类来构建特殊的扫略曲面，但是 Approx_SweepFunction 有几个纯虚函数，在子类中必须重新实现。例如在 CAD Exchanger 的 ACIS-SAT 转换中，我使用 Approx_SweepFunction 来生成变半径曲面。在 ACIS 中，这样的曲面是通过使用所谓的辅助曲面(左和右)，路径曲线，左和右半径函数，横截面类型(圆、椭圆、斜面等)实现的。图 14 中是这种曲面的例子(用黑色和黄色表示)，具有椭圆截面，辅助面是绿色和亮黄色，路径是红色的。

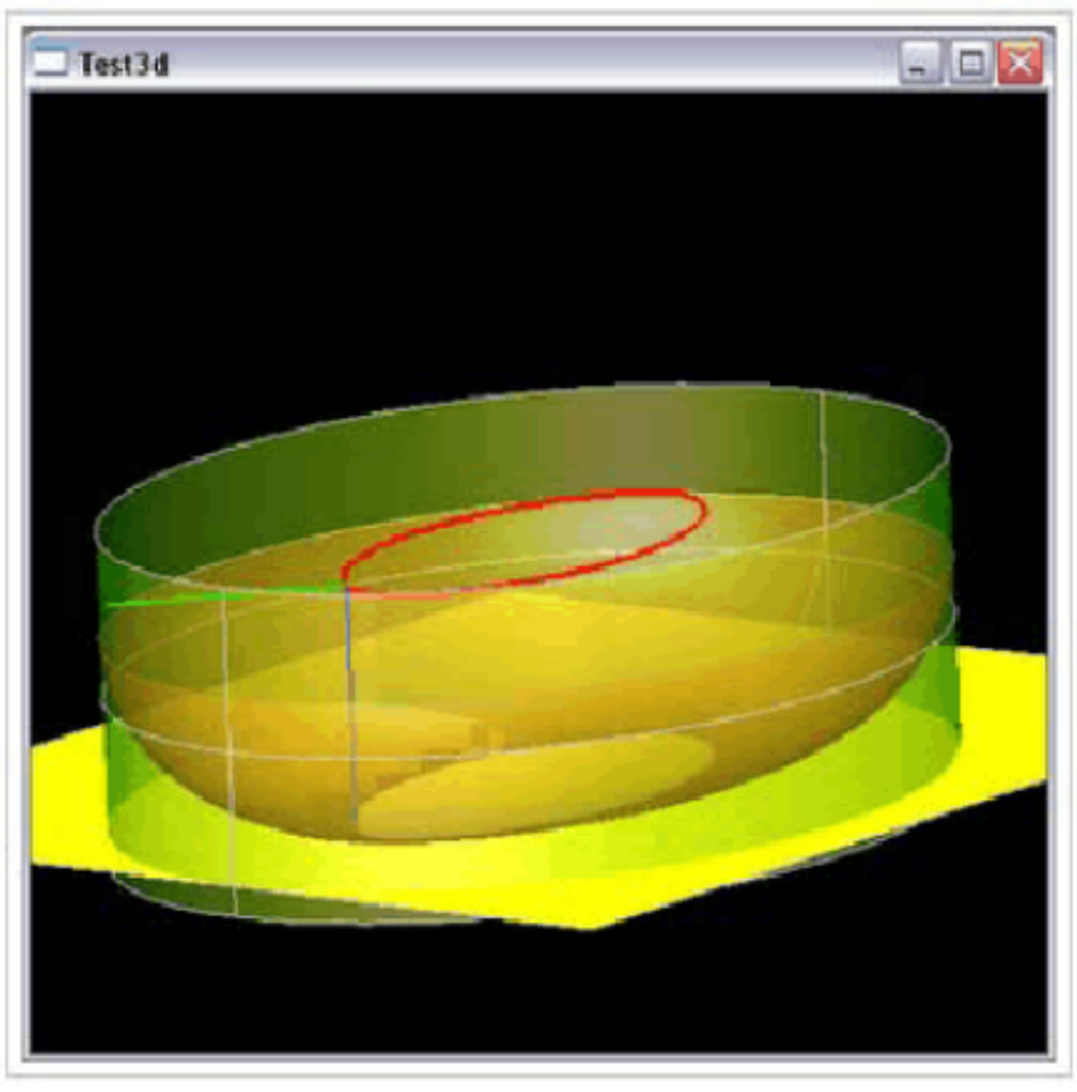

图 14　ACIS 生成扫略曲面

下面是计算在路径的给定参数位置处的横截面的方法：

```
Standard_Boolean        ACISAlgo_VarBlendSweepFunction::D0        (const
Standard_Real theParam,
const Standard_Real,
const Standard_Real,
TColgp_Array1OfPnt& thePoles,
```

```
TColgp_Array1OfPnt2d&,
TColStd_Array1OfReal& theWeights)
{
gp_Pnt2d aLRad, aRRad;
// left radius
aLRad = myLRad->Value(theParam);
// right radius
if (myRRad == myLRad) {
aRRad = aLRad;
} else {
aRRad = myRRad->Value(R2Param);
}

gp_Pnt aPathPnt;
gp_Vec aNormal;
myPath->D1 (theParam, aPathPnt, aNormal);

const gp_Ax2 Axis (aPathPnt, aNormal);
const gp_Pln aSecPln (Axis);
Handle(Geom_Plane) aGSecPln = new Geom_Plane(Axis);

//intersection of a section plan with support surfaces
Handle(TColGeom2d_HArray1OfCurve) aLCArr, aRCArr;
if (!::Intersect (aGSecPln, aSecPln, myLSurf, aLCArr) || !::Intersect (aGSecPln,
aSecPln, myRSurf, aRCArr))
return Standard_False;

//compute a section in 2D and restore it into 3D
const Standard_Integer n = thePoles.Upper();
TColgp_Array1OfPnt2d aPArr (1, n);
if (!mySec->ComputeSection (Axis, aLCArr, aLRad, aRCArr, aRRad, aPArr,
theWeights))
return Standard_False;

for (Standard_Integer i = 1; i <= n; i++) {
thePoles.SetValue (i, ElCLib::To3d (Axis, aPArr (i)));
}
return Standard_True;
}
```

上面重新实现了虚函数 D0()，用 B-Spline 曲线中的 poles 和 weights 数组来描述横截面。

这是使用 Approx_SweepFunction 的子类构造曲面的例子：

```
...
Handle(ACISAlgo_VarBlendSweepFunction)        aSweep        =        new
ACISAlgo_VarBlendSweepFunction (...);

Approx_SweepApproximation anApprox (aSweep);
const Standard_Real aTol = 1e-4;
anApprox.Perform    (aHPath->FirstParameter(),    aHPath->LastParameter(),
aTol, aTol,
Precision::Parametric (aTol), 1-3,
(GeomAbs_Shape)Min            (GeomAbs_C1,            aHPath->Continuity()),
BSplCLib::MaxDegree(), aMaxSeg);
Handle(Geom_Surface) aRes;
if (anApprox.IsDone()) {
aRes = new Geom_BSplineSurface(anApprox.SurfPoles(),
anApprox.SurfWeights(), anApprox.SurfUKnots(), anApprox.SurfVKnots(),
anApprox.SurfUMults(), anApprox.SurfVMults(),
anApprox.UDegree(), anApprox.VDegree());
```

