Construcción de una malla por revolución en 3D

P. J. Martín, A. Gavilanes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid

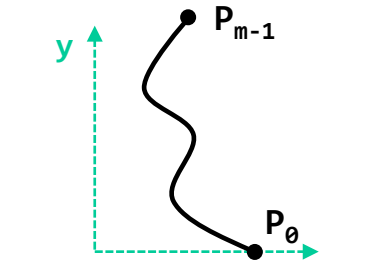
Construcción de una malla por revolución

```
int m= …; //número de puntos en el perfil original
PV3D** perfil= new PV3D*[m]; //perfil original en el plano XY
Construir perfil;
int n= …; //número de rotaciones

//Tamaños de los arrays
int numVertices= n*m;
int numCaras= n*(m-1);
int numNormales= numCaras; //1 normal por cara

//Creación de los arrays
vertice = new PV3D*[numVertices];
normal = new PV3D*[numNormales];
cara = new Cara*[numCaras];

//Colocar el perfil original en la tabla de vertices
for (int j=0; j<m; j++) vertice[j] = perfil[j]->clona();
```



Perfil en el plano XY

Construcción de una malla por revolución

```
//Vertices de la malla
for (int i=1; i<n; i++){ //generar el perfil i-ésimo
  double theta= i*360/(double)n;
  double c= cos(theta);
  double s= sin(theta);
  //R_y es la matriz de rotación sobre el eje Y
  for (int j=0; j<m; j++) {
        int indice = i*m+j;
        //Transformar el punto j-ésimo del perfil original
        double x=  c*perfil[j]->x + s*perfil[j]->z;
        double z= -s*perfil[j]->x + c*perfil[j]->z;
        PV3D* p= new PV3D(x, perfil[j]->y, z, 1);
        vertice[indice]= p;
   } //for
} //for
```

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Construcción de una malla por revolución

```
//Construcción de las caras
int indiceCara = 0;
for (int i=0; i<n; i++){ //unir el perfil i-ésimo con el (i+1)%n-ésimo
  for (int j=0; j<m-1; j++) { //esquina inferior-izquierda de una cara
     // indiceCara = i*(m-1) + j;
     int indice = i*m+j;
     VerticeNormal** vn = new VerticeNormal*[4];
     vn[0]=new VerticeNormal(indice,indiceCara);
     vn[1]=new VerticeNormal((indice+m)%numVertices,indiceCara);
     vn[2]=new VerticeNormal((indice+1+m)%numVertices,indiceCara);
     vn[3]=new VerticeNormal(indice+1,indiceCara);
     cara[indiceCara] = new Cara(4, vn);

     PV3D* v= CalculoVectorNormalPorNewell(cara[indiceCara]); //Newell
     normal[indiceCara]= v;

     indiceCara++;
  } //for
} //for
```