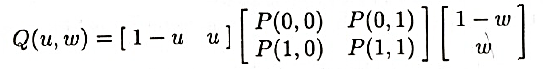
# Bilinear Surface





%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

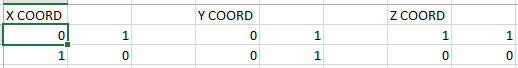
line(Rx(:,i),Ry(:,i),Rz(:,i));

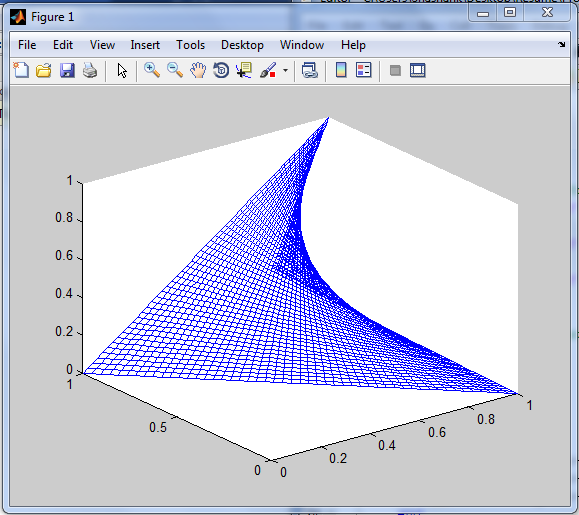
line(Rx(i,:),Ry(i,:),Rz(i,:));

hold on

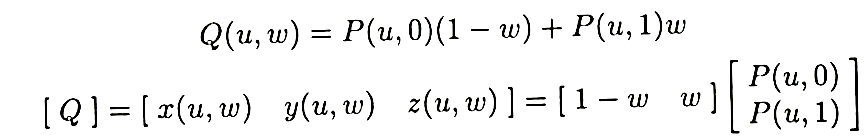
end

view(3);





# Ruled Surface



%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

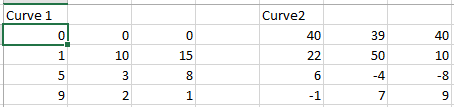
line(Rx(:,i),Ry(:,i),Rz(:,i));

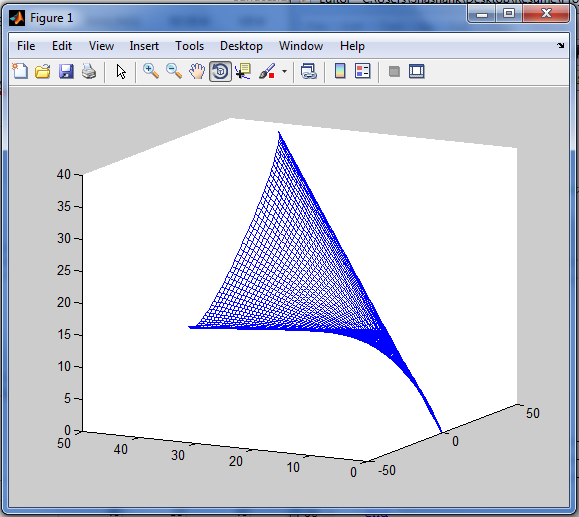
line(Rx(i,:),Ry(i,:),Rz(i,:));

hold on

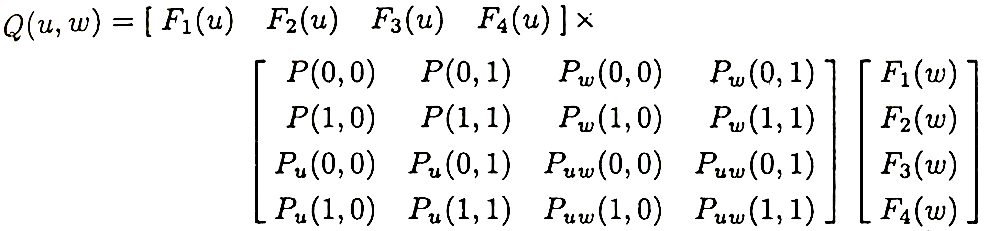
end

view(3);

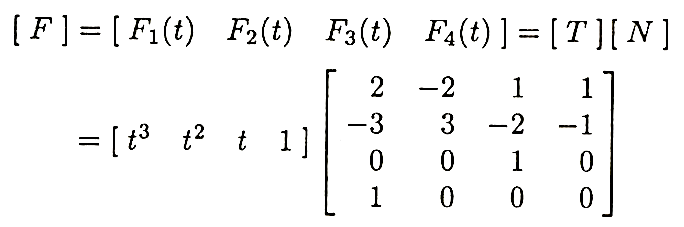




# Coons Bicubic Surface



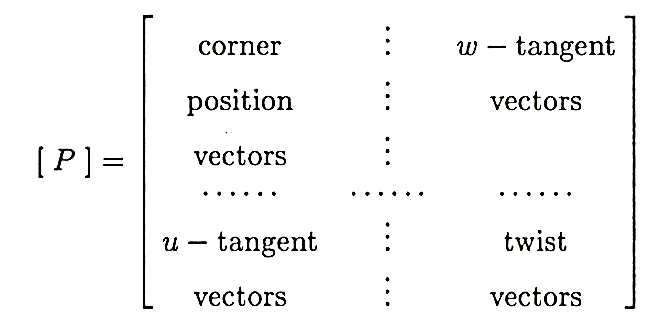
where



Thus:-



And



%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

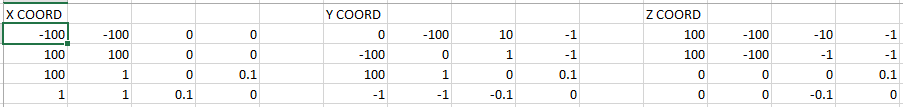
line(R\_x(:,i),R\_y(:,i),R\_z(:,i));