我猜使用上面的技术你可以生成任意的扫略曲面，例如像图 15 中所示的曲面。

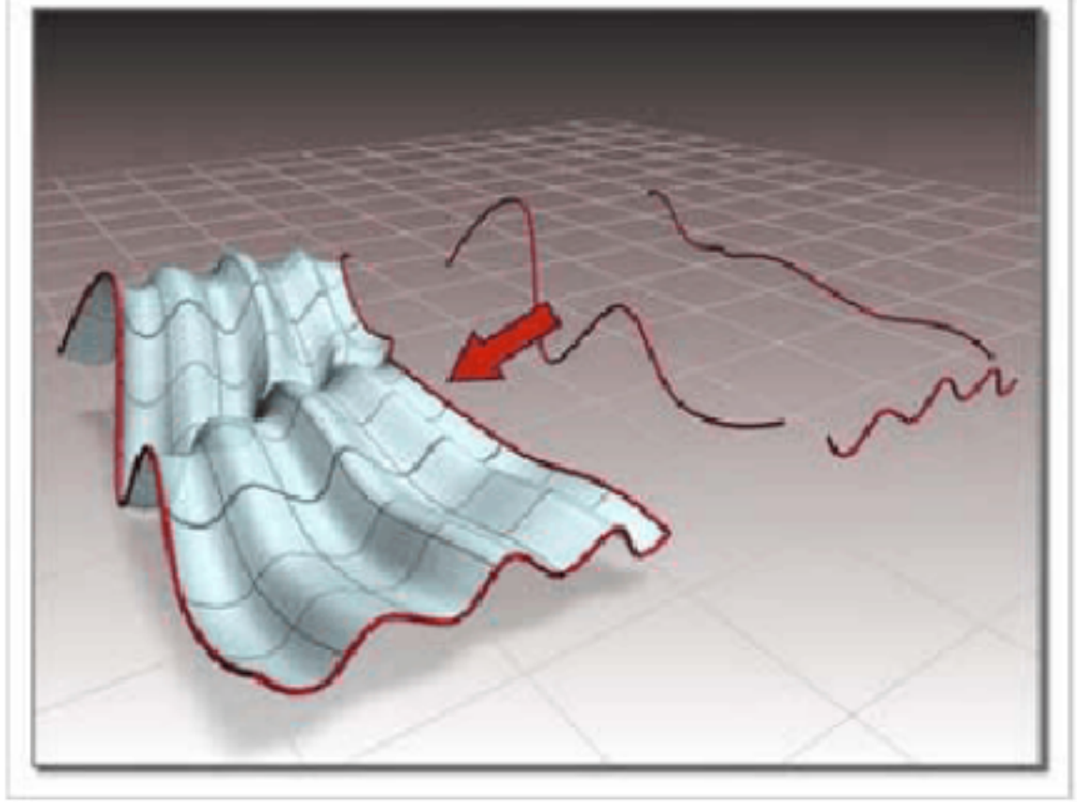

图 15　生成任意的扫略曲面的例子

拓扑算法

在前面的章节提到过，Open CASCADE 中有几何和拓扑两个层面的算法。直到目前位置我都只是关注几何层面的问题(仅仅是因为我曾经使用过它们)。下面我将尽力给出使用拓扑算法来生成任意的扫略曲面的简要介绍。

1)　将一根曲线沿着其他曲线扫略 (包括具有不变截面的管子)。

a. GeomFill_Pipe (路径(path)，母线(profile))

b. BRepOffsetAPI_Pipe (路径，母线)
2. 具有恒定半径的管子 (类似管子 tube-like):
a. GeomFill_Pipe (路径，半径):
b. BRepOffsetAPI_Pipe (路径，母线) 在此母线必须事先生成
3. 具有恒定半径和轨道的管子
a. GeomFill_Pipe (路径(path)，半径(radius)，轨道 1(rail1)，轨道 2(rail2))
b. 不支持?
4. 具有变半径的管子
a. GeomFill_Sweep (路径，半径函数)
b. 不支持?

注意拓扑算法描述母线实体时可以使用属于不同维数的元素 (can use elements of diffent dimensions for profile object)。这实际上定义了生成曲面的类型，例如，一系列顶点(a pipe of vertex)，可以生成一条边，边生成面，环生成壳，面生成实体等。

下面简要介绍一下函数 BRepOffsetAPI_MakePipeShell。当它以与环(wire spines)作为参数时，它必须能够处理各个环之间的边之间不连续(discontinuties)的情况。该函数有三种模式：
- 相交处平滑(intersection and rounding)
- 延长相交(extension and intersection)
- 当使用母线沿着环中的边移动时(along the following edge in a spine)，可以使用修改三面体模式(modifying trihedron mode) 。

