**Introduction:**

Problem Statement 1 We wrote a script to create a smooth line plot Eq. 1. (Analytical Solution)

Problem Statement 2 We wrote a script to numerically calculate the velocity using Euler’s method

Problem Statement 3 We wrote a script to numerically calculate the velocity using Euler’s method but we converged to the analytical solution by gradually reducing the step size (h) until the shape of the curve no longer changes.

Problem Statement 4 We wrote a script to numerically calculate the velocity using Heun’s method.

Problem Statement 5 We wrote a script to numerically calculate the velocity using modified Euler’s method and were also calculated the root mean square error.

Problem Statement 6 We compared the Heun’s method with step sizes of .5, .025 and .1 by Overlaying each case on two plots (one plot for Euler’s and one plot for Heun’s method), such that each plot has three plot lines corresponding to the various step sizes.

Eq. 1

Eq. 2

Eq. 3 Eq. 4 Eq. 5

Eq. 6

Eq. 7

Eq. 8

Eq. 9

Eq. 10

**Methods:**

Euler’s method: Euler’s method uses a simple linear extrapolation to estimate a y-value at a different point.

Heun’s method: Heun’s method determines two derivatives, one at the beginning and one at the end for the interval and then uses their average to obtain an improved estimate of the slope for the entire interval.

Modified Euler’s method: This technique uses the Euler’s method to predict a value of y at the midpoint of the interval. Mathematically, the y at the midpoint of an interval is represented as eq 7 and eq 8 it is used to find the slope at the midpoint. Finally, this slope is then used to extrapolate linearly by using eq 9.

11.2:

1. Started with clear and close commands.

2. Initialized all the parameters

3. Wrote out the analytical solution from problem 1

4. Created a stepsize parameter and used it as a iteration factor to create a time array

5. Created an array of zeros for velocity with same length as the time array

6. for look to loop the array with the eulers equation insided

7. plotted the results

11.4:

11. Started with clear and close commands.

2. Initialized all the parameters

3. Wrote out the analytical solution from problem 1

4. Created a stepsize array and used it as a iteration factor to create a time array

5. Created an array of zeros for velocity with same length as the time array

6. for look to loop the it as long as the length of a given array

7. Created a predictor value

8. Used the predictor value to create a predictor slope point

9. averaged the slopes

10. used linear interpolation

11. plotted the point

11.5:

