Matlab Homework 8c

The same requirements as for homework 3c apply.

1. In an earlier homework, (homework 3?), you printed out the roots ω_n of function `drumFreqEq`, "drumFreqEq.m" being

   ```
   function error = drumFreqEq(omega,k)
   error = besselj(0,omega) - k*omega.*besselj(1,omega);
   end
   ```

   in the format

   ```
   Frequencies for k = 1.1
   Frequency 1: approximate: 12.1234567, exact: 12.1234567
   Frequency 2: approximate: 12.1234567, exact: 12.1234567
   ...
   ```

   Repeat this, but now use a `for` loop over counter n, going from 1 to value $n_{\max} = 9$, to find and print the first 9 roots, using the same code for all 9 cases. Take $k = 0.5$ again.

   Note: If you did not do the earlier homework correctly, the correct solution is under the Blackboard "Course Library", "Matlab", "Some notes" link.

2. In an earlier homework, you created the matrix

$$A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

   by typing in all those numbers. This is typo-prone and you have to do it all again with a vengeance when you want to try much more points.

   So instead define a variable m that contains the desired number of rows of the matrix, i.e. 6. Then initialize A as a square zero matrix of size $m \times m$. This will already put in all the zeros. Next put in the -2 and 1 in row 1 and the 1 and -2 in final row m, in the correct columns. Next use a `for` loop over rows $i = 2$ to $m-1$ of the matrix to put in the nonzero elements in those rows. Note that the -2 always goes on the "main diagonal" where column number j equals i and 1 values go into the previous and next column. Finally, change m into 10 and check that you get the corresponding matrix of that size correctly. Except changing m, you should not have to change anything else in your code.

3. Some mathematician claims that the sum

$$\sum_{i=1}^{I} \frac{1}{i}$$

in other words

$$\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\ldots+\frac{1}{I}$$

becomes infinite when I becomes infinite. Let's check that out.

First write Matlab code that performs the sum up to I, where at first you take I to be 10.

Next put that summation process inside an "outer" "for" loop on I where you take I to be the successive values [10, 100, 1000, 10000, 100000].

After studying the values you get for the sum using these five I values, use `disp` to comment on whether it looks like the sum converges to a definite value when I becomes bigger and bigger, or whether it looks like the value seems to keep getting bigger and bigger.

Note: Summing 100,000 terms may be a bit slow on some computers. You may want to wait with that one until everything works OK.

4. The following function, the "sine integral",

$$\mathrm{Si}(x)=\int_0^x \frac{\sin\xi}{\xi}\,\mathrm{d}\xi$$

cannot be expressed in terms of simpler functions. You cannot find the needed antiderivative in terms of normal functions, even though the integrand *seems* so simple.

However, the Taylor series of the Si function is easy to find. (Just write the Taylor series for $\sin\xi$, divide by ξ, and integrate.) The result is

$$\mathrm{Si}(x)=\frac{x}{1!1}-\frac{x^3}{3!3}+\frac{x^5}{5!5}-\frac{x^7}{7!7}+\ldots$$

You can consider this to be a sum of terms where term number i is zero if i is an even number, while term i is $-x^2(i-2)/((i-1)i^2)$ times the previous nonzero term if i is odd. (Check that this is true for the terms shown above). Write code to sum the series up to some given value I of i and so compute $\mathrm{Si}(5\pi)$. The correct value is about 1.6339648 according to my table book. Experiment with the number of terms you have to sum to get the value correct to the given number of digits. Print out as

```
Value: 1.12345678
Table: 1.6339648
I needed an I value of ....
```

Note: You can skip the even values of i in the for loop using a START:STEP:END construct in the "for" command. You will want to do term $i=1$ separately and start the "for" loop at $i=3$.

Note: Si is actually a quite important function, and Matlab provides this function as "sinint". (It is apparently within the symbolic logic package; at least the Octave version is.) I would not be surprised if they used some adulterated Taylor series to evaluate the function for relatively small values of x.