

```

CSCI3656_ProblemSet9_2.m
% Program 9.2, used for 3D interpolation of the
% Worthington glacier near Valdez, Alaska

% Uses Bezier Curve Program found online at:
%
http://www.mathworks.com/matlabcentral/fileexchange/7441-bezier-interpolation-in-n-dimension-space
% Credit goes to Dr. Murtaza Khan for the Bezier function, drkhanmurtaza@gmail.com

function [ plot ] = CSCI3656_ProblemSet9_2( plot )

%           Top                               Bottom
% (33.44 87.93 105.88)      (15.59 35.07 12.88)
% (8.81 84.07 103.11)      (38.57 37.17 13.33)
% (15.62 34.83 105.98)      (61.10 67.15 17.31)
% (40.16 38.71 108.13)      (58.97 92.05 19.09)
% (61.45 67.07 108.12)      (36.98 63.24 16.51)
% (58.81 91.44 107.72)      (64.45 42.66 20.01)
% (36.97 63.29 107.14)      (89.18 46.85 27.71)
% (64.71 42.38 109.07)      (66.87 18.48 14.24)
% (89.11 46.49 109.93)      (65.90 31.93 21.0)
% (67.24 18.32 109.99)      (76.55 44.51 22.0)
% (65.90 31.93 109.51)
% (76.55 44.51 109.91)

label0 = 'Shape';
label1 = 'Bezier Curve';
label2 = 'Control Polygon';
label3 = 'Control Point';

top = [33.44 87.93 105.88; 8.81 84.07 103.11; 15.62 34.83 105.98; 40.16 38.71
108.13; 61.45 67.07 108.12; 58.81 91.44 107.72; 36.97 63.29 107.14; 64.71 42.38
109.07; 89.11 46.49 109.93; 67.24 18.32 109.99; 65.90 31.93 109.51; 76.55 44.51
109.91];
topX = [33.44, 8.81, 15.62, 40.16, 61.45, 58.81, 36.97, 64.71, 89.11, 67.24, 65.90,
76.55];
topY = [87.93, 84.07, 34.83, 38.71, 67.07, 91.44, 63.29, 42.38, 46.49, 18.32,
31.93, 44.51];
topZ = [105.88, 103.11, 105.98, 108.13, 108.12, 107.72, 107.42, 109.07, 109.93,
109.99, 109.51, 109.91];

bottom = [15.59 35.07 12.88; 38.57 37.17 13.33; 61.10 67.15 17.31; 58.97 92.05
19.09; 36.98 63.24 16.5; 64.45 42.66 20.01; 89.18 46.85 27.71; 66.87 18.48 14.24;
65.90 31.93 21.0; 76.55 44.51 22.0];
bottomX = [15.59, 38.57, 61.10, 58.97, 36.98, 64.45, 91.18, 66.87, 65.90, 76.55];
bottomY = [35.07, 37.17, 67.15, 92.05, 63.24, 44.66, 46.85, 18.48, 31.93, 44.51];
bottomZ = [12.88, 13.33, 17.31, 19.09, 16.51, 20.01, 27.71, 14.24, 21.0, 22.0];

```

```

if plot == 0
    startingPt = 1;
    [bezierCurve1] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]);
    interpPts = [bezierCurve1];

    startingPt=4;
    [bezierCurve2] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]);
    interpPts = cat(1, interpPts, bezierCurve2);

    startingPt=7;
    [bezierCurve3] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]);
    interpPts = cat(1, interpPts, bezierCurve3);

    startingPt=10;
    [bezierCurve4] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)]);
    interpPts = cat(1, interpPts, bezierCurve4);

    % Used to see all interpoilations
    %fprintf('Interpolated Points:\n');
    %disp(interpPts);

    figure

    bX = interpPts(:, 1:1);
    bY = interpPts(:, 2:2);
    bZ = interpPts(:, 3:3);

    tri = delaunay(bX, bY);
    trisurf(tri, bX, bY, bZ, 'EdgeColor', 'none');
    colormap(gray);
    hold on

    plot3(bezierCurve1(:, 1), bezierCurve1(:, 2), bezierCurve1(:, 3), 'b',
'LineWidth', 2),
    hold on

