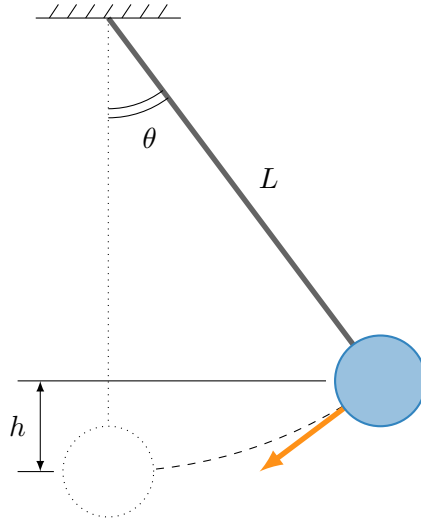




HOMEWORK 4

Due: Tuesday, May 1, 2018, 11:55pm

1. **The Pendulum Problem** In class, we discussed how the second-order differential equation describing the motion of a simple pendulum can be treated as the two separate differential equations shown below.



$$\frac{d\omega}{dt} = -\frac{g}{L} \sin(\theta) \quad (1)$$

$$\frac{d\theta}{dt} = \omega \quad (2)$$

where  $g$  is the acceleration due to gravity,  $9.81 \text{ m/s}^2$ ,  $L$  is the length of the pendulum, and  $\theta$  is the angular position relative to the vertical axis. Recall that for a simple pendulum, we neglect complicating factors like the weight of the connecting rod, friction at the hinge, and air resistance. Your task is to write a script to study the motion of such a pendulum.

- Before you start programming, write down the discretized governing equations for this problem using the explicit (forward) Euler method. Use subscripts  $k$  and  $k + 1$  to denote the known values at step  $k$  and the values to be determined at step  $k + 1$ . Remember, when using explicit Euler, the terms on the right-hand side should always reflect information from time step  $k$ .
- For a pendulum with length  $L = 1 \text{ m}$  starting from rest position  $\theta_0 = \pi/3$  at  $t = 0$ , calculate the angular position,  $\theta$ , angular velocity,  $\omega$ , and angular acceleration,  $\alpha$ , up to time  $t = 20 \text{ s}$  with  $\Delta t = 0.005 \text{ s}$ . Using the velocity and position values, calculate the total energy per unit mass of the pendulum at each timestep. The total energy per mass,  $E_{total}$ , is the sum of potential and kinetic energy as shown below

$$E_{total} = gh + \frac{1}{2}(L\omega)^2$$

where  $h$ , the vertical displacement from the rest position, is a function of  $\theta$  and  $L$ .

- Plot your three kinematics vectors—position, velocity, and acceleration vs. time—on the same axis. In a separate figure window, plot the total energy of the pendulum vs. time.
- Repeat this experiment to generate a second set of plots, but this time, treat Equation 2 *implicitly* as explained in class. That is, the calculation of the angular position at the next step,  $\theta(k + 1)$ , should be based of the newly-calculated velocity at the next step,  $\omega(k + 1)$ , on the right-hand side as opposed to the current velocity,  $\omega(k)$ .

- (e) Using your plots to support your discussion, does forward Euler conserve energy over time? Does using a smaller timestep size,  $\Delta t < 0.005$ , change this behavior? Include the corresponding plots in your report to back up this discussion. Does the semi-implicit method tested in Part (d) conserve energy? Why or why not?

**Note:** Although you're encouraged to experiment with different simulation parameters (e.g., length, initial position, timestep size) to generate interesting results for your report, please turn in the version of your code with the values listed in Part (b) and  $g = 9.81 \text{ m/s}^2$  using explicit Euler only. Don't forget to include any necessary plots in your report from this week forward!

2. **DNA Analysis.** In class, we talked about how DNA is good example of how a huge amount of information can be stored in a vector. In this problem, you'll use iteration and logical expressions to analyze a segment of human DNA stored as a 1-D array. For the sake of programming convenience, the DNA provided to you on CCLE uses numeric values instead of the usual alphabetic abbreviations (A, C, G, and T) as shown in the key below.

Base	Number
(A)denine	1
(C)ytosine	2
(G)uanine	3
(T)hymine	4

Your task is to write a script to calculate the lengths of every protein-coding segment in the DNA as shown in the example below.

C	T	A		A	T	G		A	C	C		T	A	G		A ...	
2	4	1		1	4	3		1	2	2		4	1	3		1 ...	

- (c) After reaching the end of the DNA vector, print the following statistics to the command window in the format shown below (don't panic, these values are for illustration only):

```
Total Protein-Coding Segments: 1300
Average Length: 99.11
Maximum Length: 1200
Minimum Length: 9
```

Where the lengths all reflect the number of bases.

- (d) Using your results from Part (c) , what percentage of this DNA is directly used in the protein-coding process (including the start and stop codons)? What is the *most* frequently and *least* frequently used stop codon (we're only interested in counting the stop codons that terminate a protein-coding segment, not any that may be floating around in between these segments)? In the method outlined above, the sequence **ATG-ATG-TAG** would be identified as a 9-base protein coding length since we ignore any new start codons that may appear while searching for a stop codon. In detailed words or pseudocode in the report only, how would you modify your algorithm to allow the starting point of a protein-coding segment to be updated such that cases like this were identified as a 6-base length?

**Note:** Please include the `chr1_sect.mat` in your .zip file when submitting. You're welcome to use the built in MATLAB function `mean`, `max`, and `min` if you find them helpful. As with all things programming, it's *much* easier to start with a small test case where you know the answer first! Try creating your own DNA vector with the numbers in the above examples and confirm that you recover the correct answer before using using the full sequence stored in `chr1_sect.mat`.

Using the naming convention presented in the syllabus, submit **two** separate files to the CCLE course website: (1) a .pdf of your written report and (2) a .zip file containing all of the MATLAB files written for the assignment. Remember to use good coding practices by keeping your code organized, choosing suitable variable names, and commenting where applicable. Any of your MATLAB .m files should contain a few comment lines at the top to provide the name of the script, a brief description of the function of the script, and your name and UID.