图 16 是根据这三种模式生成的，其颜色分别为红、绿、蓝。

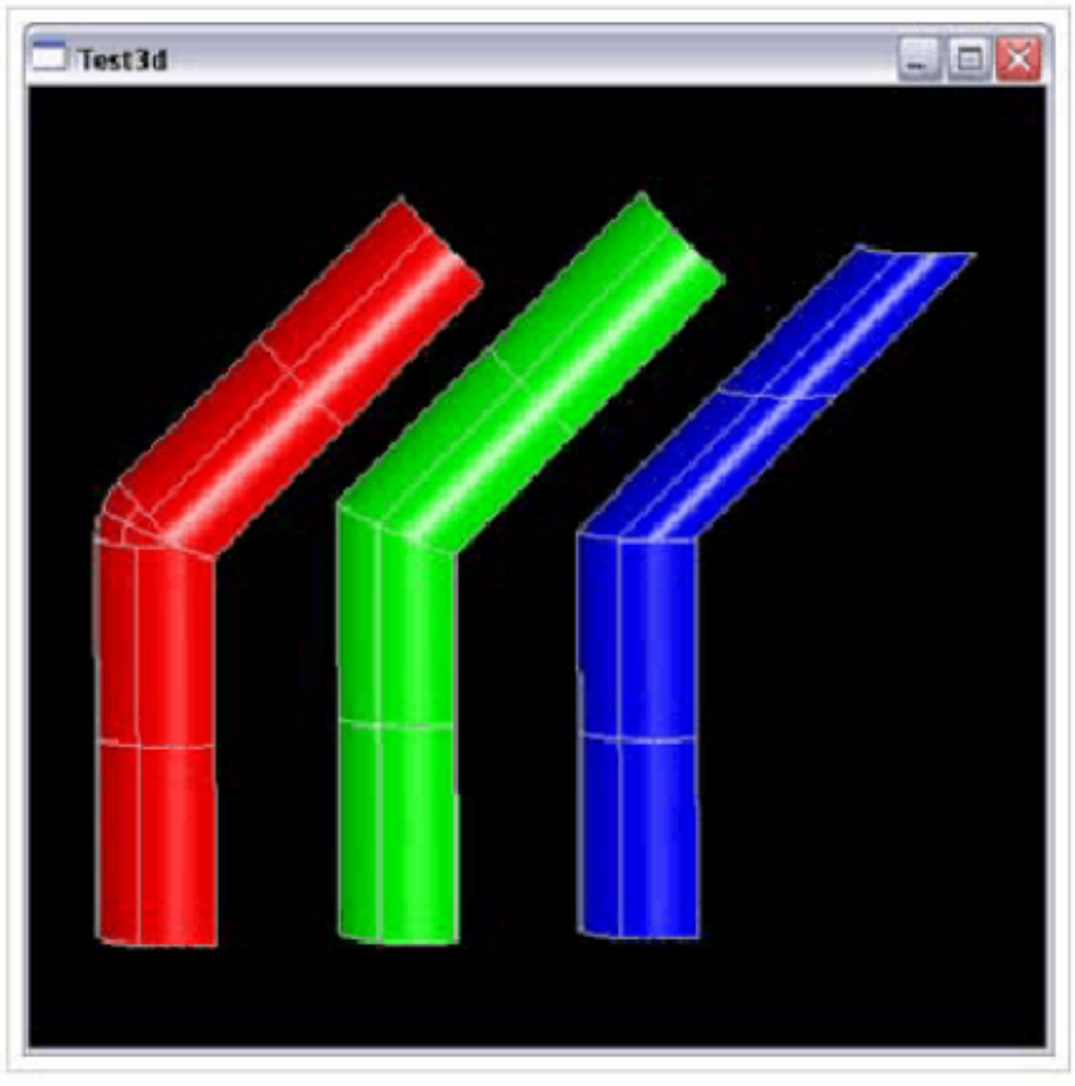

图 16　三种模式生成曲面

待续... ...

第 5 节 蒙皮

接上节... ...

时间是最宝贵的资源，在继续这个系列课程之前，我先讲一些新的东西。感谢各位对该课程的耐心和兴趣!

蒙皮和放样(Skinning and lofting)
蒙皮和放样也是一种生成模型(曲面或者壳、实体(surface or shell, or solid body))的技术，根据该技术生成曲面时，要求曲面必须通过约束曲线。 图 17 是使用蒙皮技术生成曲面的例子。