```

```

                                CSCI3656_ProblemSet9_2.m
    plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
    plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
    plot3(bezierCurve2(:, 1), bezierCurve2(:, 2), bezierCurve2(:, 3), 'b',
'LineWidth', 2),
    hold on

    plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
    plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
    plot3(bezierCurve3(:, 1), bezierCurve3(:, 2), bezierCurve3(:, 3), 'b',
'LineWidth', 2),
    hold on

    plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
    plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
    plot3(bezierCurve4(:, 1), bezierCurve4(:, 2), bezierCurve4(:, 3), 'b',
'LineWidth', 2),
    hold on

    plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
    plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points

    title('Worthington Alaskan Glacier Top Surface');

elseif plot == 1
    startingPt = 1;
    [bezierCurve5] = bezierInterp([bottomX(startingPt), bottomY(startingPt),
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]);
    interpPts2 = [bezierCurve5];

    startingPt=4;
    [bezierCurve6] = bezierInterp([bottomX(startingPt), bottomY(startingPt),
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]);
    interpPts2 = cat(1, interpPts2, bezierCurve6);

    startingPt=7;
    [bezierCurve7] = bezierInterp([bottomX(startingPt), bottomY(startingPt),
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]);
    interpPts2 = cat(1, interpPts2, bezierCurve7);

```

```

                                CSCI3656_ProblemSet9_2.m
% Used to see all interpolations
%fprintf('Interpolated Points:\n');
%disp(interpPts);

figure

bX2 = interpPts2(: , 1:1);
bY2 = interpPts2(: , 2:2);
bZ2 = interpPts2(: , 3:3);

tri = delaunay(bX2, bY2);
trisurf(tri, bX2, bY2, bZ2, 'EdgeColor', 'none');
colormap(gray);
hold on

plot3(bezierCurve5(:, 1), bezierCurve5(:, 2), bezierCurve5(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)           % plot control
polygon
plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)           % plot control
points
plot3(bezierCurve6(:, 1), bezierCurve6(:, 2), bezierCurve6(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)           % plot control
polygon
plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)           % plot control
points
plot3(bezierCurve7(:, 1), bezierCurve7(:, 2), bezierCurve7(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)           % plot control
polygon
plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)           % plot control
points

title('Worthington Alaskan Glacier Bottom Surface');
elseif plot == 3
    startingPt = 1;
    [bezierCurve1] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]);
    [bezierCurve5] = bezierInterp([bottomX(startingPt), bottomY(startingPt),

```

```
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]]);
```

```
startingPt=4;
```

```
[bezierCurve2] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]]);
```

```
[bezierCurve6] = bezierInterp([bottomX(startingPt), bottomY(startingPt),
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]]);
```

```
startingPt=7;
```

```
[bezierCurve3] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 3), topY(startingPt + 3), topZ(startingPt + 3)]]);
```

```
[bezierCurve7] = bezierInterp([bottomX(startingPt), bottomY(startingPt),
bottomZ(startingPt)], [bottomX(startingPt + 1), bottomY(startingPt + 1),
bottomZ(startingPt + 1)], [bottomX(startingPt + 2), bottomY(startingPt + 2),
bottomZ(startingPt + 2)], [bottomX(startingPt + 3), bottomY(startingPt + 3),
bottomZ(startingPt + 3)]]);
```

```
startingPt=10;
```

```
[bezierCurve4] = bezierInterp([topX(startingPt), topY(startingPt),
topZ(startingPt)], [topX(startingPt + 1), topY(startingPt + 1), topZ(startingPt +
1)], [topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)],
[topX(startingPt + 2), topY(startingPt + 2), topZ(startingPt + 2)]]);
```

```
interpPts3 = [bezierCurve1];
```

```
interpPts3 = cat(1, interpPts3, bezierCurve2);
```

```
interpPts3 = cat(1, interpPts3, bezierCurve3);
```

```
interpPts3 = cat(1, interpPts3, bezierCurve4);
```

```
interpPts3 = cat(1, interpPts3, bezierCurve5);
```

```
interpPts3 = cat(1, interpPts3, bezierCurve6);
```

```
interpPts3 = cat(1, interpPts3, bezierCurve7);
```

```
figure
```

```
bX3 = interpPts3(:, 1:1);
```

```
bY3 = interpPts3(:, 2:2);
```

```
bZ3 = interpPts3(:, 3:3);
```

```
tri = delaunay(bX3, bY3);
```

```

trisurf(tri, bX3, bY3, bZ3, 'EdgeColor', 'none');
colormap(gray);
hold on

plot3(bezierCurve1(:, 1), bezierCurve1(:, 2), bezierCurve1(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
plot3(bezierCurve2(:, 1), bezierCurve2(:, 2), bezierCurve2(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
plot3(bezierCurve3(:, 1), bezierCurve3(:, 2), bezierCurve3(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
plot3(bezierCurve4(:, 1), bezierCurve4(:, 2), bezierCurve4(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(topX, topY, topZ, 'g:', 'LineWidth', 2)          % plot control polygon
plot3(topX, topY, topZ, 'ro', 'LineWidth', 2)          % plot control points
hold on

plot3(bezierCurve5(:, 1), bezierCurve5(:, 2), bezierCurve5(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)          % plot control
polygon
plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)          % plot control
points
plot3(bezierCurve6(:, 1), bezierCurve6(:, 2), bezierCurve6(:, 3), 'b',
'LineWidth', 2),
hold on

plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)          % plot control
polygon
plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)          % plot control
points
plot3(bezierCurve7(:, 1), bezierCurve7(:, 2), bezierCurve7(:, 3), 'b',
'LineWidth', 2),

