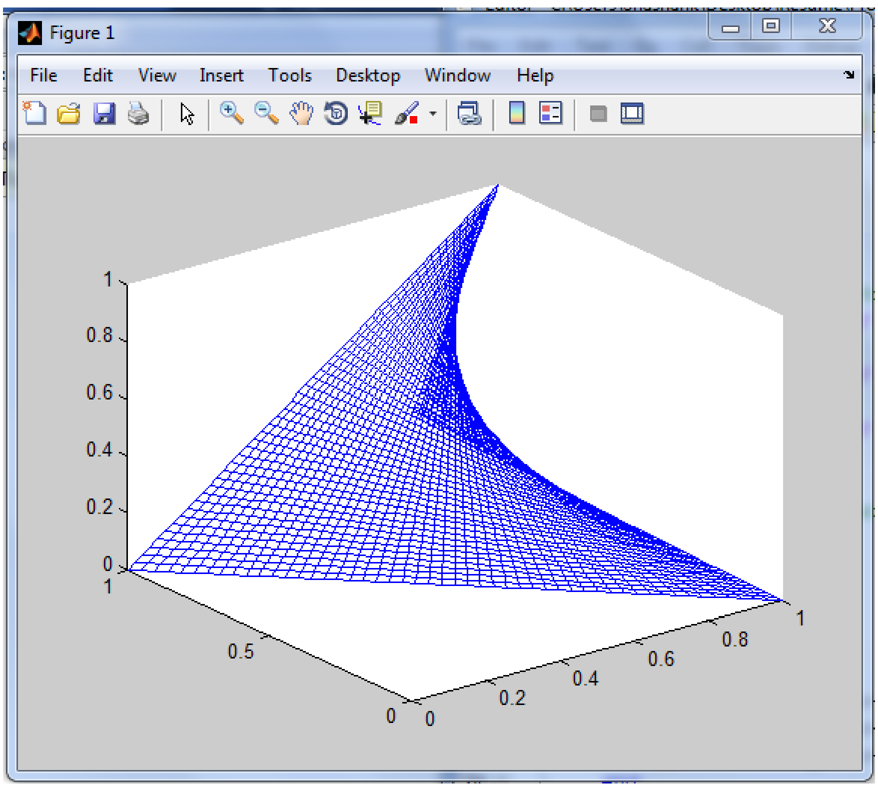
Bilinear Surface

$$Q(u,w) = P(0,0)(1-u)(1-w) + P(0,1)(1-u)w + P(1,0)u(1-w) + P(1,1)uw$$

$$Q(u,w) = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} P(0,0) & P(0,1) \\ P(1,0) & P(1,1) \end{bmatrix} \begin{bmatrix} 1-w \\ w \end{bmatrix}$$

```
%Bilinear Surface
% Input x,y,z coordinate for points,tangents andtwist vectors.
xl=xlsread('Geometric Coff', -1);
X=xl;
xl=xlsread('Geometric Coff', -1);
Y=xl;
xl=xlsread('Geometric Coff', -1);
Z=xl;

% Generates U(Parameter matrix) b/t parameter 0 to 1
min_lim=.02;
for u=0:min_lim:1
    U =[1-u u];
    for v=0:min_lim:1
        V =[1-v
            v];
        Rx(int8(u/min_lim)+1,int8(v/min_lim)+1)=U*X*V;
        Ry(int8(u/min_lim)+1,int8(v/min_lim)+1)=U*Y*V;
        Rz(int8(u/min_lim)+1,int8(v/min_lim)+1)=U*Z*V;
    end
end

% Plotting the Surf
for i=1:1:(1/min_lim)+1
    line(Rx(:,i),Ry(:,i),Rz(:,i));
    line(Rx(i,:),Ry(i,:),Rz(i,:));
    hold on
end
view(3);
```