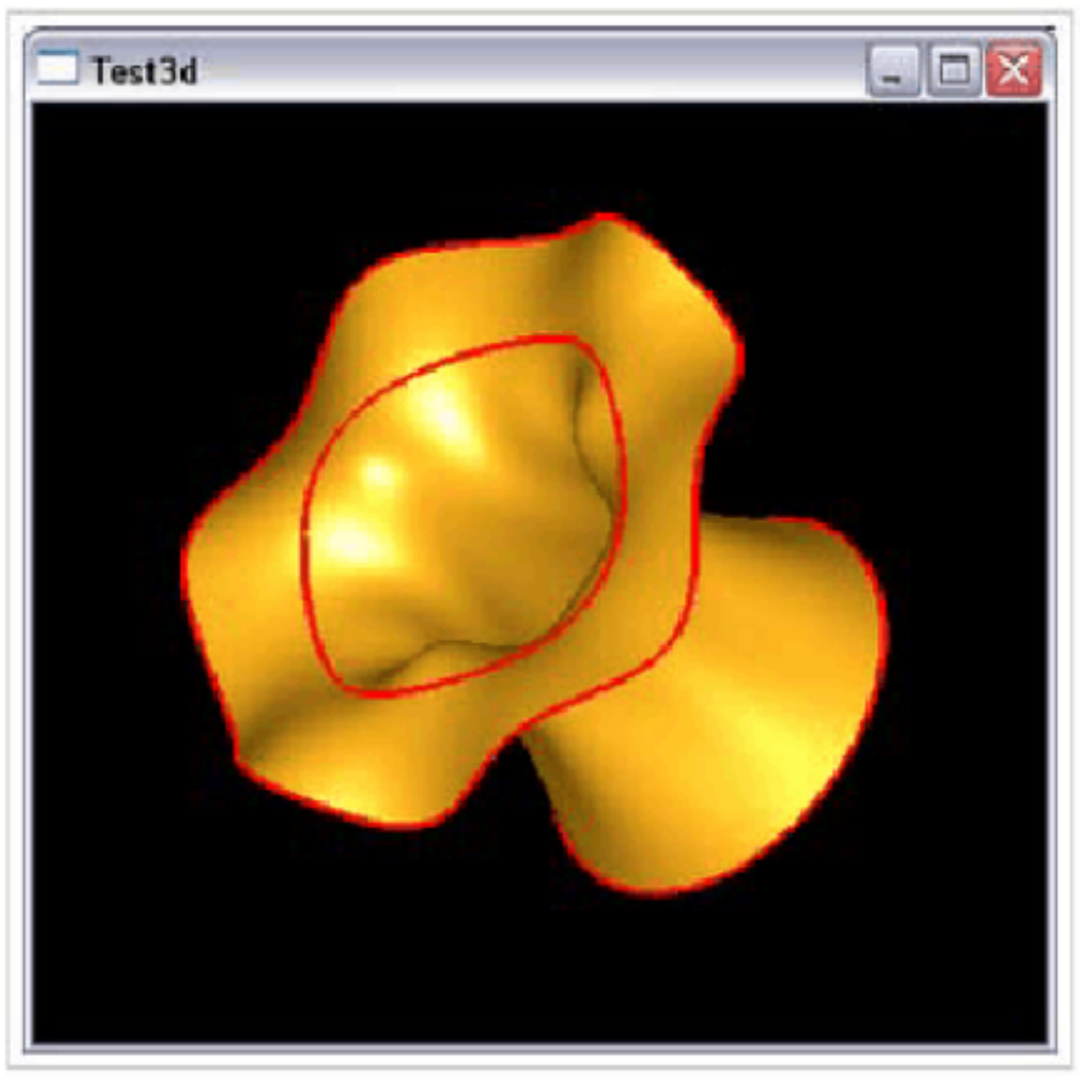

图 17　使用蒙皮技术生成曲面的例子

蒙皮和放样(Skinning and lofting)是相同的技术，Open CASCADE 对两种技术不加区别。ACIS 内核在这两种技术上有细微的区别(在输入参数的类型和约束定义的方法上不同)，但是同时也强调它们是非常相似的。

下面介绍如何使用蒙皮技术生成曲面：

```
Handle(Geom_Surface) ACISAlgo::MakeSkinSurface (
const NCollection_List<Handle(Geom_Curve)>& theSections)
```

```
{
//populate section generator
GeomFill_SectionGenerator aSecGenerator;
for (NCollection_List<Handle(Geom_Curve)>::Iterator anIt (theSections);
anIt.More();
anIt.Next()) {
const Handle(Geom_Curve)& aCurve = anIt.Value();
aSecGenerator.AddCurve (aCurve);
}
aSecGenerator.Perform (Precision::PConfusion());

Handle(GeomFill_Line) aLine = new GeomFill_Line (theSections.Size());

//parameters
const Standard_Integer aMinDeg = 1, aMaxDeg = BSplCLib::MaxDegree(),
aNbIt = 0;
Standard_Real aTol3d = 1e-4, aTol2d = Precision::Parametric (aTol3d);

//algorithm
GeomFill_AppSurf anAlgo (aMinDeg, aMaxDeg, aTol3d, aTol2d, aNbIt);
anAlgo.Perform (aLine, aSecGenerator);

Handle(Geom_Surface) aRes;
if (!anAlgo.IsDone()) {
return aRes;
}

aRes = new Geom_BSplineSurface(anAlgo.SurfPoles(), anAlgo.SurfWeights(),
anAlgo.SurfUKnots(), anAlgo.SurfVKnots(), anAlgo.SurfUMults(),
anAlgo.SurfVMults(),
anAlgo.UDegree(), anAlgo.VDegree());

...
```

曲线必须有界且连续一致参数化(consistently parameterized)(以便参数在一个方向上增加)。最后的 B 样条曲线参数化范围为[Umin, Umax; 0., 1.]，在此参数 U 依赖于曲线参数化。参数 U 沿着截面曲线，参数 V 穿过曲线。(Final B-Spline will be parameterized [Umin, Umax; 0., 1.], where U parametrization is calculated depending on the curves parametrization. parameter goes along the section curves, and V parameter goes across the curves.)

当在拓扑层面时可以使用下面的代码：

```
Standard_Boolean anIsSolid = Standard_False;
```

```
Standard_Boolean anIsRuled = Standard_False;

BRepOffsetAPI_ThruSections aGenerator (anIsSolid,anIsRuled);

...
//add constraints
for (...) {
aGenerator.AddWire (TopoDS::Wire (aConstraint));
}

Standard_Boolean anIsCheck = ...;
aGenerator.CheckCompatibility (anIsCheck);

aGenerator.Build();

const TopoDS_Shape& aResult = Generator.Shape();
```

假如边界约束是平面，拓扑算法可能会尝试生成一个封闭的实体。(The topological algorithm may attempt to create a closed solid if the boundary constraints are planar.)

