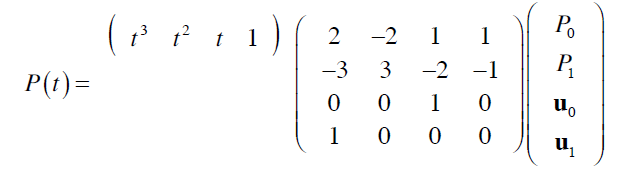
# Cubic Hermite Curves



Where Po and P1- End Points

and u0 and u1- End Tangents

% Hermite Cubic Curve

% 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

% Relation b/w Algebric and Geometric Coff.

% A = M\*B ; where A-Algebric Coff. Matrix

% Generates U(Parameter matrix) b/t parameter 0 to 1

U=[];

for u=0:.001:1

U =[U

u^3 u^2 u 1];

end

% Input B matrix for x,y,z coordinate.

xl=xlsread('Geometric Coff', -1);

B=xl;

R=U\*M\*B;

line(R(:,1),R(:,2),R(:,3));

view(3);

% Conversion to 4 point form P=Param\*M\*B where Param-default parameters

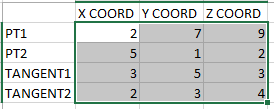
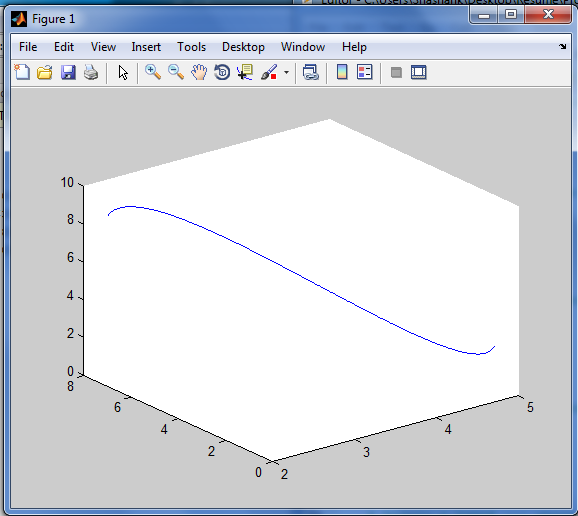
Param=[0 0 0 1

.33^3 .33^2 .33 1

.66^3 .66^2 .66 1

1 1 1 1];

P=Param\*M\*B

INPUT OUTPUT

# Cubic Curve 4 Point Form

We obtain Geometric Coefficient Matrix by satisfying the 4 Points a different parameters (Default- 0, 0.33, 0.66, 1) and inverting the equation.

%Cubic Segments 4 Point Form

% M= square Hermite matrix

M=[2 -2 1 1

-3 3 -2 -1

0 0 1 0

1 0 0 0];

% Param= Based on Default equidistant parameters

Param=[0 0 0 1

.33^3 .33^2 .33 1

.66^3 .66^2 .66 1

1 1 1 1];

% Input of Points Matrix

xl=xlsread('Four Point Form', -1);

P=xl;

% Conversion to Geometric Form--- B=inv(M)\*inv(Param)\*P

B=inv(M)\*inv(Param)\*P

% Displaying the Curve

U=[];

for u=0:.001:1

U =[U

u^3 u^2 u 1];

end

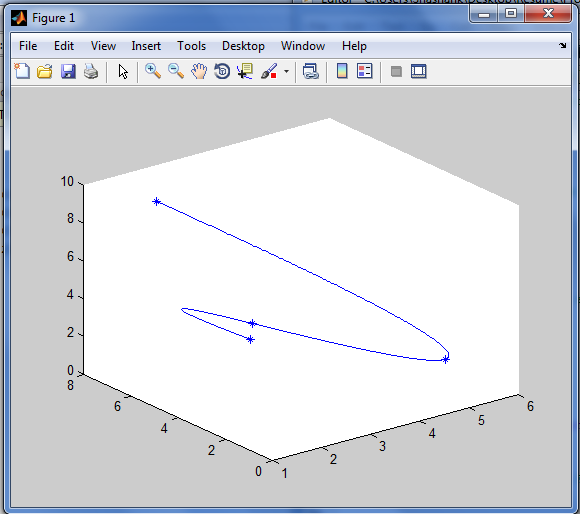
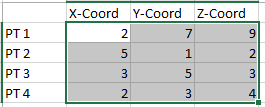
R=U\*M\*B;

line(R(:,1),R(:,2),R(:,3));

view(3);

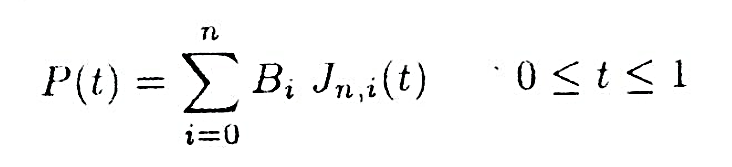
hold on;

plot3(P(:,1),P(:,2),P(:,3),'LineStyle','none','Marker','\*');

