MATH 396 | **Math for 3D Graphics** | **Fall 2008**

Lab Assignment # 4 – NURBS

Due: Tue. Nov. 18, 2008

Directions: In this assignment you'll write code to generate Non-uniform Rational B-Splines. Two functions that calculate Non-uniform B-Splines are provided. You'll need to test these functions and to modify them to plot the curves and to compute and plot NURBS.

You can work with others and discuss the problems, but each student must write his/her own, independent solution. If you are unsure about what i mean by this, please ask!

What to turn in? Each assigned problem specifies sample outputs you should produce and submit. The solution to each problem should include a print out of the function / script file followed by the specified output (*e.g.*, a plot, a vector, a matrix, etc.)

You don't have to use MATLAB to write your code and / or plot your results, if you feel more comfortable using a different programming language and / or graphing utility, you can use it, but you'll need to figure out how to integrate your computed results with your graphical utility on your own.

Problem 1. Start by reading section 15.6 of the text; pay special attention to the derivation of equations 15.79 and 15.80. Once you've read the material, you'll need to download the MATLAB functions `nubs.m`, and `blends.m` from the course website, they are available at:

http://www.csun.edu/~jb715473/math396/nubs.m

http://www.csun.edu/~jb715473/math396/blends.m

Add plotting commands to these functions (add them to the given code, don't execute them from the command line) and test them with the following three cases to reproduce the accompanying plots:

(a) `CP = [-2 0; -1.5 1.5; 1.5 1.5; 2 0];`
`knots = [0 0.001 0.002 0.003 1 1.001 1.002 1.003]; p = 40`

(b) `CP = [2 1; 1 1.5; 0 1.2; -1.2 1.2; -2 0;`
`-1.2 -1.2; 0 -1.2; 1 -1.5; 2 -1];`
`knots=[0 0.001 0.002 0.003 1 2 3 4 5 6 6.001 6.002 6.003]; p = 60`

(c) `CP = r(1:8:100,:)`, where `r` is a 100×3 matrix holding the x, y, and z cooorditanes of the helix $r(t) = (\cos t, \sin t, t)$, for $t \in [0, \pi]$,
`knots = [0 0.001 0.002 0.003 1:10 10.001 10.002 10.003]; p = 120`

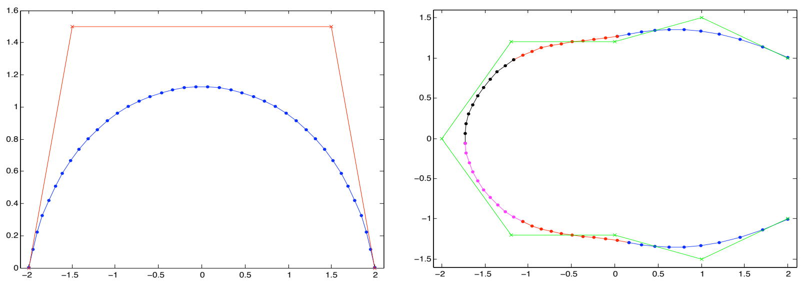

Figure 1: 2D Non-uniform B-Spline, left: spline with four control points (one curve), right: spline with nine control points (six curves).

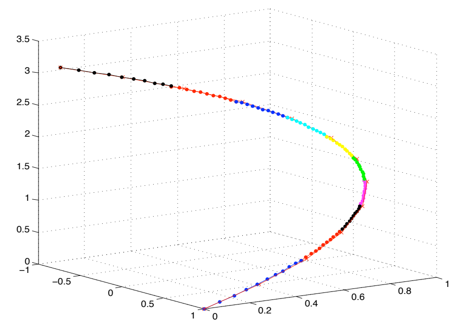

Figure 2: 3D Non-uniform B-Spline.

Some remarks and hints:

1. The values of the knot vector for a non-uniform B-Spline do not need to be distinct, they only need to be given in non-decreasing order. When values repeat in the knot vector, division by zero is redefined as zero. Alternatively, one can prevent division by zero by assigning values that are relatively close to each other.

2. You don't need to generate graphs with so many different colors, but you can try.

3. The curves $Q_i(u)$ that form the spline satisfy $Q_i(t_{i+4}) = Q_{i+1}(t_{i+4})$, at the knot t_{i+4}. In the above examples, each curve $Q_i(u)$, $i = 1, \ldots, n-2$ is plotted over the interval $u \in [t_{i+3}, t_{i+4}]$ using 11 points per curve, with the initial point of the curve Q_{i+1} and the final point of Q_i overlapping. Below are some of the commands that I used to create each of the figures, you should incorporate similar commands into the given code so that the figures are generated within the function. For part (a):

```
>> [Q,N] = nubs(CP,knots,40);
>> x=zeros(40,1);
```

```
>> y=zeros(40,1);
>> x(:) = Q(1,1,1:40);
>> y(:) = Q(1,2,1:40);
>> plot(x,y,'.-')
>> hold on
>> plot(CP(:,1), CP(:,2), 'x-r')
>> axis([-2.1 2.1 0 1.6])
```

For part (b):

```
>> x = zeros(11,1);
>> y = zeros(11,1);
>> x(:) = Q(1,1,1:11);
>> y(:) = Q(1,2,1:11);
>> plot(x,y,'.-')
>> hold on
>> x(:) = Q(2,1,11:21);
>> y(:) = Q(2,2,11:21);
>> plot(x,y,'.-r')
...
>> x=zeros(10,1);
>> y=zeros(10,1);
>> x(:) = Q(6,1,51:60);
>> y(:) = Q(6,2,51:60);
>> plot(x,y,'.-')
>> plot(CP(:,1), CP(:,2), 'x-g')
>> axis([-2.1 2.1 -1.6 1.6])
```

For part (c):

```
>> x(:) = Q(1,1,1:11);
>> y(:) = Q(1,2,1:11);
>> z(:) = Q(1,3,1:11);
>> plot3(x,y,z,'.-')
>> x(:) = Q(2,1,11:21);
>> y(:) = Q(2,2,11:21);
>> z(:) = Q(2,3,11:21);
>> hold on
>> plot3(x,y,z,'.-r')
...
>> x(:) = Q(9,1,81:91);
>> y(:) = Q(9,2,81:91);
```

```
>> z(:) = Q(9,3,81:91);
>> plot3(x,y,z,'.-r')
>> x = zeros(10,1);
>> y = zeros(10,1);
>> z = zeros(10,1);
>> x(:) = Q(10,1,91:100);
>> y(:) = Q(10,2,91:100);
>> z(:) = Q(10,3,91:100);
>> plot3(x,y,z,'.-k')
>> grid on
```

Problem 2. Write a MATLAB function called `nurbs.m` that takes as input a vector `CP` with the coordinates of the control points, a knot vector, `knots`, a weight vector, `weights`, and the number of points, `p`, in the parameter interval $u \in [knots(4), knots(n-2)]$, and plots the corresponding NURBS. Sample output: a 2D curve and a 3D curve. You can use the control points specified in problem 1, but change the values of the knot vector so that it is truly non-uniform.