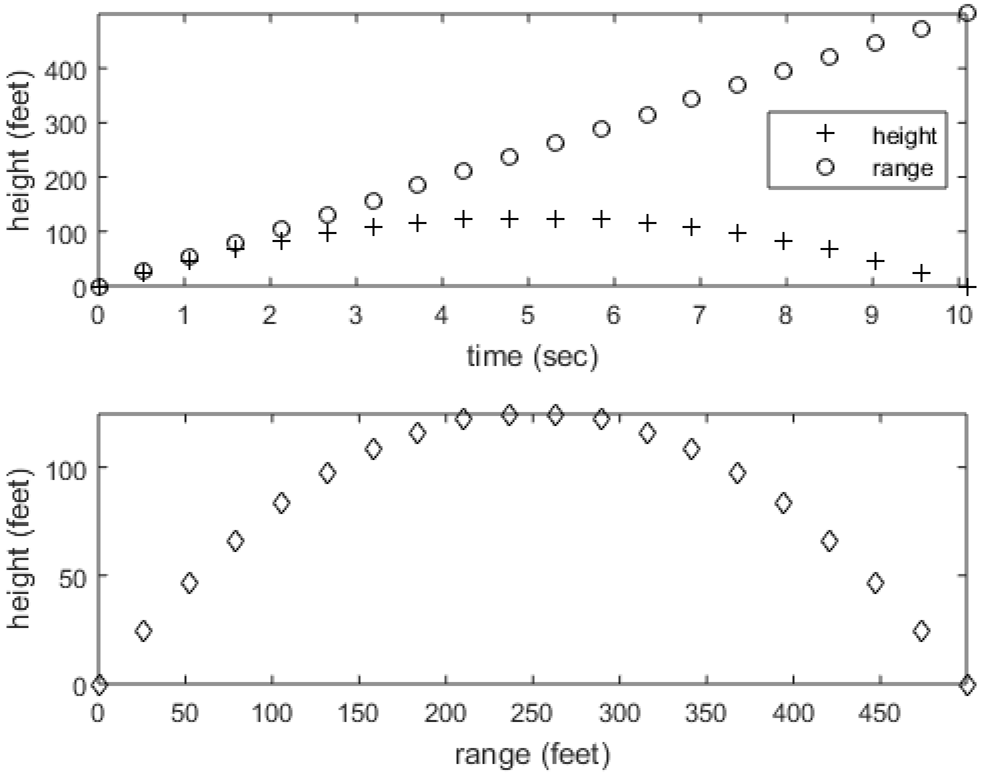
Table of Contents

```
% Robert W. Rost
%
% February 29, 2016
%
% numerical techniques
%
```

set up problem

```
clc
clear
close all
g=9.81;v0=70;th=45*pi/180;
thit=roots([-.5*g v0*sin(th) 0]);
t=linspace(0,thit(2),20);
y=-.5*g*t.^2+v0*sin(th)*t+0;
x=v0*cos(th)*t+0;
figure(1)
subplot(2,1,1)
plot(t,y,'k+',t,x,'ko')
xlabel('time (sec)');ylabel('height (feet)')
legend('height','range','location','best')
axis([0 thit(2) 0 max(x)])
subplot(2,1,2)
plot(x,y,'kd')
xlabel('range (feet)');ylabel('height (feet)')
axis([0 max(x) 0 max(y)])
```



numerical differentiation

```
fprintf('numerically find the velocity in x and y directions\n')
fprintf('and end up with as many values of v as in x and y\n')
pause
fprintf('use the forward difference for first two points\n')
pause
dt=diff(t);
vy=zeros(1,length(t));
vx=zeros(1,length(t));
% first point
vy(1)= (y(2) - y(1))/(t(2));
vx(1)= (x(2) - x(1))/(t(2));
% middle points
fprintf('use the 3 point for the points from 2 to end-1\n')
pause
for c=2:length(t)-1
    vy(c)= (y(c+1) - y(c-1))/(t(c+1) - t(c-1));
    vx(c)= (x(c+1) - x(c-1))/(t(c+1) - t(c-1));
end
% last point
fprintf('use the backward difference for the last two points\n')
pause
vy(length(t))= (y(length(t)) - y(length(t-1)))/(t(end)-t(end-1));
vx(length(t))= (x(length(t)) - x(length(t-1)))/(t(end)-t(end-1));
```

```
% analytical
fprintf('compute analytical velocities\n')
pause
vya=-.5*2*9.81*t+70*sin(th);
vxa(1:length(t))=70*cos(th);
fprintf('plot\n')
pause
figure(2)
subplot(2,1,1)
plot(t,y,'k',t,vy,'g+',t,vya,'r')
xlabel('time (sec)');ylabel('height (ft) and velocity (ft/sec)')
legend('height','numerical velocity','analytical velocity')
subplot(2,1,2)
plot(t,x,'k',t,vx,'g+',t,vxa,'r')
xlabel('time (sec)');ylabel('range (ft) and velocity (ft/sec)')
legend('range','numerical velocity','analytical velocity')
pause
```