不像在 ACIS 中，Open CASCADE 只允许设定曲线约束，但是没有给出一种设定相切约束的方法。所以你需要靠底层的算法使得生成的曲面变得光滑。

待续... ...

第 6 节 填充曲面(Plating)

接上节... ...

填充曲面(Plating)

这是 Open CASCADE 提供一种先进的曲面建模技术。也叫做空洞填充、约束填充(hole filling, or constrained filling)。ACIS 称它为为覆盖(covering)。

该方法在生成曲面时满足几个约束：

- 点约束——曲面必须通过给定的点；

- 点约束与相切约束，还有给定曲率(可选项)——跟上面一样，加上相对于另一个曲面的 Gn 连续要求(Gn requirements with respect to another surface)。
- 曲线约束——曲面必须通过给定的曲线。
- 具有相切约束的曲线——一样加上 Gn 连续要求。

后一种类型的约束用来生成 A 类曲面，与邻接面 G1 相切(G1 tangent to adjacent surfaces)。这非常重要，例如，在汽车工业中，当设计好看的汽车车体时，一定要确保光滑，且正确反光，避免出现尖锐的棱边。

约束可以是混合型的(例如，3 条曲线约束，1 条具有相切约束的曲线，2 个点约束)。

构建填充曲面的主算法是 GeomPlate_BuildPlateSurface。我在 CAD Exchanger 中利用该算法将 vertex_blend 的 ACIS 曲面和网曲面转换。前者是通过顶点混合生成的，从而生成了与邻接面 G1 连续的曲面。图 18 是一个例子。

