Homework 4 Report

Jesse Harper

Introduction 1.1

This problem was to use the model of the second order differential equation, describes the motion of a pendulum, and find the angular position, velocity, and acceleration as a function of time. The method used to do this was the forward Euler method. In addition to the method described by equation 1.2.3, the second order nature of this problem allows for a second variation based on equation 1.2.4. Both methods were performed and compared. In order to compare these two methods, both plots of the calculated quantities were taken as well as a calculation of the total energy of the system. After comparing these two methods as well as varying time step, it was found that there are significant differences in the results found by using either method or from varying time step size. The following section will go over the models and methods used.

Models and Methods 1.2

The kinematic model of the pendulum comes from Newton’s second law and the assumption that there are no losses due to drag, friction, and that the string that connects the pendulum to the wall does not deform. From a force balance, with the only force acting on the pendulum being gravity, the angular acceleration can be found in the form of equation 1.2.1 below.

$$\frac{dw}{dt} = -\frac{g}{L}sin(\theta) \qquad \textbf{eqn 1.2.1}$$

Since omega is also the first derivative of theta, equation 1.2.2 follows as below.

$$\frac{d\theta}{dt} = \omega \qquad \textbf{enq. 1.2.2}$$

The computational method that was used to solve these two differential equations is forward Euler.

$$f(k + 1) = f(k) + \frac{df}{dt}(k) * \Delta t \qquad \textbf{eqn.1.2.3}$$

In addition to using this method of calculation, a second variation of this method was also used to compare results with.

$$f(k + 1) = f(k) + \frac{df}{dt}(k + 1) * \Delta t \qquad \textbf{eqn. 1.2.4}$$

This second method was used to calculate the approximation of theta based on the first method’s approximation of omega. The following figure illustrates the diagram and clarifies variable.

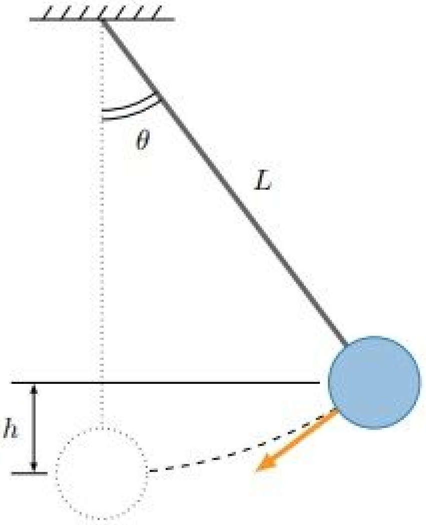

Figure 1.2.1: diagram of pendulum and forces

Another quantity of interest for this problem is the total energy of the system which can be calculated from the following equation.

$$E_{Total} = gh + \frac{1}{2}(L\omega)^{2} \qquad \textbf{eqn. 1.2.5}$$

The following section gives a summary or calculations and results.

Calculations and results 1.3

Here are plots from the results of several different variations on the forward Euler method solution to the pendulum problem. Figure 1.3.1 and 1.3.2 show the results from forward Euler with a time step, dt, equal to 0.005 seconds.

