
```

% The projectile problem with zero air resistance
% in a gravitational field with constant g.
% Written by Daniel T. Valentine .. September 2006
% Revised by D. T. Valentine ..... 2012/2016
% An eight-step structure plan applied in MATLAB:
%
% 1. Define the input variables.
%
g = 9.81; % Gravity in m/s/s.
vo = input('What is the launch speed in m/s?');
tho = input('What is the launch angle in degrees?');
tho = pi*tho/180; % Conversion of degrees to radians.
%
% 2. Calculate the range and duration of the flight.
%
txmax = (2*vo/g) * sin(tho);
xmax = txmax * vo * cos(tho);
%
% 3. Calculate the sequence of time steps to compute
% trajectory.
%
dt = txmax/100;
t = 0:dt:txmax;
%
% 4. Compute the trajectory.
%
x = (vo * cos(tho)) .* t;
y = (vo * sin(tho)) .* t - (g/2) .* t.^2;
%
% 5. Compute the speed and angular direction of the
% projectile. Note that vx = dx/dt, vy = dy/dt.
%
vx = vo * cos(tho);
vy = vo * sin(tho) - g .* t;
v = sqrt(vx.*vx + vy.*vy); % Speed
th = (180/pi) .* atan2(vy,vx); % Angular direction
%
% 6. Compute the time and horizontal distance at the
% maximum altitude.
%
tymax = (vo/g) * sin(tho);
xymax = xmax/2;
ymax = (vo/2) * tymax * sin(tho);
%
% 7. Display in the Command Window and on figures the output.
%
disp(['Range in meters =',num2str(xmax),', ' ...
' Duration in seconds =', num2str(txmax)]);
disp(' ')
disp(['Maximum altitude in meters = ',num2str(ymax), ...
', Arrival at this altitude in seconds = ', num2str(tymax)])
plot(x,y,'k',xmax,y(size(t)), 'o',xmax/2,ymax, 'o')

```

```

title(['Projectile flight path: v_o = ', num2str(vo), ' m/s' ...
      ', \theta_o = ', num2str(180*tho/pi), ' degrees'])
xlabel('x'), ylabel('y') % Plot of Figure 3.4.
figure % Creates a new figure.
plot(v,th,'r')
title('Projectile speed vs. angle')
xlabel('V'), ylabel('\theta') % Plot of Figure 3.5.
%
% 8. Stop.

```

```

%
%=====
%
%=====

```

```

function x = quadratic(a,b,c)
% Equation:
%  $a*x^2 + b*x + c = 0$ 
% Input: a,b,c
% Output: x = [x1 x2], the two solutions of
% this equation.
if a==0 & b==0 & c==0
disp(' ')
disp('Solution indeterminate')
elseif a==0 & b==0
disp(' ')
disp('There is no solution')
elseif a==0
disp(' ')
disp('Only one root: equation is linear')
disp(' x ')
x1 = -c/b;
x2 = NaN;
elseif b^2 < 4*a*c
disp(' ')
disp(' x1, x2 are complex roots ')
disp(' x1 x2')
x1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
x2 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
elseif b^2 == 4*a*c
x1 = -b/(2*a);
x2 = x1;
disp('equal roots')
disp(' x1 x2')
else
x1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
x2 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
disp(' x1 x2')
end
if a==0 & b==0 & c==0
elseif a==0 & b==0

```

```
else  
disp([x1 x2]);  
end  
end
```

Error using dbstatus

*Error: File: D:\lclin\2017 matlab class\in-class practice programs
\practice for students\practice_programs__Chapter3.m Line: 1 Column: 1
Function definitions are not permitted in this context.*

Published with MATLAB® R2016a