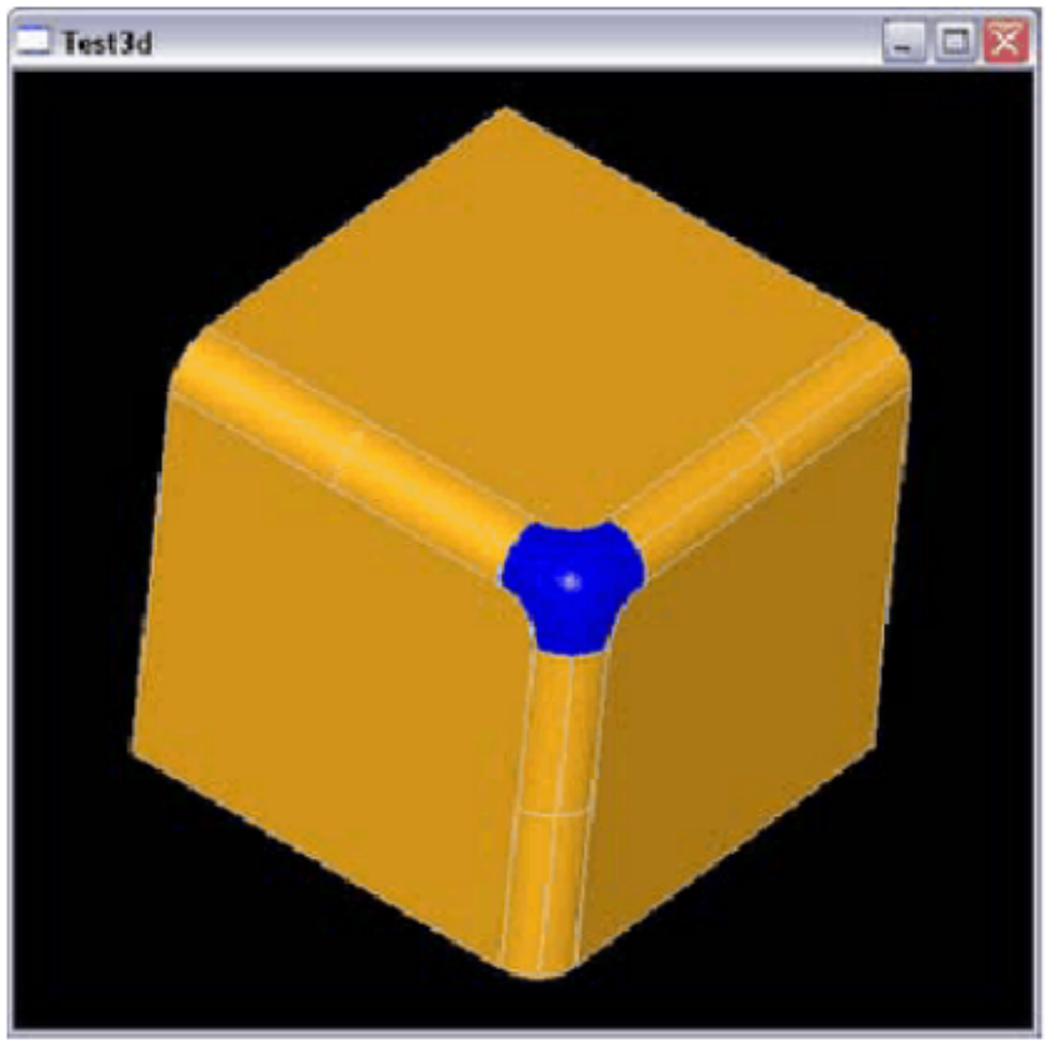

图 18 通过顶点混合生成曲面的例子

ACIS 中的网曲面定义为两组曲线，两组曲线近似垂直，从而形成网格结构。所以最终曲面要覆盖网格，就像是屋顶覆盖房梁一样。图 19 是网格结构生成曲面的例子。

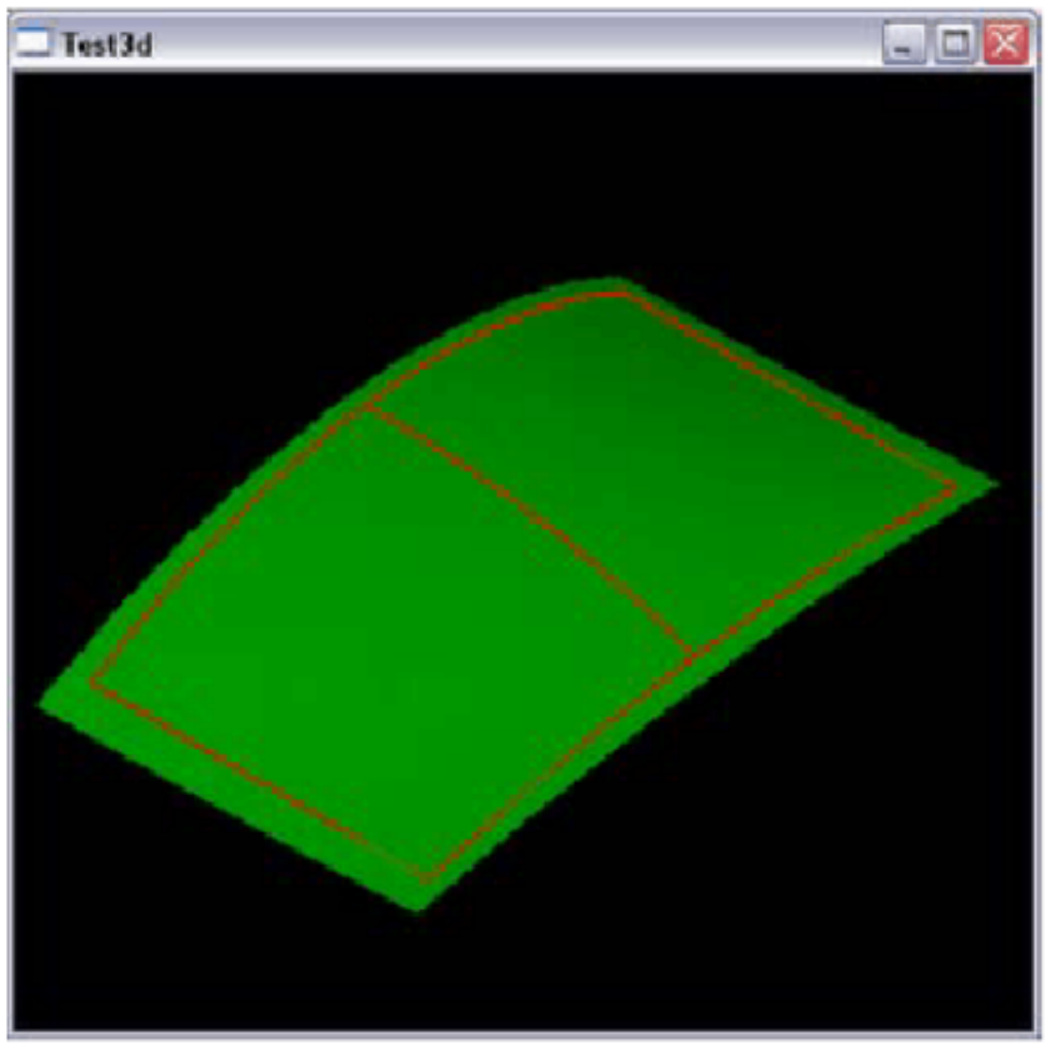

图 19 网格结构生成曲面的例子

像通常一样，下面是选自 CAD Exchanger 中的代码：

```
/*! The objects in \a theBoundaries must be of the type
Adaptor3d_HCurveOnSurface or
GeomAdaptor_HCurve indicating type of a constraint. Otherwise an exception
Standard_TypeMismatch is thrown.

If the \a theBoundaries list is empty then Standard_ConstructionError is
thrown.

If the algorithm fails returns a null surface.
*/
Handle(Geom_Surface) ACISAlgo::MakeSurface (const
TColStd_ListOfTransient& theBoundaries,
const Standard_Real theTol,
const Standard_Integer theNbPnts,
const Standard_Integer theNbIter,
const Standard_Integer theMaxDeg)
{
//constants for algorithm
const Standard_Integer aNbIter = theNbIter; //number of algorithm
iterations
const Standard_Integer aNbPnts = theNbPnts; //sample points per each
constraint
const Standard_Integer aDeg = 3; //requested surface degree ?
const Standard_Integer aMaxDeg = theMaxDeg;
```

```
const Standard_Integer aMaxSeg = 10000;
const Standard_Real aTol3d = 1.e-04;
const Standard_Real aTol2d = 1.e-05;