X COORD			Y COORD			Z COORD	
0	1		0	1		1	1
1	0		0	1		0	0



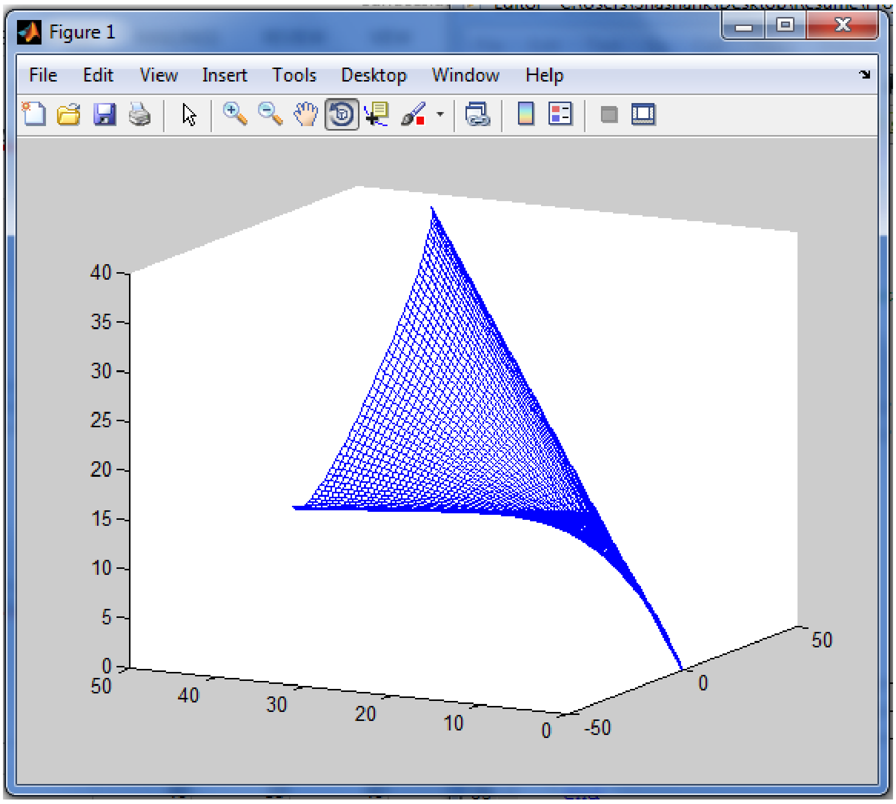
Ruled Surface

$$Q(u,w) = P(u,0)(1-w) + P(u,1)w$$

$$[\,Q\,] = [\,x(u,w) \quad y(u,w) \quad z(u,w)\,] = [\,1-w \quad w\,]\begin{bmatrix} P(u,0) \\ P(u,1) \end{bmatrix}$$

```
%Ruled Surface
% M= square Hermite matrix
M=[2 -2 1 1
    -3 3 -2 -1
    0 0 1 0
    1 0 0 0];

% U*M*B ;;; where U-Paremetric matrix, B-Geometric Coff. Matrix
% Input geometric coff for the 2 Hermite curves.
xl=xlsread('Geometric Coff', -1);
P1=xl;
xl=xlsread('Geometric Coff', -1);
P2=xl;

% Generates U(Parameter matrix) b/t parameter 0 to 1
min_lim=.02;
for u=0:min_lim:1
    U =[u^3 u^2 u 1];
    p_u0=transpose(U*M*P1);
    p_u1=transpose(U*M*P2);
    for v=0:min_lim:1
        V =[1-v v];
        x=V*[p_u0(1,:)
            p_u1(1,:)];
        y=V*[p_u0(2,:)
            p_u1(2,:)];
        z=V*[p_u0(3,:)
            p_u1(3,:)];
        Rx(int8(u/min_lim)+1,int8(v/min_lim)+1)=x;
        Ry(int8(u/min_lim)+1,int8(v/min_lim)+1)=y;
        Rz(int8(u/min_lim)+1,int8(v/min_lim)+1)=z;
    end
end

% Plotting of Surface
for i=1:1:(1/min_lim)+1
    line(Rx(:,i),Ry(:,i),Rz(:,i));
    line(Rx(i,:),Ry(i,:),Rz(i,:));
    hold on
end
view(3);
```

Curve 1				Curve2		
0	0	0		40	39	40
1	10	15		22	50	10
5	3	8		6	-4	-8
9	2	1		-1	7	9