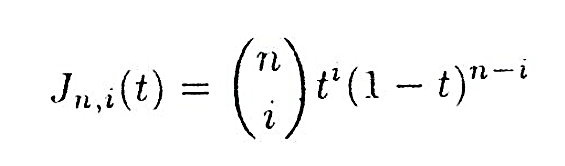


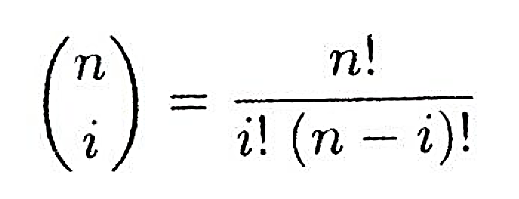
INPUT OUTPUT

# Bezier Curves



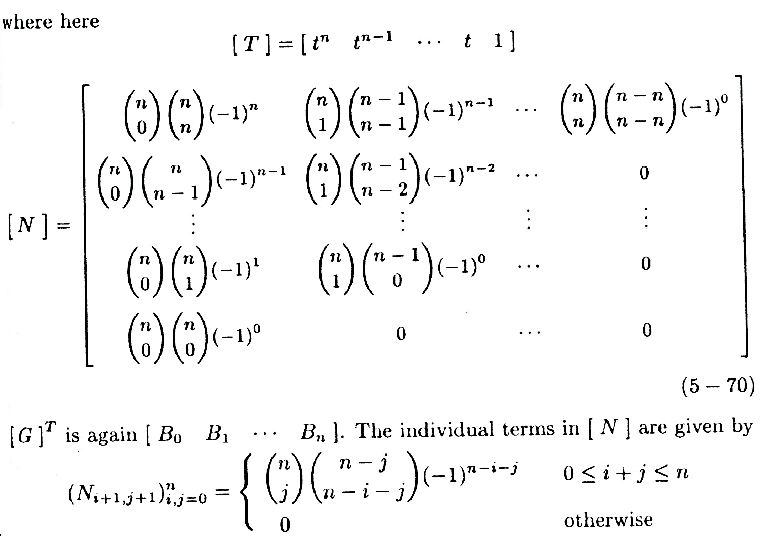
Where Bezier or Bernstein Basis or Blending Function is





Generalised Matrix Representation





%Bezier Curve

% P-Control Point Matrix

% B\_i-Bernstein Polonomial Function Value for i\_th term

% BEZ= SUMISSION 0 to Deg (B\_i\*P\_i)

xl=xlsread('Control Points', -1);

G=xl;

[Deg,dump]=size(xl);

Deg=Deg-1;

P=[];

% ONE WAY TO DO IT.

% for u=0:.001:1

% t=[0 0 0];

% for i=0:1:Deg

% B\_i=factorial(Deg)/(factorial(Deg-i)\*factorial(i))\*u^i\*(1-u)^(Deg-i);

% t=t+B\_i\*P(i+1,:);

% end

% Bez=[Bez

% t];

% end

% BETTER WAY

% P= T\*N\*G where T= parameter matrix (t^3, t^2, t, 1)

% G= Points Matrix

% N= General Bezier Basis matrix

T=[];

for i=0:.01:1

xyz=[];

for j=Deg:-1:0

xyz=cat(2,xyz,i^j);

end

T=cat(1,T,xyz);

end

N=[];

for i=0:1:Deg

for j=0:1:Deg

if (i+j)>=0 && (i+j)<=Deg

N(i+1,j+1)=nchoosek(Deg,j)\*nchoosek(Deg-j,Deg-i-j)\*(-1)^(Deg-i-j);

else

N(i+1,j+1)=0;

end

end

end

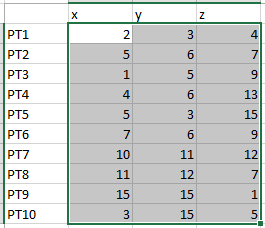
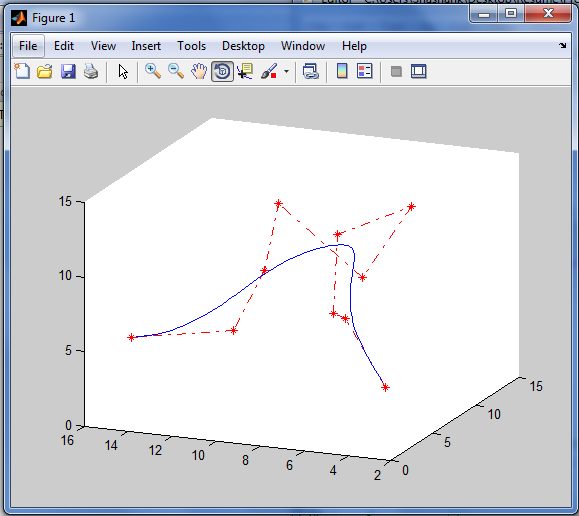
P= T\*N\*G;

line(P(:,1),P(:,2),P(:,3));

view(3);

