**ПРИМЕР №9**

Задача: реализация 3D модели Сеточная поверхность (Лист изогнутый)

**Поэтапное построение эскиза.**

1. Создаем локальную СК:  
   MbPlacement3Dpl;
2. Затем создаем массив точек, вершин ломаной, и объявляем их.

Создаем массив точек:

SArray<MbCartPoint>arrPnts(19);(возможно, тут надо описать, какое слово что значит)

Добавляем в массив точки:

arrPnts.Add(MbCartPoint(30, 40));

arrPnts.Add(MbCartPoint(30, 20));

arrPnts.Add(MbCartPoint(0, 20));

arrPnts.Add(MbCartPoint(0, 0));

arrPnts.Add(MbCartPoint(100, 0));

arrPnts.Add(MbCartPoint(100, 20));

arrPnts.Add(MbCartPoint(70, 20));

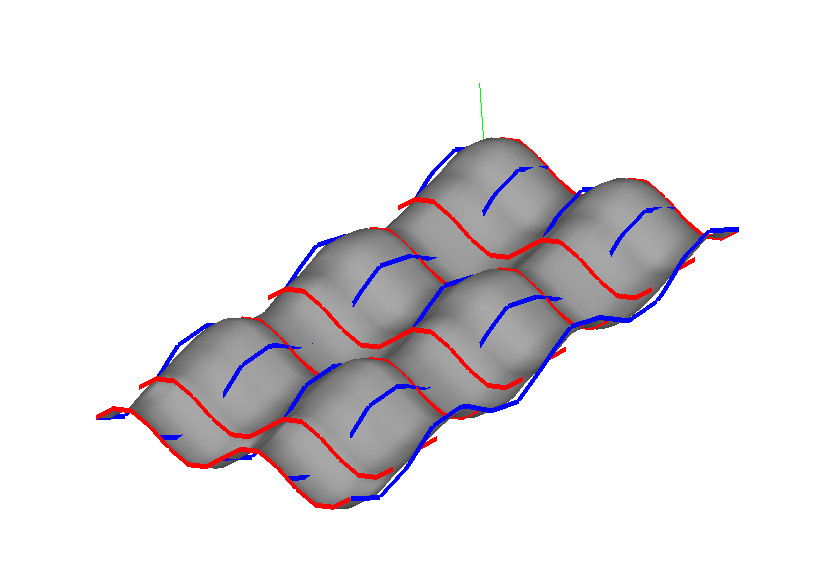
arrPnts.Add(MbCartPoint(70, 40));

1. Создаем ломаную:

MbPolyline\* pPolyline = newMbPolyline(arrPnts, false/\* Флаг незамкнутой линии \*/);

1. Создаем фаску:  
   ChamferPolyContour(pPolyline, 5,5, false, false, arrPnts[4]);
2. Создаем контур:  
   MbContour\* pContour = newMbContour(\*pPolyline, true);
3. Создаем скругление:  
   FilletPolyContour(pPolyline, 5, false, arrPnts[2], pContour);
4. Создаем дугу (Арку):  
   MbCartPointarcCenter(50, 40);   
   const double RADIUS = 20;  
   MbArc\* pArc = new MbArc(arcCenter, RADIUS, arrPnts[7], arrPnts[0], 1 /\*initSense\*/);
5. Добавляем арку в контур:  
   pContour->AddSegment(pArc);
6. Добавляем контур в сцену:  
   viewManager->AddObject(Style(1, RGB(0, 0, 255)), pContour, &pl);
7. Уменьшаем счетчик ссылок динамически созданных объектов ядра:  
   ::DeleteItem(pPolyline);  
   ::DeleteItem(pArc);  
   ::DeleteItem(pContour);

**Результат построения.**



**Код программы.**

void MakeUserCommand2()

{

// Первое семейство координатных кривых:

// - кривые Безье, представляющие два периода синусоиды (растянутые

// на отрезок U\_SIN\_DX).

// - U\_CURVES\_CNT кривых расположены вдоль оси Z на равных промежутках

// в интервале шириной U\_DZ

const int U\_CURVES\_CNT = 10;

const double U\_DZ = 30;

const int U\_SIN\_PERIOD\_CNT = 2;

const double U\_SIN\_DX = 15;

const int U\_BEZIER\_PNT\_CNT = 15;

RPArray<MbCurve3D> arrCurvesU;

for (int i = 0; i < U\_CURVES\_CNT; i++)

{

SArray<MbCartPoint3D> arrPnts;

for (int n = 0; n < U\_BEZIER\_PNT\_CNT; n++)

{

MbCartPoint3D p;

p.x = n \* U\_SIN\_DX / (U\_BEZIER\_PNT\_CNT - 1);

p.y = sin(p.x / U\_SIN\_DX \* 2 \* M\_PI \* U\_SIN\_PERIOD\_CNT);

p.z = i \* U\_DZ / (U\_CURVES\_CNT - 1);

arrPnts.push\_back(p);

}

MbBezier3D\* pc = new MbBezier3D(arrPnts, false);

arrCurvesU.Add(pc);

}

// Второе семейство координатных кривых:

// - кривые Безье, представляющие три периода синусоиды (растянутые

// на отрезок V\_SIN\_DZ).

// - V\_CURVES\_CNT кривых расположены вдоль оси X на равных промежутках

// в интервале шириной V\_DX

const int V\_CURVES\_CNT = 5;

const double V\_DX = U\_SIN\_DX;

const int V\_SIN\_PERIOD\_CNT = 3;

const double V\_SIN\_DZ = U\_DZ;

const int V\_BEZIER\_PNT\_CNT = 15;

RPArray<MbCurve3D> arrCurvesV;

for (int i = 0; i < V\_CURVES\_CNT; i++)

{

SArray<MbCartPoint3D> arrPnts;

for (int n = 0; n < V\_BEZIER\_PNT\_CNT; n++)

{

MbCartPoint3D p;

p.x = i \* V\_DX / (V\_CURVES\_CNT - 1);

p.z = n \* V\_SIN\_DZ / (V\_BEZIER\_PNT\_CNT - 1);

p.y = sin(p.z / V\_SIN\_DZ \* 2 \* M\_PI \* V\_SIN\_PERIOD\_CNT);

arrPnts.push\_back(p);

}

MbBezier3D\* pc = new MbBezier3D(arrPnts, false);

arrCurvesV.Add(pc);

}

// Построение поверхности

MbMeshSurface\* pSurf = new MbMeshSurface(arrCurvesU, arrCurvesV,

false, false, false);

// Отображение поверхности

viewManager->AddObject(Style(1, LIGHTGRAY), pSurf);

// Отображение координатных кривых

for (int i = 0, iLim = arrCurvesU.size(); i < iLim; i++)

viewManager->AddObject(Style(5, LIGHTRED), arrCurvesU[i]);

for (int i = 0, iLim = arrCurvesV.size(); i < iLim; i++)

viewManager->AddObject(Style(5, LIGHTBLUE), arrCurvesV[i]);

// Уменьшение счетчика ссылок динамически созданных объектов ядра

::DeleteItem(pSurf);

for (int i = 0, iLim = arrCurvesU.size(); i < iLim; i++)

::DeleteItem(arrCurvesU[i]);

for (int i = 0, iLim = arrCurvesV.size(); i < iLim; i++)

::DeleteItem(arrCurvesV[i]);

}