Using One-Dimensional Splines to Fit General Curves in the Plane
Solutions

The process is this:

1. Define $s_1 = 0$ and $s_i = s_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$ for i = 2, …, n.
2. Fit the cubic spline f satisfying $f(s_i) = x_i$ for i = 1, …, n. (i.e. preprocess)
3. Fit the cubic spline g satisfying $g(s_i) = y_i$ for i = 1, …, n. (i.e. preprocess)
4. For any value $s \in [0, s_n]$, the point $(f(s), g(s))$ is a point on the curve. (i.e evaluate)

Assignment:

Write a `Matlab` function `Spline2D (x, y, m)` whose inputs are column arrays `x` and `y` (of the same length) and `m`, a positive integer. Output is a pair of arrays `[xout, yout]` of length `m` that are the mappings of m equipaced points from 0 to s_n onto the curve.

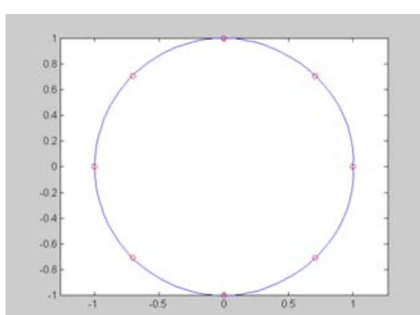
```
function [xout, yout] = Spline2D (x, y, m)
n = length (x);
s = zeros (n,1);
for i = 2:n
    s(i) = s(i-1) + sqrt((x(i)-x(i-1))^2+(y(i)-y(i-1))^2);
end
sout = linspace(0, s(n), m)';
xout = spline(s, x, sout);
yout = spline(s, y, sout);
```

Test your function with these data sets:

1. Nine points on a circle: `t = linspace (0, 2*pi, 9)'; x = cos(t); y = sin(t);`
   Use m = 101.

```
t = linspace (0, 2*pi, 9)'; x = cos(t); y = sin(t);
[xout, yout] = Spline2D (x, y, 101);
plot (x, y, 'ro', xout, yout, 'b');
axis equal;
```

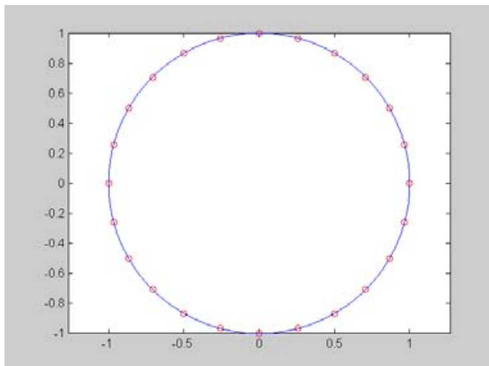
results in



2. 25 points on a circle: `t = linspace (0, 2*pi, 25)'; x = cos(t); y = sin(t);`
   Use m = 501.

```
t = linspace (0, 2*pi, 25)'; x = cos(t); y = sin(t);
[xout, yout] = Spline2D (x, y, 501);
plot (x, y, 'ro', xout, yout, 'b');
axis equal;
```

results in



3. 31 points on a squiggle: `t = linspace (0, 2.5, 31)'; x = sin(t)+1/2*cos(t).*sin(5*t);`
   `y = sin(t);` Use m = 501.

```
t = linspace (0, 2.5, 31)'; x = sin(t)+1/2*cos(t).*sin(5*t);
y = sin(t);
[xout, yout] = Spline2D (x, y, 501);
plot (x, y, 'ro', xout, yout, 'b');
axis equal;
```

results in