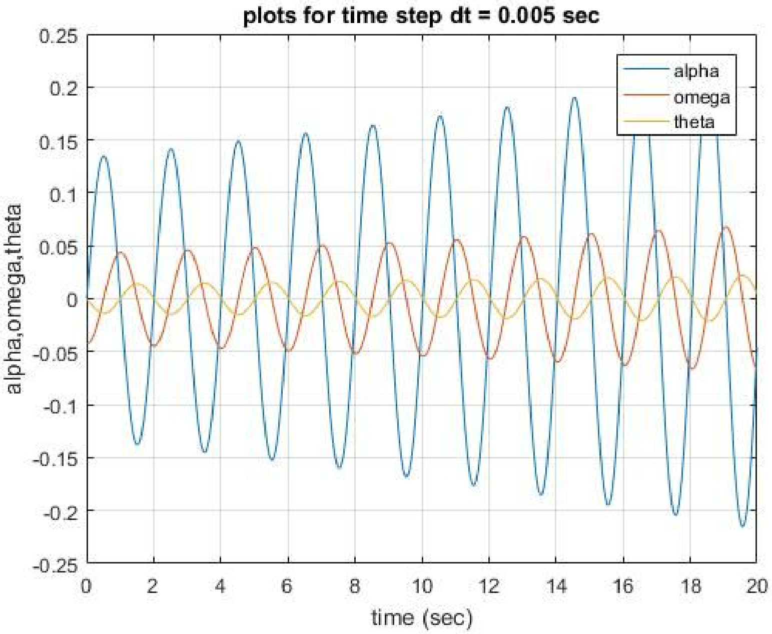

Figure 1.3.1: angular acceleration (alpha), velocity(omega), and position(theta) dt = 0.005 sec

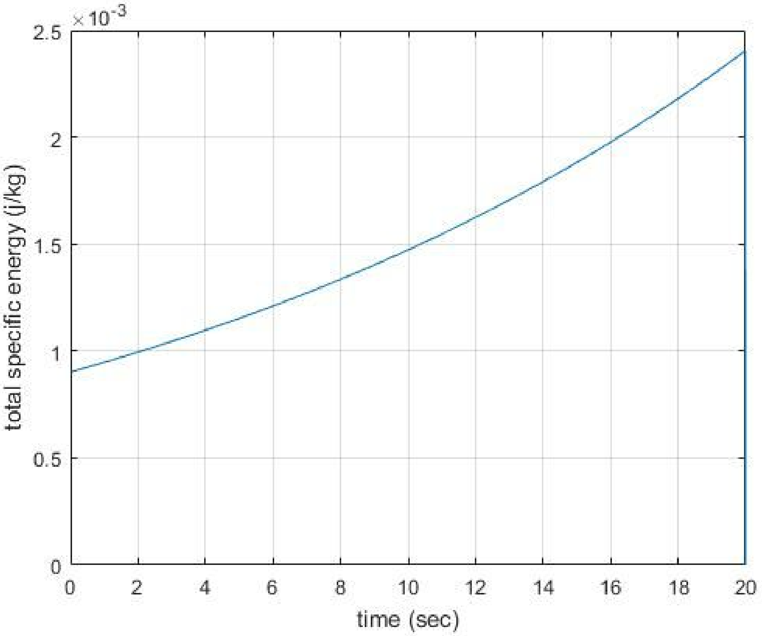

Figure 1.3.2: Total specific energy versus time with dt = 0.005

Figure 1.3.3 and 1.3.4 show the results from forward Euler with a time step, dt, equal to 0.00005 seconds.

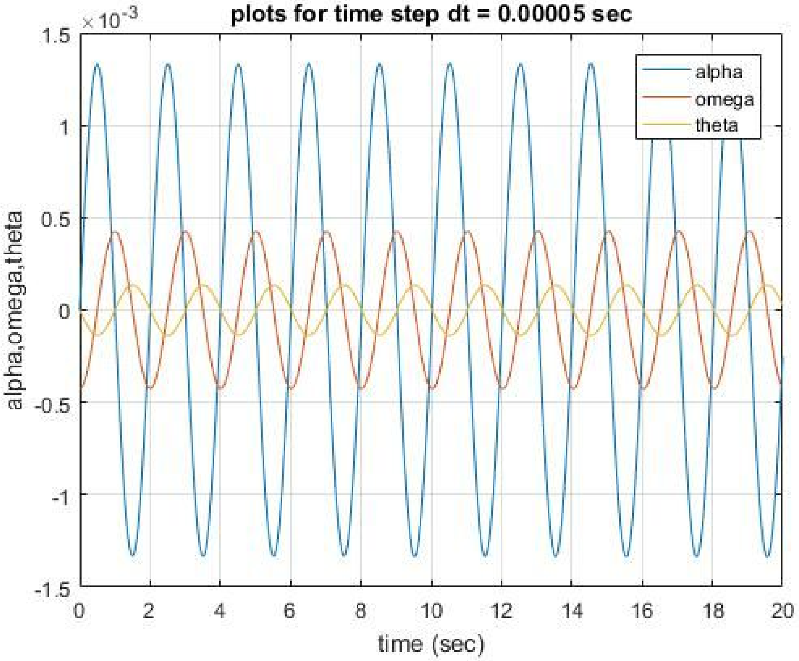

Figure 1.3.3: angular acceleration (alpha), velocity(omega), and position(theta) dt = 0.00005 sec

