EML 3002L **M.E. Tools Lab** **11/08/18**
Matlab Exam 1 *Van Dommelen* *12:30-1:45 pm*

NO CELL PHONES, HEADPHONES/BUDS, NO CALCULATORS. SAVE FREQUENTLY. SAVE BEFORE PUBLISHING!!!

ONLY MATLAB MAY BE ACTIVE ON YOUR COMPUTER. Acrobat may only be open at the end, when you are ready except for publishing and actively looking at exam2q1.pdf, exam2q2.pdf, or exam2q3.pdf with it.

VARIABLE NAMES MUST BE APPROPRIATE AND CLEARLY SHOW WHAT VARIABLE IT IS OR A 50% CREDIT REDUCTION IS APPLIED.

After translation into mathematics, *only* Matlab may be used to solve the full problem *as posed*. Use the appropriate procedures as covered in the lectures.

Open notes: print-outs of lectures and homeworks allowed. You can also use your own homework .m files.

1. Create a script `solvesys.m` (lowercase) that, assuming that a matrix size n is set before the script is called, solves the following system of equations

$$\begin{aligned} x_1 + 2x_2 &= 0 \\ x_1 + 4x_3 &= 2 \\ x_2 + 4x_4 &= 3 \\ \vdots &= \vdots \\ x_{n-2} + 4x_n &= n-1 \\ x_n = 0 & \end{aligned}$$

For example, if $n = 6$, the system in matrix/vector form $A\vec{x} = \vec{b}$ should look like, written out,

$$\begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \end{pmatrix}$$

Script `solvesys.m` should initialize the system matrix A and right hand side vector $\vec{b}$ to zeros, then put in the *individual* nonzero coefficients using an appropriate `for` loop over the interior equations. (Do not use special matrices here. And the nonzero coefficients of the initial and final equations should be done separately.) The final matrix and right hand side vector should be printed out.

Then have Matlab check whether it will be able to find a reasonable solution to the system, and if not, display a message to that effect.

Otherwise let Matlab solve the system using the proper method, and print the solution. However, assuming that the right hand sides of the equations are measured values with a relative error of 0.001, let Matlab also estimate the relative error that that solution will have, and if it is greater than 5%, let Matlab print the following warning:

```
*** May have an error of *%
```

using an appropriate `fprintf` (no decimals in the percentage). Your script should work for any reasonable value of n, but use the provided script `exam2q1.m` to test it out for $n = 3$ and $n = 6$.

2. One extremely important elementary function is the natural log ln; it is used to evaluate powers. Unfortunately, computers used to be very slow to evaluate the ln function. Let's explore one possibility.

Using manipulations that should be relatively easy for a digital computer but that do not concern us here, the problem can be reduced to evaluating

$$\ln\frac{1}{1-x}=\frac{x}{1}+\frac{x^2}{2}+\frac{x^3}{3}+\frac{x^4}{4}+\ldots=\sum_{i=1}^{\infty}t_i \qquad \text{where} \quad t_i=\frac{x^i}{i}$$

where $x \geq 0$ and in the worst case

$$x=1-\frac{1}{\sqrt{2}}$$

Your job is to write a script (being `examq2.m`) that tries to sum the above Taylor series in terms of x to the maximum achievable accuracy, and to no more than that. However, make sure to never sum more than 500 terms. To make the summation more efficient, use the fact that, as long as i is at least 2,

$$t_i=t_{i-1}x\frac{i-1}{i} \qquad (i \geq 2)$$

(This relation could be improved upon, but we will keep it simple.) Before your summation solution, set variable x equal to the worst case value above. After your summation solution, display a warning if the maximum accuracy was not achieved, and also, using `fprintf`, print out:

```
Total number of terms summed: *
The obtained value is 1.1234567890123456
The "exact"  value is 1.1234567890123456
The difference between the two is 1.1E-12.
```

where for the "exact" value use $\ln\sqrt{2}$ as evaluated by Matlab directly. Absolutely no data numbers in the `fprintf` FORMATSTRINGs.

3. Using the Symbolic Math Toolbox, do the following:

   (a) An unstable system has the Laplace transform solution

   $$(s^3+s^2+s+1)/(s^4-2s^3-14s+15)$$

   i. To get an idea of the system properties, factor this ratio exactly. Your answer should take the form of a single ratio of multiplied factors.

   ii. Find the exact partial fraction expansion.

   (b) The equation of the normalized "Lemniscate of Bernoulli" is:

   $$(x^2+y^2)^2=x^2-y^2$$

   Let Matlab find the exact values of y in terms of x.

   (c) Define the following symbolic function:

   $$f=\frac{1}{\sqrt{a^2-x^2}}$$

   Next:

   i. Let Matlab find the symbolic antiderivative F.

   ii. Let Matlab exactly evaluate F when $a=1$ and $x=1/\sqrt{2}$

   iii. Let Matlab convert F into a *numerical* function F_{num} that evaluates the antiderivative numerically.

   iv. Evaluate F_{num} when the values 1 and $1/\sqrt{2}$ are substituted for a and x respectively. Print out this value to 32 decimals behind the point using `fprintf`.

   v. Compare with the earlier exact value, printed out to 32 correct digits.