Applied Engineering Problem Solving (CHEN.3170)

Homework Assignment #3b -- Fall 2017

More Programming in Matlab

Problem #1: Properties of Silicon Carbide

This problem focuses on the reading and plotting of data from an existing data file. The data file **SiC.dat** gives the temperature dependence for several properties of silicon carbide (SiC). To properly complete this project, you should perform the following tasks:

a. Download the files **SiC.dat** and **loadColData.m** from the course website (see the HW Assignment page). **Note:** These files were obtained from the Companion Website associated with a numerical methods textbook by Prof. Gerald Recktenwald (this text was used for this course in previous years).

Once you have the appropriate files, write a Matlab program to do the following:

b. Read the **SiC.dat** data file using the **loadColData.m** function (see documentation within the function file).

c. Plot the density, specific heat, and thermal conductivity versus temperature over the range given in the data file. We would like to put all these data on the same plot but, since these three properties all have different units, we will need to use the subplot command to accomplish this task. However, since the thermal conductivity has the largest variation with temperature (as seen by visual inspection of the data), it would be nice to include this in a larger plot window. To accomplish this, when plotting thermal conductivity, you should use plot 1 of a 1x2 array of plots, and when plotting the other two variables, use plots 2 and 4 of a 2x2 array of plots. In doing this we can put three subplots on the same page, with one larger plot window on the left and two smaller windows on the right. This is the format desired here. Also be sure to properly label all the plots!!!

Problem #2: Geometric Series

A geometric series is a series with a constant ratio between successive terms. For example, the series

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\ldots$$

is a geometric series because each term except the first can be obtained by multiplying the previous term by 1/2.

We can generalize this relationship with the following infinite series representation

$$S=\sum_{k=1}^{\infty} a\,r^{k-1} = a + ar + ar^2 + ar^3 + \ldots$$

where a is the first term in the series and r is the common ratio between consecutive terms. For the geometric series to be convergent, the absolute value of r must be less than unity (i.e. $|r| < 1$).

Since the geometric series occurs so frequently in applications, this series has been studied extensively and a lot is known about the geometric series. In particular, it can be shown that for $|r| < 1$, the series converges to $S = a/(1-r)$. Thus, we see that, for the above example with $a = 1/2$ and $r = 1/2$, the series should converge to a value of $S = 1$.

Now, assume for the moment that we do not know all this information about the geometric series, and we are asked to simply numerically evaluate the series when $a = 1/2$ and $r = 1/2$. Of course we cannot use an infinite number of terms, so we must truncate at some finite number of terms, N, where the finite series can now be written as

$$S_N = \sum_{k=1}^{N} a r^{k-1}$$

where S_N is the partial sum after N terms -- and as $N \to \infty$, $S_N \to S = a/(1-r)$.

In this problem, with $a = 1/2$ and $r = 1/2$, you are asked to work with this series from two different perspectives. As part of your solution, you should write a single Matlab script file that has two distinct parts, as follows:

a. In the first part of the Matlab program request the value of N from the user, compute the value of S_N using the above series for this particular N, compare this result to the known limiting value of unity, and report all this information back to the user (simply print it to the screen).

b. In the second part of the program, request the iteration tolerance in percent (i.e. the maximum percent change that is allowed so that the difference between term k and k+1 is considered negligible) in the computation of the geometric series, and report back to the user the number of terms needed to achieve this goal along with the actual error based on the known solution.

Run your program a couple of times with different input data (i.e. different values of N and different values for the maximum iteration tolerance allowed) and print out the program results. Turn in these data along with a copy of the Matlab program as documentation for this problem and clearly explain how this exercise illustrates the use of various looping structures in Matlab.

Documentation

Documentation for this assignment should include a listing of any user-generated Matlab files (i.e. you do not need to include the **loadColData.m** file used in Prob. #1), the resultant Matlab plots and/or tabular data, and a brief description of the data and results of your analyses for each of the problems. Keep all the parts for a given problem together. An overall professional job is expected!

See HW#1 for a description of the expected format -- every HW in this course should follow these basic instructions…