const Standard_Real anAngTol = 1.e-02; //angular
const Standard_Real aCurvTol = 1.e-01; //curvature

Handle(Geom_Surface) aRes;
GeomPlate_BuildPlateSurface aPlateBuilder (aDeg, aNbPnts, aNbIter, aTol2d,
aTol3d,
anAngTol, aCurvTol);

TColStd_ListIteratorOfListOfTransient anIt (theBoundaries);
if (anIt.More()) {
int i = 1;
for (; anIt.More(); anIt.Next(), i++) {
const Handle(Standard_Transient)& aCur = anIt.Value();
if (aCur.IsNull()) {
assert (0);
Standard_ConstructionError::Raise ("ACISAlgo::MakeSurface()");
} else if (aCur->IsKind (STANDARD_TYPE (Adaptor3d_HCurveOnSurface)))
{
//G1 constraint
const Handle(Adaptor3d_HCurveOnSurface)& aHCOS =
Handle(Adaptor3d_HCurveOnSurface)::DownCast (aCur);
Handle (GeomPlate_CurveConstraint) aConst =
new GeomPlate_CurveConstraint (aHCOS, 1/*GeomAbs_G1*/,
aNbPnts, aTol3d, anAngTol, aCurvTol);
aPlateBuilder.Add (aConst);
} else if (aCur->IsKind (STANDARD_TYPE (GeomAdaptor_HCurve))) {
//G0 constraint
const Handle(GeomAdaptor_HCurve)& aHC =
Handle(GeomAdaptor_HCurve)::DownCast (aCur);
Handle (GeomPlate_CurveConstraint) aConst =
new GeomPlate_CurveConstraint (aHC, 0 /*GeomAbs_G0*/, aNbPnts,
aTol3d);
aPlateBuilder.Add (aConst);
} else {
Standard_TypeMismatch::Raise ("ACISAlgo::MakeSurface()");
