

```
%% Problem 2C Open
close all;
clear all;
clc;

%% Input
%% Problem 2a K>50
%% Random Interger
% Kmax = 0;
% Tmax = 0;
% closeopentype = num2str(2); % (1 for Open) (2 for Close)
% while Kmax<50
% n = 9;
% Wpt = randi([1,10],1,n+1);
% Ps4 = randi([-20,20],3,n+1);
% Ps4(end+1,:) = ones(1,n+1);
%% User Input
fprintf('Input n, Ps, W, and Type for NURBS Curve.\n');
fprintf('For OPEN type: 1\n');
fprintf('For CLOSED type: 2\n');
n = input('n (input interger>3):');
Wpt = input('W (1 by n+1 matrix all positive values):');
Ps4 = input('Ps (4 by n+1 matrix last rows is all ones):');
openclose = input('Open or Closed Value:');
closeopentype = num2str(openclose);
%% Only used for Fixed Values
% fprintf('Input n, Ps, W, and Type for NURBS Curve.\n');
% fprintf('For OPEN type: 1\n');
% fprintf('For CLOSED type: 2\n');
% fprintf('n (input interger>3):\n');
% fprintf('W (1 by n+1 matrix all positive values): auto input below\n');
% fprintf('Ps (4 by n+1 matrix last rows is all ones): auto input below\n');
% fprintf('Open or Closed Value: auto input below\n');
%% Fix Values
% Note it is a matrix of (n+1)by(1)
% Note: matrix (4)by(n+1) last row is all 1

%% Problem 2c K>50 Open
% n = [6];
% closeopentype = num2str(1);
% Wpt = [7,5,10,2,8,8,6]
% Ps4 = [-13,-15,-18,-2,-12,-2,-7;4,-12,-11,-20,-17,-16,-8;-8,16,-18,16,-8,20,-18;✓
1,1,1,1,1,1,1]

% n = 6;
% closeopentype = num2str(1);
% Wpt = [6,6,2,9,5,3,9];
% Ps4 = [11,7,-4,-9,-5,14,-7;16,7,-9,16,0,4,-2;-9,-15,9,13,8,3,9;1,1,1,1,1,1,1];
%% Problem 2c T>30 Open
% Working
% n = 7;
```

```

% closeopentype = num2str(1);
% Wpt = [7,4,7,9,2,6,6,6];
% Ps4 = [-2,2,6,-3,15,-13,5,6;19,5,14,-17,-12,7,-1,3;-13,-19,-4,3,-17,10,-8,7;
1,1,1,1,1,1,1,1];

% n = 10;
% closeopentype = num2str(1);
% Wpt = [8,7,10,2,9,6,2,10,9,1,3];
% Ps4 = [-4,-3,-3,15,5,-5,-13,16,14,-6,5;5,9,6,-15,11,-7,10,-17,-2,11,14;17,19,16,-12,
-10,-2,-16,12,4,4,-2;1,1,1,1,1,1,1,1,1,1,1];

%% Problem 2c K>50 Closed
% n = 6;
% closeopentype = num2str(2);
% Wpt = [6,3,2,4,9,10,9];
% Ps4 = [-17,-7,10,17,9,-2,15;-7,20,14,-16,9,0,7;-11,2,-14,10,-15,1,13;1,1,1,1,1,1,1];

% n = 9;
% closeopentype = num2str(2);
% Wpt = [5,7,10,2,8,6,5,9,4,2];
% Ps4 = [5,12,-15,4,16,-12,-15,-19,-9,17;5,20,-11,-16,2,-2,-1,8,-15,5;-7,20,-20,-4,
-5,19,15,20,8,16;1,1,1,1,1,1,1,1,1,1];
%% Problem 2c T>30 Closed
% n = 6;
% closeopentype = num2str(2);
% Wpt = [5,1,6,10,8,4,6];
% Ps4 = [3,-9,-4,-1,-13,-3,-12;0,0,4,7,5,-3,14;-15,-15,13,11,11,0,0;1,1,1,1,1,1,1];

% n = 8;
% closeopentype = num2str(2);
% Wpt = [8,10,2,8,2,10,9,5,8];
% Ps4 = [0,-17,-13,-19,2,1,19,18,-4;13,4,-3,18,-13,20,7,-1,8;-6,17,10,11,0,15,-4,-11,2;
1,1,1,1,1,1,1,1,1,1];
%% Output
% Hermite Curve Define in term of u
syms u
Ms4 = [-1 3 -3 1 ;...
        3 -6 0 4;...
        -3 3 3 1;...
        1 0 0 0];
U = [u^3; u^2; u; 1];
MU = Ms4*U;
k = 4;

for i = 1:n+1
    Wpmatrix(:,i) = Wpt(i)*Ps4(:,i);
end

if strcmp(closeopentype,'1')
    for i = 1:(n-k+2)
        pu(:,i) = (Wpmatrix(:,i:3+i)*MU)/(Wpt(1,i:3+i)*MU);    % P(u) function base off