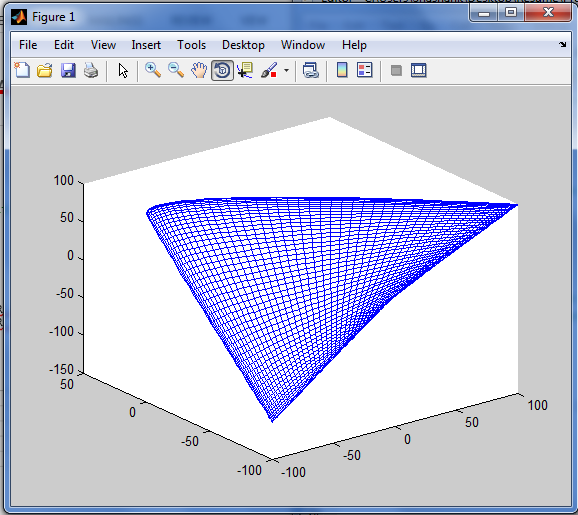
line(R\_x(i,:),R\_y(i,:),R\_z(i,:));

hold on

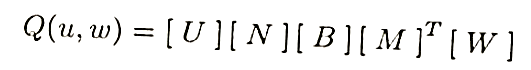
end

view(3);

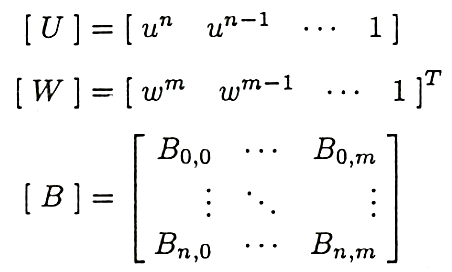




# Bezier Surface



where



%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');

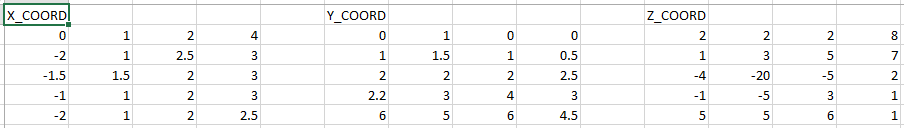
end

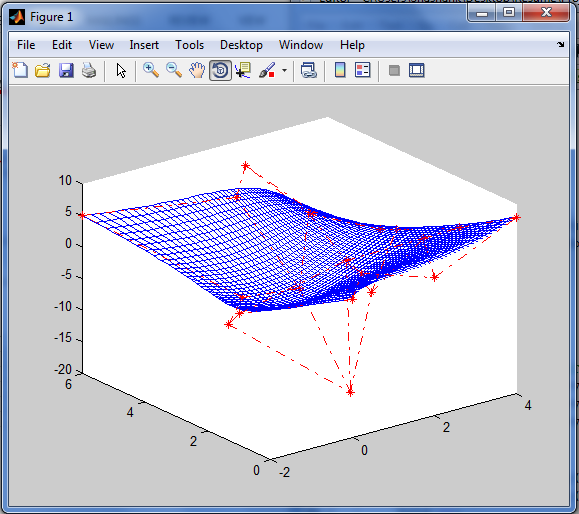
for i=1:1:Ny+1

plot3(Px(:,i),Py(:,i),Pz(:,i),'LineStyle','-.','Marker','\*','color','r');

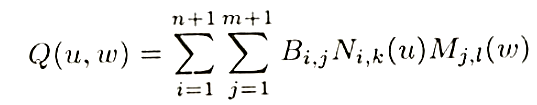
end

view(3);

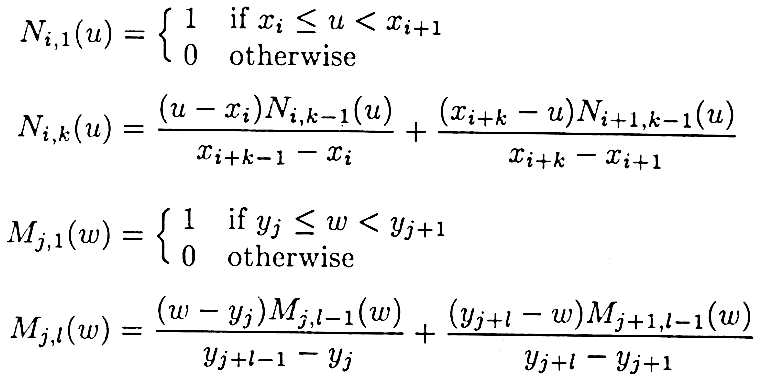




# B-Spline Surface



where



%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

