ECOR 2606 Lab 10

1/. File SCO2.txt contains tabulated NIST (National Institute of Standards and Technology) data for supercritical carbon dioxide. Column one is the temperature (in K), column two is the pressure (in bar), column three is enthalpy, column four is entropy, column five is the constant volume specific heat (c_v), and column six is the constant pressure specific heat (c_p).

The data can be imported into the Matlab environment by using load command:

```
load SCO2.txt
```

Provided that Matlab is able to find the file (it must be in a folder on your path) this will produce a variable called SCO2 and having 161 rows and 6 columns. We are interested primarily in columns one (temperature in K) and six (c_p). For convenience, create two vectors containing the values in these columns.

In theory, the enthalpy change between any two temperatures can be calculated by integrating c_p.

$$\Delta h = \int_{T1}^{T2} c_p dT$$

Use numerical integration to integrate c_p between T = 300 K (first row of SCO2) and T = 1100 K (last row of SCO2). Use both trapezoidal integration and Simpson's 1/3 rule. The trapezoidal integration can be performed using Matlab function "trapz" but you will have to perform the Simpson's rule integration using your own code. Write a little function that, given *h* and an odd number of *y* values (assumed to correspond to evenly spaced *x* values), computes and return the value of the integral.

Note: The ability to use *start:step:limit* to extract selected elements of a vector and function *sum* come in handy here. The following sample code should give the basic idea.

```
v = [ 59 3 7 2 12 67];
% sum up every 2nd element starting with the 1st element
s1 = sum(v(1:2:length(v)))    % s1 = 78
% sum up every 2nd element starting with the 2nd element
s2 = sum(v(2:2:length(v)))    % s2 = 72
```

Compare your results to the actual enthlapy change between the two temperatures (recall that column 3 of SCO2 contains enthalpy values). The less than perfect match is largely a result of rapidly changing properties near the critical point. Try repeating the experiment for T = 320 K (5th row of SCO2) to T = 1100 K (last row). This should give better results.

Create a single m-file (part1.m) that does all of the above and submit it as proof of your attendance in the lab. Your Simpson's rule function should be included as a subfunction.

2/. For convenience, create vectors containing the first ten temperature values and the first ten c_p values. Your data will range from 300K to 345K. Plot the data points to get an idea of what things looks like.

Now further suppose that we need to be able to estimate c_p for temperatures anywhere in this range. This will involve interpolation. Three possible techniques are

1/. A linear spline (aka piecewise linear interpolation)
2/. Polynomial interpolation using all data points and a ninth order polynomial
3/. A cubic spline.

Create a plot showing the both the data points and curves representing all three of these techniques. Use all three techniques to estimate c_p at temperatures of 302.5K and 343K. Is a ninth order polynomial a good idea?

File cpvsT.txt contains NIST values of c_p (column 2) for T (column 1) from 300K to 345K in steps of 1K. See how good your interpolating curves actually are by creating a plot showing your curves and the actual data.

3/. Assume $F = \int_0^H 200\left(\frac{z}{5+z}\right)e^{-2z/H}dz$. What value of H makes F equal to 700?

Note that F is undefined at H = 0 and make sure that any function passed to Matlab function *quad* (or *quadl*) is vector friendly.   Once you have a function that is given H and returns F the rest should be really easy.

Time left over? Try solving part 3 by writing a function that uses generated data points and Simpson's rule to calculate the value of the integral.