```

```

hold on

    plot3(bottomX, bottomY, bottomZ, 'g:', 'LineWidth', 2)           % plot control
    polygon                                                         % plot control
    plot3(bottomX, bottomY, bottomZ, 'ro', 'LineWidth', 2)         % plot control
    points

        title('Worthington Alaskan Glacier Surface');
else
    fprintf('ERROR Incorrect Parameter!\n Pass 0 for Top\n Pass 1 for Bottom\n
    Pass 3 for Superimposed\n');
end

if plot == 0 || plot == 1 || plot == 3
    legend(label0, label1, label2, label3);
    ylabel('y');
    xlabel('x');
    zlabel('z');
    view(3);
    box;
end

% Bezier interpolation for given four control points.
% Each control point can be in N-Dimensional vector space.
function [Q]=bezierInterp(P0,P1,P2,P3,varargin)
% Input:
% P0,P1,P2,P3 : four control points of bezier curve
%               control points can have any number of coordinates
% t : vector values of t paramter at which bezier curve is evaluated
%     (optional argument) default 101 values between 0 and 1.

% Output:
% Q evaluated values of bezier curves. Number of columns of Q are equal to
% number of coordinates in control point. For example for 2-D, Q has two
% columns. Column 1 for x value and column 2 for y values. Similarly for
% 3-D, Q will have three columns

if (nargin<4)
    disp('Atleast four input arguments (four control points) are required');
    return
end

[r0 c0]=size(P0); [r1 c1]=size(P1); [r2 c2]=size(P2); [r3 c3]=size(P3);
if (r0~=r1 || r0~=r2 || r0~=r3 || c0~=c1 || c0~=c2 || c0~=c3)
    disp('arg1,arg2,arg3,arg4 must be of equal size');
    return
end

```

CSCI3656_ProblemSet9_2.m

```

%%% Default Values %%%
t=linspace(0,1,101); % uniform parameterization
defaultValues = {t};
%%% Default Values %%%
%%% Assign Valus %%%
nonemptyIdx = ~cellfun('isempty',varargin);
defaultValues(nonemptyIdx) = varargin(nonemptyIdx);
[t] = deal(defaultValues{:});
%%% Assign Valus %%%

[r c]=size(t);
if(r>1 && c>1)
    disp('arg5 must be a vector');
    return
end

% Equation of Bezier Curve, utilizes Horner's rule for efficient computation.
%  $Q(t)=(-P_0 + 3*(P_1-P_2) + P_3)*t^3 + 3*(P_0-2*P_1+P_2)*t^2 + 3*(P_1-P_0)*t + P_0$ 

c3 = -P0 + 3*(P1-P2) + P3;
c2 = 3*(P0 - (2*P1)+P2);
c1 = 3*(P1 - P0);
c0 = P0;
for k=1:length(t)
    Q(k,:)=((c3*t(k)+c2)*t(k)+c1)*t(k) + c0;
end

```