}
}
} else {
Standard_ConstructionError::Raise ("ACISAlgo::MakeSurface()");
}
```

```
//construct
aPlateBuilder.Perform();

if (!aPlateBuilder.IsDone()) {
return aRes;
}

const Handle(GeomPlate_Surface)& aPlate = aPlateBuilder.Surface();
//approximation (see BRepFill_Filling - when no initial surface was given)
Standard_Real aDMax = aPlateBuilder.G0Error();
TColgp_SequenceOfXY aS2d;
TColgp_SequenceOfXYZ aS3d;
aPlateBuilder.Disc2dContour (4, aS2d);
aPlateBuilder.Disc3dContour (4, 0, aS3d);
Standard_Real aMax = Max (aTol3d, 10. * aDMax);
GeomPlate_PlateG0Criterion aCriterion (aS2d, aS3d, aMax);
{
//data races in AdvApp2Var used by GeomApprox_Surface, use global mutex
Standard_Mutex::Sentry aSentry (theBSMutex);
GeomPlate_MakeApprox aMakeApprox (aPlate, aCriterion, aTol3d, aMaxSeg,
aMaxDeg);
aRes = aMakeApprox.Surface();
}
return aRes;
}
```

上面的算法只能够处理曲线或者具有相切约束的曲线，你可以用同样的方法添加点约束。

算法首先生成初始近似曲面，你可以预先设定一个曲面或者平面。我还没有碰到需要预指定初始曲面的情况，可能在非常复杂的情况下，这样做会给算法一些启发。假如真有人碰到了这种情况，也是一件非常有趣的事情。

待续... ...

另外这节是在美国西海岸 Hillsboro, Oregon 的一个旅馆里面写的。这的自然景观是我见过的最漂亮的。以后每次回 Oregon 都要好好欣赏一番……

中英文对照表

B-Spline surface B-样条曲面
final approximation 最终逼近
offset surface 偏移曲面
procedural surface 过程曲面
trimmed surface 剪裁曲面