# Math 411 Hwk 4

**Problem 1.** *Write a Matlab function* `myspline`, *which takes as input the knots* $\{(t_j, z_j)\}_{j=0}^n$, *which are also interpolation points, and the sample points* $\{x_k\}_{k=1}^m$, *and uses the cubic spline interpolation method to return the values* $\{y_k\}_{k=1}^m$ *so that* $y_k = s(x_k)$ *for each* $k = 1, \ldots, m$, *where* $s(x)$ *is the cubic spline subject to the given boundary conditions. Your function should be able to handle "natural" (default), "clamped", and "not-a-knot" options. If derivative values are not given as input in the "clamped" case, have them default to zero. Be sure also to include well-written and complete documentation in your code so that a user will be able to know how to execute your code by typing* `help myspline`. *Below is an example:*

```
>> help myspline

  Returns the values of a cubic spline with knots (t,z) at
  the sample points x.  Boundary condition options are

      'natural'     - natural splines
      'clamped'     - clamped splines with end point
                       derivatives at f(1) and f(2)
      'not-a-knot'  - not a knot splines

  Usage:

  myspline(t,z,x);                    % defaults to natural
  myspline(t,z,x,'natural');          % same as above
  myspline(t,z,x,'clamped');          % zero default derivative
  myspline(t,z,x,'clamped',[4 5]);
  myspline(t,z,x,'not-a-knot');
```