**NPRE 100 Matlab Homework**

**Problem 1**

Plot the function: as a function of beta, where h = 1, plot the function over -4π <**** < 4π. Note that the tangent function is discontinuous.

For example:

x=linspace(0,pi/2-0.01,1000)

y=x+tan(x)

plot(x,y)

**Problem 2**

Find the first *n* (*n = 5*) roots of . Use two different methods:

1. Use the “fzero” function in Matlab; (use the help command if you don’t know how to use fzero)
2. Zoom in and around the zeros and do visual inspection.

**Problem 3**

Plot on the same graph two functions:

1)  vs. 

2)  vs. 

Use -4π <**** < 4π, *h = 1*.

**Problem 4**

Point Reactor Kinetics Equations with one delayed neutron group are:

For some initial conditions and specific values of the parameters in the equation, the solution of these equations is

,

where *A = 1.5, B = -0.5, w1 = 0.04, w2 = -5*. Plot two graphs of n(t) vs. time:

1) 0 < t < 0.1s

2) 0 < t < 10s

(You will learn about these equations in NPRE 247 and NPRE 455.)

**Problem 5**

Plot Watts equation (neutron fission spectrum):

The unit of E is in MeV. Plot the function over 0 < E < 7 MeV.

**Problem 6**

Plot the Maxwellian distribution of velocities. The equation is

where *N* represents the total number of particles; *m* is the mass of each particle; n is a quantity such that *ndv* gives the number of particles with velocities between *v* and *v+dv*; *k* is the Boltzmann constant and *T* the absolute temperature.

For this question, use

m = 1.67\*10-27 kg,

k = 1.38\*10-23 J/K,

N =1000,

1. Plot the distribution of number of particles vs. velocity at temperature T = 300K.
2. On the same graph, plot two other distributions at 600K and 1000K. Use “hold on” after the first use of plot function.

**Problem 7**

1. Plot log(x) (base 10) for 0 < x < 10. On the same graph, plot .
2. Recall that . Estimate the derivative by finite difference formula:

for increasingly smaller value of h, between 0.1 and 10-15 at x = 1. Make a table with comparison of exact (analytical) and approximate (numerical results).

|  |  |  |
| --- | --- | --- |
| h | f`(x) exact | f`(x) approximate |
| 0.1 |  |  |
| … |  |  |
| 10-15 |  |  |

**Problem 8**

Solve the decay equation numerically for 0 < t < 10

, where is 5.

with the initial condition N(t = 0) =100. Use finite difference approximation to the derivative:

Substituting approximate derivative to the original differential equation:

and rearranging, we can obtain *N(t)* at new time given solution at the previous time:

Show that the solution becomes more accurate as the time step *h* becomes smaller, by plotting on the same graph:

* Analytic solution
* Numerical solution with course time step
* Numerical solution with intermediate time step
* Numerical solution with fine time step

This is to demonstrate an example of a simple looping program in Matlab. Follow example solution on the next page.

**Reminder**

Every axis should be labeled with title and units.

**Example solution for Problem 8**

n = 100; %assign the number of time steps

lambda = 5;

finalt = 10;

deltat = finalt / n;

t = linspace(0,10,n);**[1]**

N(1) = 100;**[2]**

for i=1:(n-1)**[3]** %the index of the for-loop must be a variable

N(i+1)=N(i)-deltat\*lambda\*N(i); %finite difference for N(t)

end %for-loop must be ended with an *end* statement

plot(t,N); %plot results

Explanation:

[1] 0 and 10 mean initial time starts from 0 second and end at 10 seconds. n means there are n partitions in the domain of 0 to 10, which also means there are n+1 nodes from 0 to 10. t is a row vector with n+1 nodes. The time step size is (10-0) / n.

[2] In Matlab, the first element in a Matrix is N (1), not N (0). So the initial condition marks as N (1), not N (0). Before using a matrix, it should be defined. When using a for-loop, we should assign the first element value before the for-loop.

[3] “for” statement has the following general structure:

for index=expression

statement group

end