Coons Bicubic Surface

$$Q(u,w) = [\,F_1(u) \quad F_2(u) \quad F_3(u) \quad F_4(u)\,] \times \begin{bmatrix} P(0,0) & P(0,1) & P_w(0,0) & P_w(0,1) \\ P(1,0) & P(1,1) & P_w(1,0) & P_w(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uw}(0,0) & P_{uw}(0,1) \\ P_u(1,0) & P_u(1,1) & P_{uw}(1,0) & P_{uw}(1,1) \end{bmatrix} \begin{bmatrix} F_1(w) \\ F_2(w) \\ F_3(w) \\ F_4(w) \end{bmatrix}$$

where

$$[\,F\,] = [\,F_1(t) \quad F_2(t) \quad F_3(t) \quad F_4(t)\,] = [\,T\,][\,N\,]$$
$$= [\,t^3 \quad t^2 \quad t \quad 1\,] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Thus:-

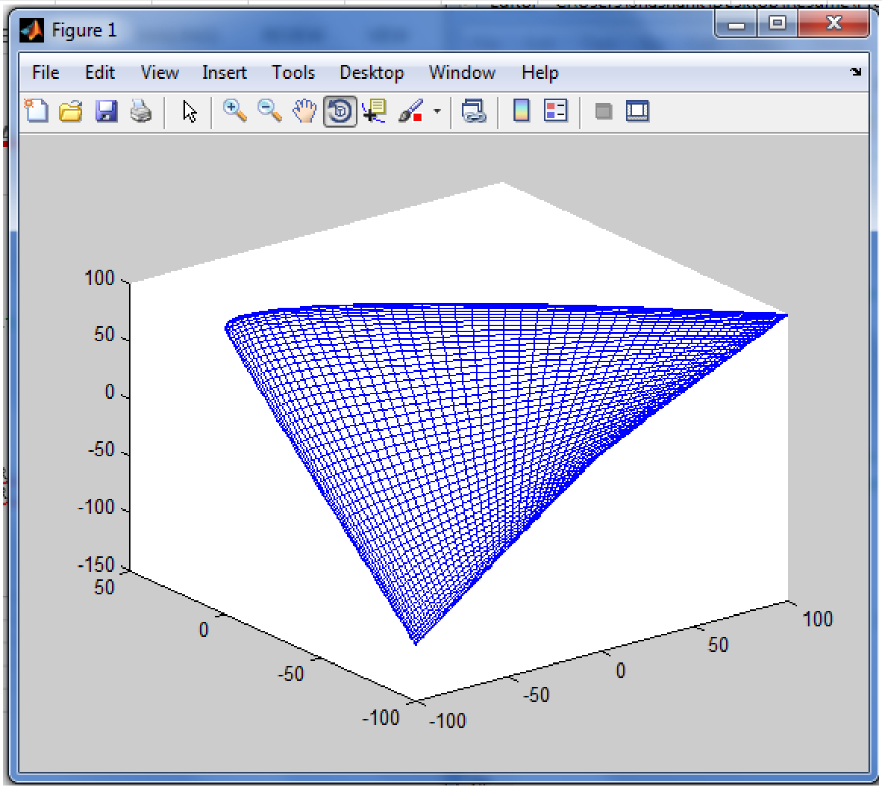
$$Q(u,w) = [\,U\,][\,N\,][\,P\,][\,N\,]^T[\,W\,]$$

And

$$[\,P\,] = \begin{bmatrix} \text{corner} & \vdots & w-\text{tangent} \\ \text{position} & \vdots & \text{vectors} \\ \text{vectors} & \vdots & \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ u-\text{tangent} & \vdots & \text{twist} \\ \text{vectors} & \vdots & \text{vectors} \end{bmatrix}$$

```
%COONS BiCubic Surface Patch
% M= square Hermite matrix
M=[2 -2 1 1
    -3 3 -2 -1
    0 0 1 0
    1 0 0 0];

% Result = U*M*B ;;; where U-Paremetric matrix, B-Geometric Coff. Matrix
% Generates U and V(Parameter matrix) b/t parameter 0 to 1
U=[];
V=[];
min_lim=.02;
for u=0:min_lim:1
    U =cat(1,U,[u^3 u^2 u 1]);
    V=cat(2,V,[u^3
        u^2
        u
        1]);
end

% Input x,y,z coordinate for points,tangents andtwist vectors.
xl=xlsread('Geometric Coff', -1);
X=xl;
xl=xlsread('Geometric Coff', -1);
Y=xl;
xl=xlsread('Geometric Coff', -1);
Z=xl;

% Computation of Resultant coordinate Matrices
R_x=U*M*X*transpose(M)*V;
R_y=U*M*Y*transpose(M)*V;
R_z=U*M*Z*transpose(M)*V;

% Plotting of the Surface
for i=1:1:(1/min_lim)+1
    line(R_x(:,i),R_y(:,i),R_z(:,i));
    line(R_x(i,:),R_y(i,:),R_z(i,:));
    hold on
end
view(3);
```