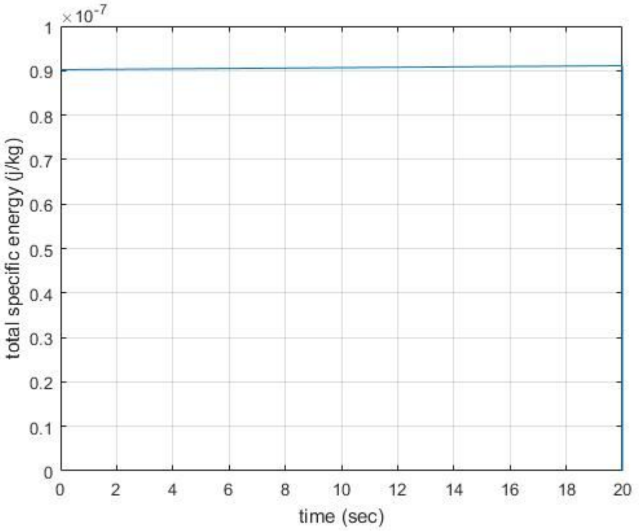

Figure 1.3.4: Total specific energy versus time with dt = 0.00005

Figure 1.3.5 and 1.3.6 both show results when equation 2.2.2 is calculated using the implicit variation on forward Euler.

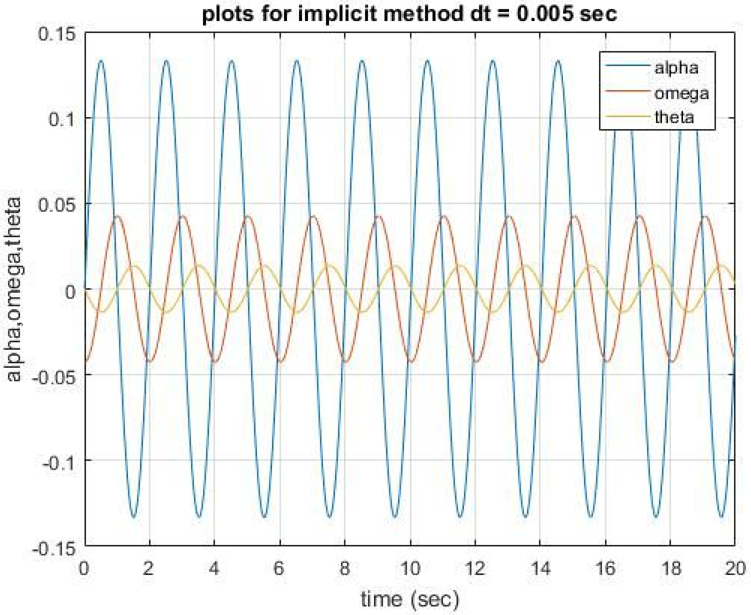

Figure 1.3.5: angular acceleration (alpha), velocity(omega), and position(theta) implicit method