numerically find the velocity in x and y directions

and end up with as many values of v as in x and y

use the forward difference for first two points

use the 3 point for the points from 2 to end-1

use the backward difference for the last two points

compute analytical velocities

plot

numerical integration

```
fprintf('find the integral from x=0 to x=289.1786 (x(12))\n')
dx=diff(x);
% left rieman sum
lrs=0;
for c=1:length(x)/2
    lrs = lrs + (y(c)*dx(c));
end
for c = 2:11
    rrs = rrs + (y(c) *dx(c));
end
% for c = 1:10
%     trap = trap + ((y(c+1)-y(c))*dx(c))
% end
figure(3)
for c=1:length(x)/2
    plot([x(c) x(c)],[0 y(c)])
    hold on
    plot([x(c) x(c+1)],[y(c) y(c)])
    plot([x(c+1) x(c+1)],[0 y(c)])
end
plot(x,y,'ko')
xlabel('range (ft)');ylabel('height (ft)');title('left Reiman sum')
% right reiman sum
rrs=0;
%for c=% add code
```

```
%    rrs=% add code
%end
figure (4)
for c=2:length(x)/2+1
    plot([x(c) x(c)],[0 y(c)])
    hold on
    plot([x(c) x(c-1)],[y(c) y(c)])
    plot([x(c-1) x(c-1)],[0 y(c)])
end
plot(x,y,'ko')
xlabel('range (ft)');ylabel('height (ft)');title('rigth Reiman sum')
% trap
trap=0;
%for c=% add code
%    trap=% add code
%end
figure (5)
for c=1:length(x)/2
    plot([x(c) x(c)],[0 y(c)])
    hold on
    plot([x(c) x(c+1)],[y(c) y(c+1)])
    plot([x(c+1) x(c+1)],[0 y(c+1)])
end
plot(x,y,'ko')
xlabel('range (ft)');ylabel('height (ft)');title('trapeziodal rule')
%
% Simpsons Rule
%
mysimp=0;
for c=1:2:length(x)/2
mysimp=mysimp+((y(c)+(4*y(c+1))+y(c+2))*(x(c+2)-x(c))/6);
end
figure (6)
for c=1:2:length(x)/2
    plot([x(c) x(c)],[0 y(c)])
    hold on
    plot([x(c) x(c+2)],[y(c) y(c+2)])
    plot([x(c+2) x(c+2)],[0 y(c+2)])
end
plot(x,y,'ko')
xlabel('range (ft)');ylabel('height (ft)');title('Simpson''s rule')
ta=thit(2);
%xa=x(length(x)/2+2);
xa=x(11);
ia=-(.5*9.81*ta^3)/3+(70*sin(3*pi/4)*ta^2)/2;
iyx=tan(th)*(xa^2/2)-(g/(2*v0^2*(cos(th))^2))*(xa^3/3);
fprintf('analytical = %0.5f\n',iyx)
fprintf('left rieman sum = %0.5f\n',lrs)
fprintf('right rieman sum = %0.5f\n',rrs)
fprintf('trapezodial = %0.5f\n',trap)
fprintf('simpson = %0.5f\n',mysimp)
```

find the integral from x=0 to x=289.1786 (x(12))

```
Undefined function or variable 'rrs'.

Error in lec8 (line 76)
    rrs = rrs + (y(c) *dx(c));
```

Published with MATLAB® R2015a