GE U111 ENGINEERING PROBLEM SOLVING & COMPUTATION, SPRING 2004
HOMEWORK #6 - TRAJECTORIES WITHOUT AIR FRICTION – PART I
BACKGROUND INFORMATION & EXERCISE 1

DATE ASSIGNED:	March 15, 2004
DATE DUE:	March 22, 2004 (at the beginning of class)
READING:	Chapter 3.1-3.2 and this Trajectories handout

HW 6 Prelab: Come to the lab with the Prelab worksheet assignments completed. This will be checked in lab.

This exercise will use MATLAB to calculate and plot the frictionless trajectories of a projectile such as you will study in physics. You will learn to use MATLAB to define variable names, use trig functions, solve a quadratic equation, define an array, use array multiplication and exponentiation (“ **.*** ” and “ **.^** ” in MATLAB) to calculate arrays of *x*- and *y*-points, and then plot these points to see the trajectory.

The problem is illustrated in Figure 1. A ball is projected at an angle of 60° with an initial velocity of 100 m/sec and traces out a parabolic trajectory as shown in Figure 1. We want to find the distance $x_2 - x_1$ for *several* different heights, *y*. The initial velocity can be broken into a horizontal and vertical component, V_{ox} and V_{oy}, and then the kinematic equations can be used for the *x*- and *y*-displacement. Since there is no horizontal force in the *x*-direction, the *x*-component (distance) after an elapsed time *t* is given by:

$$x = V_{ox}\, t = V_o \cos(60) t$$

In the *y*-direction (height) there is a constant downward acceleration vector, $g = -9.8$ m/s^2, as well as the initial velocity, $V_{oy} = V_o \sin(60)$, and the height of the ball is given by:

$$y = V_{oy} t + .5gt^2$$

Note: Assign *g* to be a negative number (downward vector). The instantaneous *y*-velocity is given by:

$$V_y = V_{oy} + gt$$

At the top of the trajectory $Vy = 0$, so the time to reach the top can be found from:

$$t_{top} = -V_{oy} / g.$$

In the exercise, we will select several heights (values of *y*), and solve the quadratic equation:

$$.5gt^2 + V_{oy} t - y = 0$$

to find the times to reach each of the given heights. In general, there will be two times at which the ball will be at a height y. (Why is this? Are there any exceptions?) For each given height *y*, we will find both times, t_1 and t_2, and the horizontal distance between these points, $x_2 - x_1 = V_{ox} (t_2 - t_1)$. We will approach this problem using an *array* for *y* so that we can solve for four different heights at once. We will then calculate the *x*- and *y*-coordinates for 101 values of time and plot the *y* vs. *x* positions to show the trajectory using a line graph.

So, the first step is to solve the Prelab and review trajectory physics, and the next step is to become acquainted with how MATLAB functions so that it may be used to solve this and other real-world problems.

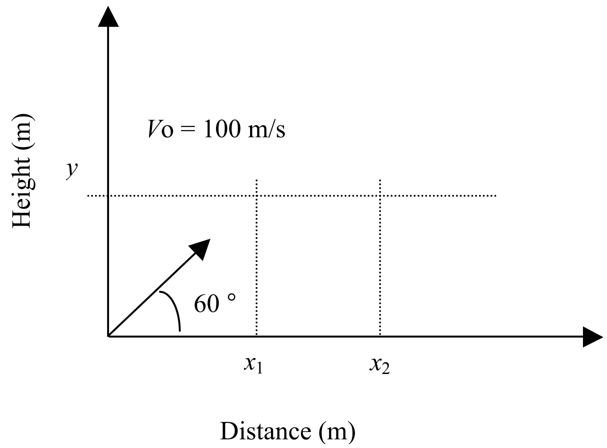

Figure 1. Height vs. Distance for projectile motion problem.

MATLAB EXERCISE:

Type all commands exactly as you see them, but do not type the >> or any comments in *{brackets}*. *Type the normal MATLAB comments that start with %. You are required to fill in some of the MATLAB commands & comments as part of this assignment.*

```
>> %Set up a diary and disk file.  First make sure you've inserted your diskette
>>cd a:                  {makes your floppy disk in drive a: the default drive}
>>diary intro_ _ _       {place your initials instead of __ __ __ }
                         {DIARY command makes record of session in file a:\intro_ _ _.dia}

>> %THIS LAB WAS COMPLETED BY ______      {place your name here in the comment}

>> % Define initial velocities & acceleration (g)%
>>g=-9.80                % acceleration of gravity 9.8 m/sec downward %

g =

   -9.8000
                         {MATLAB variables are case-sensitive; G not equal g}
>>G
??? Undefined function or variable 'G'.


>>initial_velocity=100   {variables can be up to 19 characters long}

initial_velocity =

   100

>>initial-angle=60       {hyphens in variable names are mistaken for minus sign, you cannot use them}
??? initial-angle=
Missing operator, comma, or semi-colon.  Error: Assignment cannot produce result

>>initial_angle=60;      {use semicolon after entry to suppress printing}
>>pi                     {pi is a built-in constant}

ans =

    3.1416
```

```
>>type command here       %conversion to angle in radians from degrees using pi %
                          {trig functions all use angles in radians }

angle_radians =           {this is a calculation, not an assignment}

    1.0472

>>who                     %lists defined variables %

Your variables are:

angle_radians       g                   initial_velocity
ans                 initial_angle

>> % Find velocity components and maximum height % {cos()is a built-in function}

>>Vox=initial_velocity*cos(angle_radians)


Vox =

   50.0000


>> type command here      %initial velocity for y %
                          {use up-arrow to recall previous command and modify to calculate Voy}
                          {Note: Voy is based on sin … sin( ) is one of the many built-in functions}

Voy =

   86.6025

>>time_to_top=-Voy/g

time_to_top =

    8.8370


>>max_height=0.5*g*time_to_top^2 + Voy*time_to_top     {note exponentiation operator ^}


max_height =

  382.6531

>> %Solve quadratic equation to find x2-x1 at y=0, 150, 300 and 400 m%


>>height=[0 150 300 400]      %define an array of 4 heights %

height =

     0   150   300   400      {what will happen when we solve for time at y=400?}
```

```
>>length(height)              %gives the array length %

ans =

     4

>>A=g/2;                     %Provide Comment Here                              %
>>B=Voy;                     %Provide Comment Here                              %
>>C=-height;                 %Provide Comment Here                              %

>>time_1=(-B+sqrt(B^2-4*A*C))/(2*A)  %solving for time_1 using quadratic equation
                                     {solve for all values of the height at once! }
time_1 =

        0           1.9464            4.7299            8.8370- 1.8815i

                             {last value is complex (why?)}

>> type command here         %solving for time_2 using quadratic equation %
                             {try using the up arrow to get prior command and make small changes}

time_2 =

  17.6740          15.7276           12.9441            8.8370+ 1.8815i

>>time_1(1)                  %display the first (1) element of time_1 array %

ans =

     0

>>time_1(1:3)                {1:3 corresponds to elements 1 2 3; Display elements 1-3 in array time_1}

ans =

         0    1.9464    4.7299            {These are the real number values from time_1}

>>rtime_1=time_1(1:3);       %define rtime_1 array not including complex answers %
>>rtime_2=time_2(1:3);       %Provide Comment Here                         %

>>x2_minus_x1=(rtime_2-rtime_1)*Vox          {subtracting arrays subtracts corresponding elements}

x2_minus_x1 =

  883.6994  689.0589  410.7064

>>                           % Find and plot the (x,y) points at y=0,150,300 m %

>>time=[rtime_1 rtime_2]           %combine two time arrays into larger array %

time =

         0    1.9464    4.7299   17.6740   15.7276   12.9441

>>length(time)               %Provide Comment Here                          %
```

```
ans =

     6

>>time*time                 {to multiply elements of two arrays, you must use .* }
??? Error using ==> *
Inner matrix dimensions must agree.
>>time.*time                {array multiplication is accomplished like this: .*, with the array notation}

ans =

         0    3.7885   22.3722  312.3698  247.3569  167.5486

>>y=0.5*g*time.*time+Voy*time  %Provide Comment Here             %

y =

         0  150.0000  300.0000    0.0000  150.0000  300.0000

>>x=Vox*time

x =

         0   97.3202  236.4965  883.6994  786.3791  647.2029

>>figure(1)                    %begin a new figure (Figure No. 1)%
>>plot(x,y,'o')                %plot x-y pairs we found as points %
>>xlabel('Distance(m)'); ylabel('Height(m)')

{Print out the graph now using File → Print in the Figure menu}
>>%Plot out the full trajectory along with the points %

>>N=0:100;                     %define a 101-point array N=0, 1, 2, 3, ... 100 %
                               {Don't forget the semicolon (;) to suppress displaying long arrays! }
>>max_time=max(time)           %Provide Comment Here                     %
max_time =

   17.6740
                               {"max" is a built-in MATLAB function}
>>TIME=N*max_time/100;         %make TIME 101 values from 0 to max_time %
>>length(TIME)                 %number of elements in array called TIME %

ans =

   101

>>XX=TIME*Vox;                 %Provide Comment Here                     %
                               {Again, ALWAYS use semicolon to suppress displaying long arrays!}
>>YY=0.5*g*TIME.^2+Voy*TIME;   {can also use .^ to do array exponentiation}


>>figure(2)                    %begin a new figure instead of overwriting fig 1%
>>plot(TIME,XX, TIME,YY)       %plot XX & YY on the same figure, TIME is x-axis %
>>xlabel('Time(s)');ylabel('Height and Distance(m)')   %Provide Comment Here    %


>>figure(3)                    %Provide Comment Here        %
```

```
>>plot(XX,YY,'-',x,y,'o')      %plot YY vs. XX as line; y vs. x as points %
>>xlabel('Distance(m)');ylabel('Height(m)')                    %label the axes
>>title('Trajectory for Vo=100 m/s, angle=60 degrees')       %title the graph
```

{Print out Figures 2 and 3 now using File ➔ Print in the Figure menu}

```
>>diary off                     %Stop recording from Command Window into diary %
```

When done, you should print your diary file (intro_ _ _.dia), remove your diskette… pick up your three printed plots. These are the things you will need to complete your homework & turn in.