11. Started with clear and close commands.

2. Initialized all the parameters

3. Wrote out the analytical solution from problem 1

4. Created a stepsize array and used it as a iteration factor to create a time array

5. Created an array of zeros for velocity with same length as the time array

6. for look to loop the it as long as the length of a given array

7. Created a slope parameter that calculated the slope at the half way point

8. used half way slope point in the linear interpolation

9. plotted the point

**Results:**

11.1

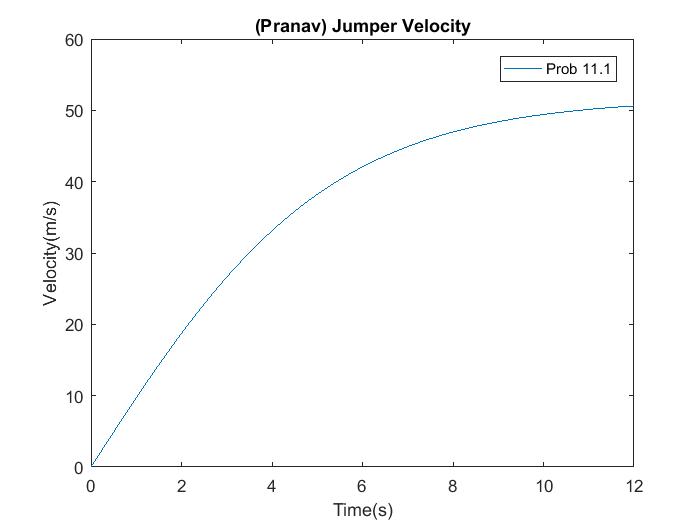


Figure 1 Analytical solution for velocity from the given v(t) function

11.2

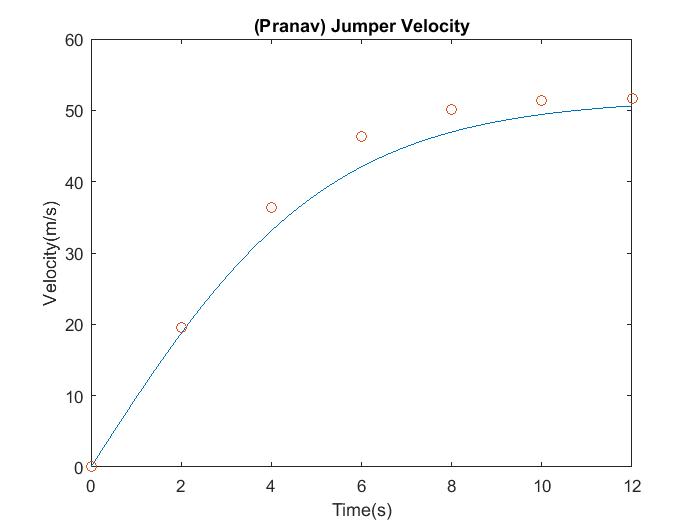


Figure 2 Euler’s method approximation (circles)

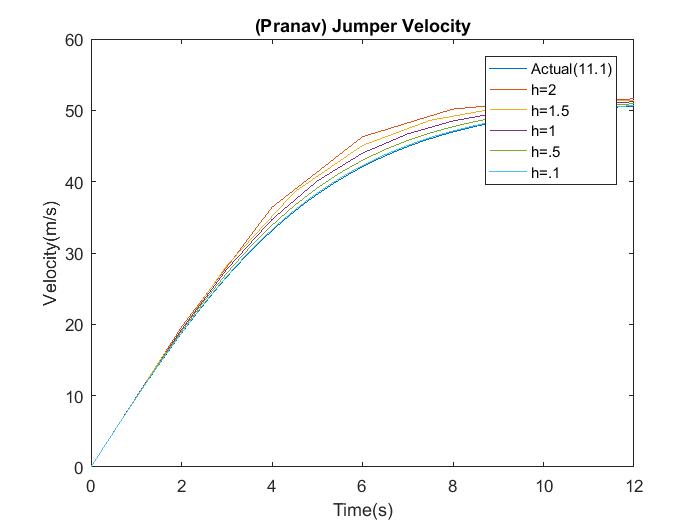
11.3

Figure 3 Euler’s method approximation of velocity

11.4

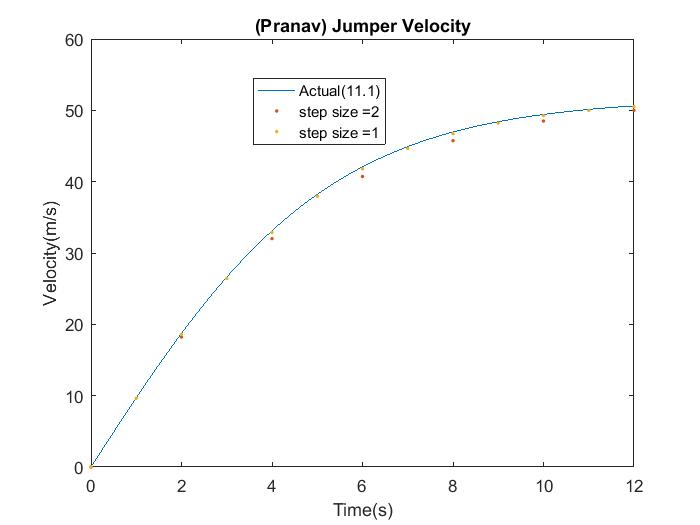


Figure 4 Heun’s method approximation (dots)