```

NURBS

```
end
pt = pu;
pt_t = diff(pt,u);
pt_tt = diff(pt_t,u);
pt_ttt = diff(pt_tt,u);
% Rearrange Pu
space = 100;
u = linspace(0,1,space);
for i = 1:(n-k+2)
    Pgroup(:, (i-1)*space+1:i*space) = double(subs(pu(:,i)));
end
end

if strcmp(closeopentype, '2')
    kvalue = 4;
    for i=1:n+1
        if i+3>=length(Wpt)

            FrontWP = WPmatrix(:,i:end);
            EndWP = WPmatrix(:,1:kvalue-length(FrontWP(1,:)));
            NewWP4x4 = [FrontWP, EndWP];

            FrontWpt = Wpt(:,i:end);
            EndWpt = Wpt(:,1:kvalue-length(FrontWpt(1,:)));
            Wpt4x1 = [FrontWpt, EndWpt];

        else
            NewWP4x4 = WPmatrix(:,mod(i,n+1):mod(i+3,n+1));
            Wpt4x1 = Wpt(mod(i,n+1):mod(i+3,n+1));
        end
        pu(:,i)= (NewWP4x4*MU)/(Wpt4x1*MU);
    end
    Ps4(:,end+1) = Ps4(:,1);
    pt = pu;
    pt_t = diff(pt,u);
    pt_tt = diff(pt_t,u);
    pt_ttt = diff(pt_tt,u);
    % Rearrange Pu
    space = 100;
    u = linspace(0,1,space);
    for i = 1:n+1
        Pgroup(:, (i-1)*space+1:i*space) = double(subs(pu(:,i)));
    end
end

% Subs Here is substituting in u
pt_t_v = double(subs(pt_t));
pt_tt_v = double(subs(pt_tt));
pt_ttt_v = double(subs(pt_ttt));
```

```

pt_t_v2 = pt_t_v(1:3,:);
pt_tt_v2 = pt_tt_v(1:3,:);
pt_ttt_v2 = pt_ttt_v(1:3,:);

for i = 1:length(Pugroup)
    Kcurve(i) = norm(cross(pt_t_v2(:,i),pt_tt_v2(:,i)))/norm(pt_t_v2(:,i))^3;
    Torsion(i) = dot(pt_t_v2(:,i),cross(pt_tt_v2(:,i),pt_ttt_v2(:,i)))/norm(cross(pt_t_v2(
(:,i),pt_tt_v2(:,i)))^2;
end
%% Finding max Curvature K and Torsion T
Kvalue = abs(Kcurve);
Tvalue = abs(Torsion);
[Kmax, Klocation] = max(double(Kvalue));
[Tmax, Tlocation] = max(double(Tvalue));

Kxyz_position = Pugroup(1:3,Klocation);
Txyz_position = Pugroup(1:3,Tlocation);

% Max K and T.
Kmax = Kmax;
Tmax = Tmax;
% end

%% Plot
plot3(Pugroup(1,:),Pugroup(2,:),Pugroup(3,:), 'b', 'linewidth',2);
grid on
hold on
plot3(Kxyz_position(1),Kxyz_position(2),Kxyz_position(3), 'r*', 'linewidth',2);
plot3(Txyz_position(1),Txyz_position(2),Txyz_position(3), 'ro', 'linewidth',2);
plot3(Ps4(1,:),Ps4(2,:),Ps4(3,:), 'm-');
plot3(Ps4(1,:),Ps4(2,:),Ps4(3,:), 'm*');

xlabel('x'); ylabel('y'); zlabel('z')
title('NURBS Curve with k=4')
legend('NURBS Curve','K_M_A_X','T_M_A_X','Control Polygon','Control Pts');

fprintf('This program output are n, Ps, W and Type. Then plot the NURBS.\n\n')
fprintf('Type (Open=1) (Close=2) :'); disp(closeopentype)
fprintf('n ='); disp(n)
fprintf('W =\n'); disp(Wpt)
if strcmp(closeopentype,'1')
    fprintf('Ps =\n'); disp(Ps4)
else
    fprintf('Ps =\n'); disp(Ps4(:,1:end-1))
end
fprintf('Max Curvature:'); disp(Kmax)
fprintf('Max Curvature x,y,z position =\n'); disp(Kxyz_position)

fprintf('Max Torsion:'); disp(Tmax)
fprintf('Max Torsion x,y,z position =\n'); disp(Txyz_position)

```

```
fprintf('Check graph in plot.\n');  
fprintf('For Numerical Values of type in desire values base off Workspace:\n')
```