**ENED 1090: Engineering Models I**

**Homework Assignment #5**

**Due: Week of October 5th at the beginning of your Recitation Section**

**Problem 1:** Consider the code show below. Complete the table showing the values for the variables as the loop progresses. **Do this without MATLAB** first then check your results using MATLAB. *This problem was on the midterm last year.*

x = 1;

y = 1;

for k = 1:4

Sum = x + y;

x = y;

y = Sum;

end

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sum** | **x** | **y** |
| Before Loop | Not Defined Before Loop | 1 | 1 |
| k = 1 | 2 | 1 | 2 |
| k = 2 | 3 | 2 | 3 |
| k = 3 | 5 | 3 | 5 |
| k = 4 | 8 | 5 | 8 |

**Problem 2:** Consider the code show below. Complete the table showing the values for the variables as the loop progresses. **Do this without MATLAB** first then check your results using MATLAB. *This problem was on the midterm last year.*

t = [ 0 -2 4 7 2 ];

count = 0;

for k = 1:length(t)

if t(k) < 4

count = count + 1;

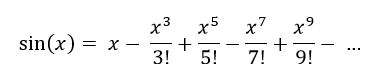
location(count) = k;

end

end

|  |  |  |
| --- | --- | --- |
|  | **count** | **location** |
| **Before Loop** | **0** | **Not Defined Before Loop** |
| **k = 1** | **1** | **1** |
| **k = 2** | **2** | **2** |
| **k = 3** | **3** | **5** |
| **k = 4** | **N/A** | **N/A** |
| **k = 5** | **N/A** | **N/A** |

**Problem 3:** Consider the Taylor series for a sine wave shown below.



Write a script file that does the following:

* Prompts the user for an angle (in radians) between 0 and 2pi.
* Prompts the user for the number of terms to use in the Taylor series
* **Uses a** **for loop** to calculate the estimate for the sine of the angle by summing the number of terms in the Taylor series specified by the user. (Note: don’t count the missing terms – the even powers of x – in your count of terms).
* Uses fprintf statement(s) to display the estimate of the sine, the actual sine, and the absolute value of the difference between actual and estimate. All of these values should be displayed using 6 places behind the decimal point.

Run your script to complete the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Angle** | **Number of Terms** | **Estimate of sin(Angle)** | **Actual sin(Angle)** | **Error** |
| **pi/2** | **1** | 0.924832 | 1.000000 | 0.075168 |
| **pi/2** | **2** | 1.004525 | 1.000000 | 0.004525 |
| **pi/2** | **3** | 0.999843 | 1.000000 | 0.000157 |
| **pi/2** | **6** | 1.000000 | 1.000000 | 0 |
| **15\*pi/8** | **5** | -1.792231 | -0.382683 | 1.409547 |
| **15\*pi/8** | **10** | -0.382665 | -0.382683 | -0.382683 |
| **15\*pi/8** | **13** | -0.382683 | -0.382683 | 0 |

**PASTE SCRIPT HERE:**

%Models HW5 Problem 3

clear; clc;

%input

x = input('Angle value between 0 and 2pi(rad)? ');

N = input('How many terms? ');

actual = sin(x);

estimate = 0;

%analysis

for n = 0:N

taylor\_term = ((-1)^n)\*(x^(2\*n+1))/factorial(2\*n+1);

estimate = estimate + taylor\_term;

end

error = abs(actual-estimate);

%output

fprintf('The estimate is %0.6f.\nThe actual is %0.6f.\n The error is %0.6f.\n',estimate,actual,error);

**Problem 4:**

1. Complete the table below following the directions given in the table. Do this by hand – no MATLAB script required.

|  |  |
| --- | --- |
| **Directions** | **Vector x** |
|  | x = [1 -2 3 5 4 2]; |
| Compare the 1st and 2nd entries in x. Swap the two entries if the first entry is larger than the second. Write down what the x-vector now looks like. | x = [-2 1 3 5 4 2]; |
| Now compare the 2nd and 3rd entries of the updated x vector (not the original). Swap the two entries if the second entry is larger third entry and record what the x-vector now looks like. | x = [-2 1 3 5 4 2]; |
| Repeat for entries 3 and 4. | x = [-2 1 3 5 4 2]; |
| Repeat for entries 4 and 5 | x = [-2 1 3 4 5 2]; |
| Repeat for entries 5 and 6 | x = [-2 1 3 4 2 5]; |

1. Write a script that does the following

* Creates the vector x = [1 -2 3 5 4 2];
* Uses a for loop to complete the procedure you just did by hand in part (a). ***Note: you must use a for loop to get credit for this part of the problem.***

***Hint: it might be helpful to leave the semicolon off of all of your statements in the for loop to see what is happening during each iteration to help you make sure that your code is working correctly.***

* Displays the final vector x after the for loop ends.

Run your script and see if you get the correct vector x at the end.

**PASTE RESULTS HERE:**

-2 1 3 4 2 5

**PASTE SCRIPT HERE:**

%Models HW5 P4

clear; clc;

%input

x = [1 -2 3 5 4 2];

%analysis

for k = 1:5

if x(k) > x(k+1)

s = x(k);

x(k) = x(k+1);

x(k+1) = s;

end

end

%output

disp(x);