X COORD					Y COORD					Z COORD			
-100	-100	0	0		0	-100	10	-1		100	-100	-10	-1
100	100	0	0		-100	0	1	-1		100	-100	-1	-1
100	1	0	0.1		100	1	0	0.1		0	0	0	0.1
1	1	0.1	0		-1	-1	-0.1	0		0	0	-0.1	0



Bezier Surface

$$Q(u,w) = [\,U\,][\,N\,][\,B\,][\,M\,]^T[\,W\,]$$

where

$$[\,U\,] = [\,u^n \quad u^{n-1} \quad \cdots \quad 1\,]$$

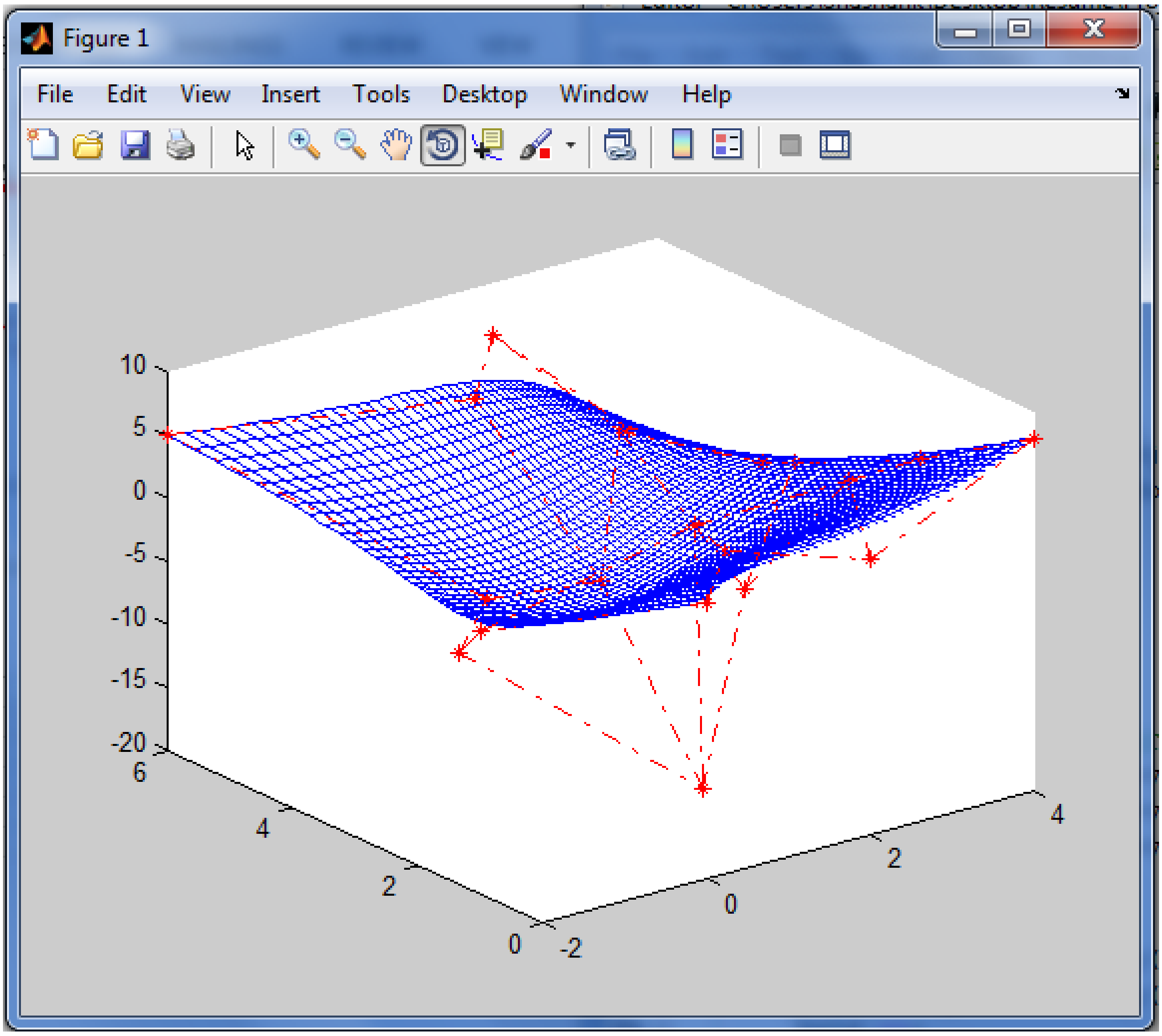
$$[\,W\,] = [\,w^m \quad w^{m-1} \quad \cdots \quad 1\,]^T$$