11.5

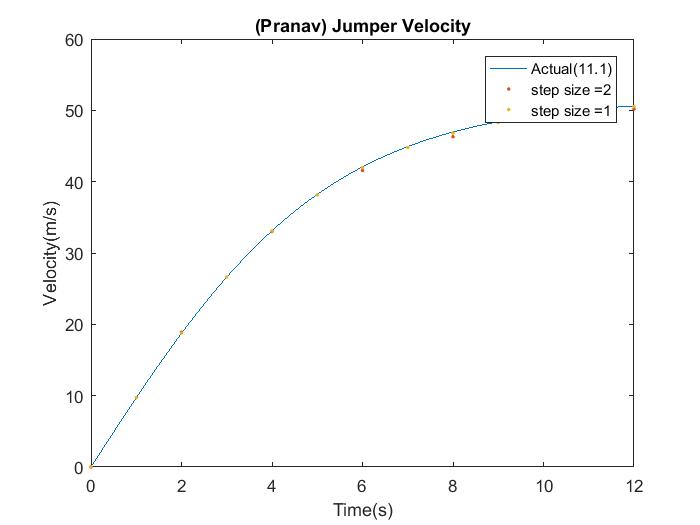


Figure 5 Modified Euler’s method approximation (dots)

>> RMS

Euler =

5.4849

Heuns =

2.1662

Mod =

0.7950

11.6

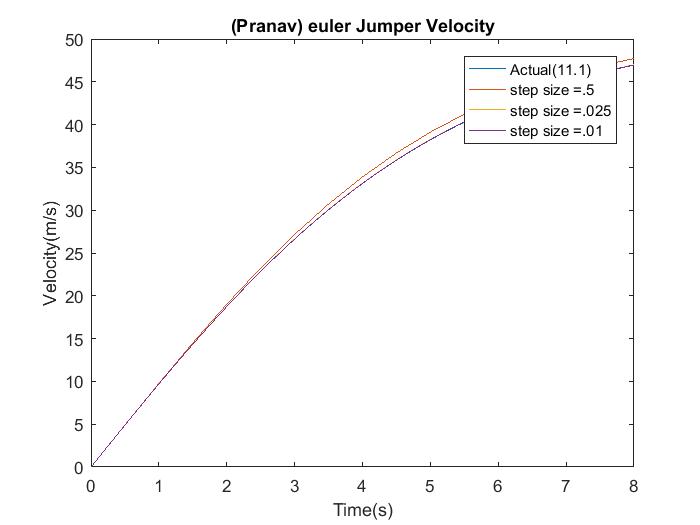


Figure 6 Euler’s method approximation with step sizes of h = 0.5, h = 0.025, and h = 0.01

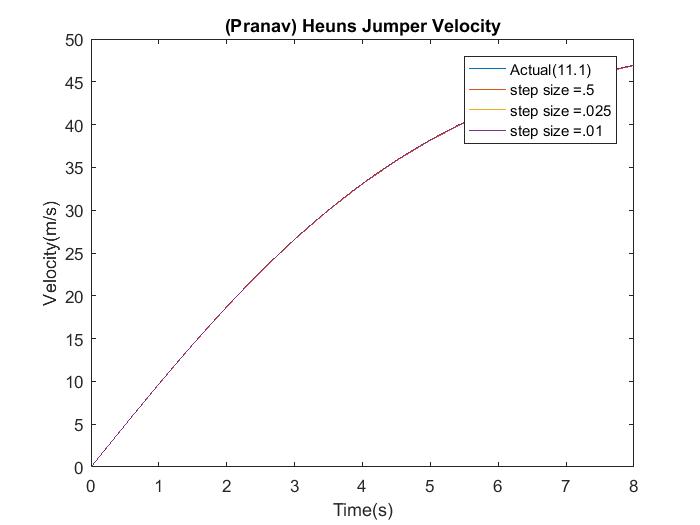


Figure 7 Heun’s method approximation of with step sizes of h = 0.5, h = 0.025, and h = 0.01

**Discussions:**

Discussions section and explain what you notice with the solution. Did the numerical solution

change? What does this tell you about convergence of both methods? Is one better than the other

in terms of convergence? Also, would you decrease the time step further? Why or why not?

