Matlab Homework 9c

The same requirements as for homework 3c apply.

1. In the previous homework, 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
```

   using a `for` loop over the root counter n, going from 1 to a value $n_{\max}$. Repeat this, but now make Matlab stop printing out roots when it sees that the error in the guessed value has become less than 0.03. Take $k = 0.5$ again. Make sure the output is lined up in columns and neat; this time output the frequencies with 2 digits behind the decimal point

```
Frequencies for k = 1.1
Frequency  1: approximate: 12.12, exact: 12.12
Frequency  2: approximate: 12.12, exact: 12.12
...
```

   Octave users: The Octave `fzero` does not always find the closest root for some reason. This can be fixed up by providing it a complete search interval `[guess guess+1]` instead of just the initial guess `guess`.

2. In the previous homework, you created matrices like

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

   by initializing a zero matrix, putting in the nonzero elements in rows 2 to $m-1$, (with m the size of the matrix) in a `for` loop, and manually adding the nonzero elements in rows 1 and m. This time do the same, but put in *all* nonzero elements in a `for` loop over all m rows. Use `if` statements to ensure that you do not write nonzero values outside the boundaries of the matrix in rows 1 and m, as you would normally do. Check for $m = 6$ and 10

3. Some mathematician claims that the sum

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

   in other words

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

is infinite. Let's check that out. Write a `for` loop that sums until Matlab finds that the estimated error has become less than 0.05, or up to say 100,000 terms, as many terms as Matlab will sum in a reasonable time. Take the estimated error to be i times the first neglected term. If the estimated error does become less than 0.05, print a message "The mathematician seems to be wrong. The sum seems to be equal to VALUE to an estimated error of 0.05." For VALUE print out the actual obtained value of the sum with 2 digits behind the decimal point. (If the estimated error is 0.05, it is useless and confusing to print out more that two decimals.)

The same mathematician claims that the sum

$$\sum_{i=1}^{I} \frac{1}{i^2}$$

is finite for infinite I, and in fact equal to $\pi^2/6$. To check this, copy over the code for the series above and in that, change the $1/i$ term into $1/i^2$, and "wrong" into "right". *If* the mathematician is right, also check whether the found sum is equal to $\pi^2/6$ within an error of about 0.05

4. In the previous homework you summed the Taylor series of the sine integral,

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

at $x = 5\pi$ to get the correct value 1.6339648 to that many digits. Repeat this, but this time let Matlab itself decide when to finish summing based on the allowed tolerance of 0.00000005 that you provide this. Warning: this is an alternating series; use the appropriate error estimate for that. Print out how many terms Matlab ended up doing and its final value for $\mathrm{Si}(5\pi)$.

Next copy the code and then make it keep summing until the accuracy no longer improves. Compare the result you get then with the accurate value 1.633964846102835 (obtained from Matlab's `sinint` function). If the Taylor series result is not as accurate as you would expect, comment on what you think is the reason. Remember that normal Matlab numbers have a relative error of about 10^{-16}. In other words, there are about 16 good digits starting from the first nonzero digit.