$$[\,B\,] = \begin{bmatrix} B_{0,0} & \cdots & B_{0,m} \\ \vdots & \ddots & \vdots \\ B_{n,0} & \cdots & B_{n,m} \end{bmatrix}$$

```
%Bezier Surf
% Input of X,Y,Z COORD in 3 different MATRICES.
xl=xlsread('Control Points', -1);
Px=xl;
xl=xlsread('Control Points', -1);
Py=xl;
xl=xlsread('Control Points', -1);
Pz=xl;
[Nx,Ny]=size(xl);
Nx=Nx-1;
Ny=Ny-1;

% Calculation of PARAMETER MATRICES
min_para=.02;
U=[];
for i=0:min_para:1
    xyz=[];
    for j=Nx:-1:0
        xyz=cat(2,xyz,i^j);
    end
    U=cat(1,U,xyz);
end

V=[];
for i=0:min_para:1
    xyz=[];
    for j=Ny:-1:0
        xyz=cat(2,xyz,i^j);
    end
    V=cat(1,V,xyz);
end
V=transpose(V);

% Calculation of 2 different BEZIER BASIS FUNCTIONS
N=[];
for i=0:1:Nx
    for j=0:1:Nx
        if (i+j)>=0 && (i+j)<=Nx
            N(i+1,j+1)=nchoosek(Nx,j)*nchoosek(Nx-j,Nx-i-j)*(-1)^(Nx-i-j);
        else
            N(i+1,j+1)=0;
        end
    end
end

M=[];
for i=0:1:Ny
    for j=0:1:Ny
        if (i+j)>=0 && (i+j)<=Ny
            M(i+1,j+1)=nchoosek(Ny,j)*nchoosek(Ny-j,Ny-i-j)*(-1)^(Ny-i-j);
        else
            M(i+1,j+1)=0;
        end
    end
end
```

```
% CALCULATION of the RESULT MATRICES and PLOTTING of answer
Bez_x=U*N*Px*transpose(M)*V;
Bez_y=U*N*Py*transpose(M)*V;
Bez_z=U*N*Pz*transpose(M)*V;

for i=1:1:(1/min_para)+1
    plot3(Bez_x(i,:),Bez_y(i,:),Bez_z(i,:));
    plot3(Bez_x(:,i),Bez_y(:,i),Bez_z(:,i));
    hold on;
end

for i=1:1:Nx+1
    plot3(Px(i,:),Py(i,:),Pz(i,:),'LineStyle','-.','Marker','*','color','r');
end
for i=1:1:Ny+1
    plot3(Px(:,i),Py(:,i),Pz(:,i),'LineStyle','-.','Marker','*','color','r');
end
view(3);
```

X_COORD					Y_COORD					Z_COORD			
0	1	2	4		0	1	0	0		2	2	2	8
-2	1	2.5	3		1	1.5	1	0.5		1	3	5	7
-1.5	1.5	2	3		2	2	2	2.5		-4	-20	-5	2
-1	1	2	3		2.2	3	4	3		-1	-5	3	1
-2	1	2	2.5		6	5	6	4.5		5	5	6	1