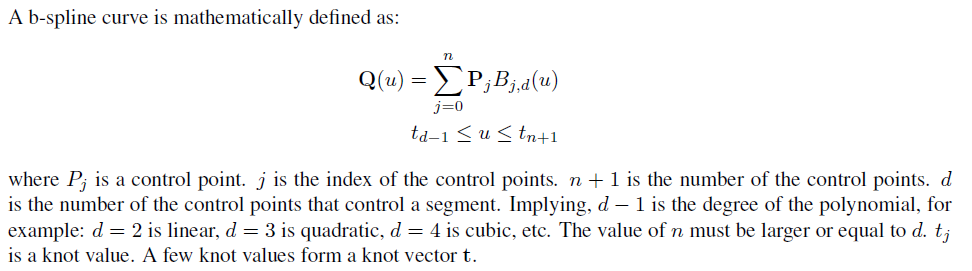
hold on;

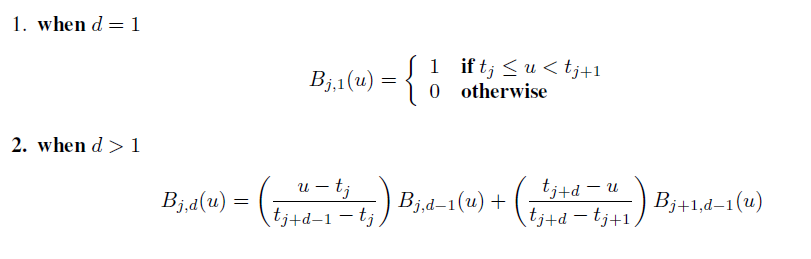
plot3(G(:,1),G(:,2),G(:,3),'LineStyle','-.','Marker','\*','color','r');

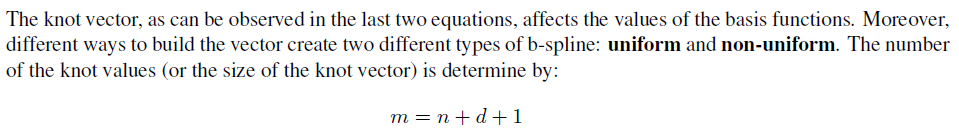
 

INPUT OUTPUT

# B-Splines







%B-Spline Curves

% 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)

xl=xlsread('Control Points', -1);

P=xl;

[n,dump]=size(P);

xl=xlsread('Control Points', -1);

Knot=xl;

k = str2num(input('Enter the order of the B-Spline (k):','s'));

% Knot=[0 0 0 0 1 2 2 2 2];

% k=4;

B\_spl=[];

% Calculating the sumission for each parameter

for t=Knot(k-1):.001:Knot(n+2)-.001

sub=[0 0 0];

for i=1:1:n

sub=sub+P(i,:)\*N\_ik(t,i,k,Knot);

end

B\_spl=[B\_spl

sub];

end

% Plotting of B-Spline

plot3(B\_spl(:,1),B\_spl(:,2),B\_spl(:,3));

view(3);

hold on;

plot3(P(:,1),P(:,2),P(:,3),'LineStyle','-.','Marker','\*','color','r');

function [ val ] = N\_ik( t,i,k,Knot )

if k~=1 % Use of Recursion

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) % For 0/0 Cases

val1=0;

end

if isnan(val2)

val2=0;

end

val=val1+val2;

else

if t>=Knot(i) && t<Knot(i+1)

val=1;

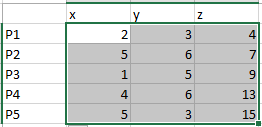
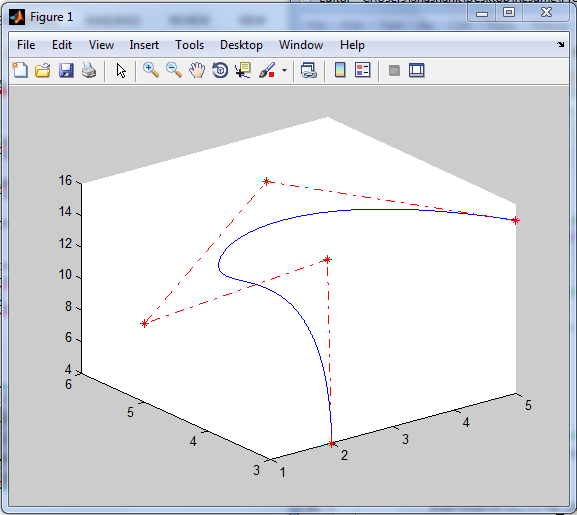
else

val=0;

end

end

end

INPUT Points OUTPUT