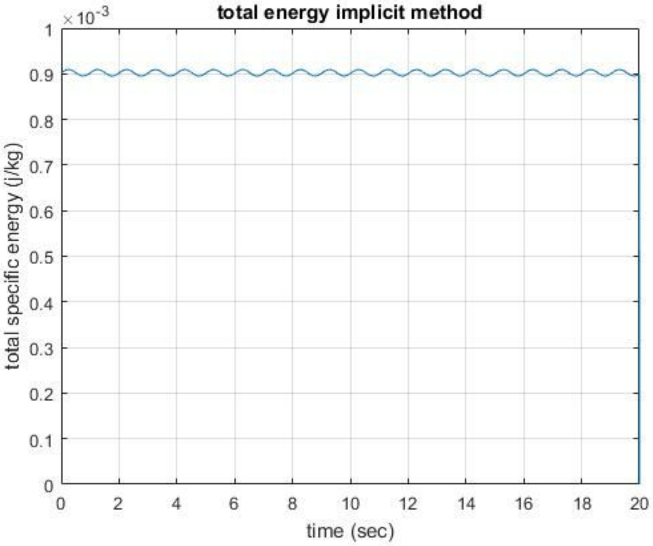

Figure 1.3.6: Total specific energy versus time implicit method

Discussion 1.4

There were several differences between the the use of time step size which show how size affects results. The conservation of energy, or lack of it, shows that smaller time steps are required to maintain accuracy. It is expected that total system energy should remain constant. However, as figure 1.3.2 shows, the energy actually increases when the time step is set to 0.005 seconds. Figure 1.3.4, which corresponds to the 0.00005 second time step, shows the system is nearly conserved. This result is also shown by figures 1.3.1 and 1.3.3 which show a variation in amplitude in the 0.005 second time step that is not present in the 0.00005 second time step. So forward Euler does not conserve energy, with larger time steps, but approaches energy conservation with very small time steps.

The variation on forward Euler, using implicit approximations, shows more accuracy with the conservation of energy at 0.005 second time step size. Figure 1.3.6 shows an oscillating, but stable offset, value of total specific energy. Also, as figure 1.3.5 shows, the position, velocity, acceleration, data has a much more stable amplitude which is consistent with the theory. This makes the implicit variation much more accurate for second order differential equation analysis as it nearly conserves energy.

Introduction 2.1

This problem was on searching and storing data contained within a .MAT file. A file “chr1_sect.mat” was provided. The task was to write a script which would load the contents of this .MAT file and search through the provided data. The data contained within this file was a 1 dimensional array, called “dna”, which contained a “dna” sequence. Once the data was loaded, the script was then required to search through and count the number of full proteins, as well as other statistics, in order to find a number of different features about this sequence of dna. These features were; total number of proteins, average length, maximum length, and minimum length. Once finished, the program printed out this data to the command window. The following report goes over the methods used, results, and a brief discussion on the results found.

Models and Methods 2.2

Once the .MAT file was loaded this problem became mainly two different problems: searching/storage and statistical analysis of the stored data. The raw data comes in the form of a 1 dimensional array of enumerated bases. Each base represents one of 4 bases: 1 = A, 2 = C, 3 = G, 4 = T; The bases come in triples call codons. Only complete codons are valid data sets, therefore searching through the array “dna” can only be done by 3’s.

The iterative method used for this search was a while loop which kept track of an iterating index variable, k. This loop, as well as another nested while loop which will be described following, is only incremented by 3’s to ensure only intacted codons are evaluated.

Pseudo Code:

```
While loop : until k =  length(dna) - 2

    -  Search for start codon
       -if start is found loop while: until an end-codon is found
                 - Store protein length data
                 - once end is found update k to the end of protein
       - else advance k by 3
```

An array called “proteins” stores the length of each complete protein found. This array is then used at the end of the program to compute the average length of a protein in this sequence of DNA. In addition to length, the longest protein and shortest proteins are kept track of until the entire DNA is searched.

Calculations and Results 2.3

Here is sample output of the code that displays all relevant data requested for this problem

```
Total Protein-coding Segments: 4388
Average Length  85.90
Maximum Length 1149.00
Minimum Length   6.00

The most used end codon is TGA
The least used end codon is TAG
The percentage of dna used in coding is 29.88 percent
```

Discussion 2.4

Based on the results from the program, the most used end codon triple is TGA. The least used codon is TAG and the percent of DNA used in the coding of proteins is only about 30% of the DNA sequence. For sequences such as the one posed, “ATGATGTAG”, an additional iteration through a complete protein could be made to check if it fits this description. My code recordes, at least temporarily the start index and end index of a protein before moving on to search the rest of the DNA. with these indexes, a for loop could be used to search only this protein and determine if it should really be measured as a length 6 protein. Then this value, instead of 9, could be stored as the length of that particular protein.