B-Spline Surface

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,l}(w)$$

where

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } x_i \le u < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

$$M_{j,1}(w) = \begin{cases} 1 & \text{if } y_j \le w < y_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$M_{j,l}(w) = \frac{(w - y_j)M_{j,l-1}(w)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - w)M_{j+1,l-1}(w)}{y_{j+l} - y_{j+1}}$$

```
%B-Spline Surface
clc
clear all
close all

% P-Control Point Matrix
% N_i,k= normalized B-spline basis function
% B_spline= SUMISSION 0 to Deg (P_i*N_i,k)
% order=k
% no. of pts.= n+1
% no. of knots= n+k+1

% Inputs- P(Control point matrix), Knot(Knot vector), k(order)

% Input of X,Y,Z COORD in 3 different MATRICES.
xl=xlsread('Control Points', -1);
Px=xl;
xl=xlsread('Control Points', -1);
Py=xl;
xl=xlsread('Control Points', -1);
Pz=xl;
[Nx,Ny]=size(xl);
nx=Nx-1;
ny=Ny-1;

% Knot vectors in U&V Parameter Direction
Knot1=[0 0 0 0 1 2 2 2 2];
Knot2=[0 0 1 2 3 4 5 5];

% Order in Both Direction
ku=4;
kv=4;
Bsurf_x=[];
Bsurf_y=[];
Bsurf_z=[];
```

```
% Calculation of Surface
for u= Knot1(ku-1):.05:Knot1(nx+2)-.001
    mat_x=[];
    mat_y=[];
    mat_z=[];
    for v= Knot2(kv-1):.05:Knot2(ny+2)-.001
        subx=0;
        suby=0;
        subz=0;
        for i=1:1:nx+1
            for j=1:1:ny+1
                lc=N_ik(u,i,ku,Knot1)*N_ik(v,j,kv,Knot2);
                subx=subx+lc*Px(i,j);
                suby=suby+lc*Py(i,j);
                subz=subz+lc*Pz(i,j);
            end
        end
        mat_x=cat(1,mat_x,subx);
        mat_y=cat(1,mat_y,suby);
        mat_z=cat(1,mat_z,subz);
    end
    Bsurf_x=cat(2,Bsurf_x,mat_x);
    Bsurf_y=cat(2,Bsurf_y,mat_y);
    Bsurf_z=cat(2,Bsurf_z,mat_z);
end

% PLOTTING of answer
[x,y]=size(Bsurf_x);
for i=1:1:x
    plot3(Bsurf_x(i,:),Bsurf_y(i,:),Bsurf_z(i,:));
    hold on;
end
for i=1:1:y
    plot3(Bsurf_x(:,i),Bsurf_y(:,i),Bsurf_z(:,i));
end

for i=1:1:Nx
    plot3(Px(i,:),Py(i,:),Pz(i,:),'LineStyle','-.','Marker','*','color','r');
end
for i=1:1:Ny
    plot3(Px(:,i),Py(:,i),Pz(:,i),'LineStyle','-.','Marker','*','color','r');
end
view(3);

function [ val ] = N_ik( t,i,k,Knot )
if k~=1
    val1=(t-Knot(i))*N_ik(t,i,k-1,Knot)/(Knot(i+k-1)-Knot(i));
    val2=(Knot(i+k)-t)*N_ik(t,i+1,k-1,Knot)/(Knot(i+k)-Knot(i+1));

    if isnan(val1)
        val1=0;
    end
    if isnan(val2)
        val2=0;
    end
    val=val1+val2;

else
    if t>=Knot(i) && t<Knot(i+1)
        val=1;
    else
        val=0;
    end
end
end
```

X_COORD				Y_COORD				Z_COORD			
0	1	2	4	0	1	0	0	2	2	2	8
-2	1	2.5	3	1	1.5	1	0.5	1	3	5	7
-1.5	1.5	2	3	2	2	2	2.5	-4	-20	-5	2
-1	1	2	3	2.2	3	4	3	-1	-5	3	1
-2	1	2	2.5	6	5	6	4.5	5	5	6	1