In the solutions with as shown in figures 6 and 7 is that with a step size of .5 Heun’s method approximates the actual values to the point where you can’t tell them apart on the graph but for the Euler method the step size .5 doesn’t do as good of a job as Heun’s method. As you decrease the step size for the Heun’s method it doesn’t really affect the graph but for Euler method the decrease in step size resulted in the approximation being dramatically more accurate. So for Heuns method I wouldn’t want to change the step size but for Euler method I would use smaller step size like .025 and without the need to decrease it any further. The results correlate to the fact that Heun’s is a better method to approximate data because its faster since it doesn’t need as small of a step size as Euler’s method. The pitfalls for these methods are that the you are dependent on a derivative that can lead you in a wrong direction if the points have a weird correlation and also that it might take computer a lot of time and a relatively low stepsize to approximate values to a desired accuracy.

11.1

clear;

close;

t = 0:.01:12;

m = 68.1;

cd = .25;

g = 9.81;

v = sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t); %eqn11.2

figure(1)

plot(t,v)

title('(Pranav) Jumper Velocity');

legend('Prob 11.1')

ylabel('Velocity(m/s)');

xlabel('Time(s)');

11.2

clear;

close;

t=0:.01:12;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) Jumper Velocity');

stpsiz=2;

t2=0:stpsiz:12;

v2=zeros(1,length(t2));

v2(1)=0;

for ii=1:length(v2)-1

v2(ii+1)=v2(ii)+((g-cd\*(v2(ii)^2)/m)\*stpsiz); % eulers equation y(i+1) = y(i) + f(x,y)h

end

plot(t2,v2,'o')

11.3

clear;

close;

t=0:.01:12;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) Jumper Velocity');

stpsiz =[2 1.5 1 .5 .1];

for cntr=1:5

t2 = 0:stpsiz(cntr):12;

v2=zeros(1,length(t2));

v2(1)=0;

for ii=1:length(v2)-1

v2(ii+1)=v2(ii)+(g-(cd)\*((v2(ii)^2)/m))\*stpsiz(cntr);

end

plot(t2,v2)

end

legend('Actual(11.1)','step size =2','step size =1.5','step size =1','step size =.5','step size =.1')

11.4

clear;

close;

t=0:.01:12;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) Jumper Velocity');

stpsiz =[2 1];

for cntr=1:2

t2 = 0:stpsiz(cntr):12;

v2 = zeros(1,length(t2));

v2(1)=0;

for ii=1:length(t2)-1

predictor = v2(ii)+(g-(cd/m)\*(v2(ii)^2))\*stpsiz(cntr);

ypredicslope =g-(cd/m)\*(predictor^2);

slope =g-(cd/m)\*(v2(ii)^2);

avgslope = (slope+ypredicslope)/2

v2(ii+1)=v2(ii)+(avgslope)\*stpsiz(cntr);

end

plot(t2,v2,'.')

end

legend('Actual(11.1)','step size =2','step size =1')

11.5

clear;

close;

t=0:.01:12;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) Jumper Velocity');

stpsiz =[2 1];

for cntr=1:2

t2 = 0:stpsiz(cntr):12;

v2 = zeros(1,length(t2));

v2(1)=0;

for ii=1:length(t2)-1

midslope =v2(ii)+(g-(cd/m)\*(v2(ii)^2))\*stpsiz(cntr)/2;

v2(ii+1)=v2(ii)+(g-(cd/m)\*(midslope^2))\*stpsiz(cntr);

end

plot(t2,v2,'.')

end

legend('Actual(11.1)','step size =2','step size =1')

11.6

Euler

clear;

close;

t=0:.01:8;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) euler Jumper Velocity');

stpsiz =[.5 .025 .01];

for cntr=1:3

t2 = 0:stpsiz(cntr):8;

v2=zeros(1,length(t2));

v2(1)=0;

for ii=1:length(v2)-1

v2(ii+1)=v2(ii)+(g-cd\*(v2(ii)^2)/m)\*stpsiz(cntr);

end

plot(t2,v2)

end

legend('Actual(11.1)','step size =.5','step size =.025' , 'step size =.01')

Heuns

clear;

close;

t=0:.01:8;

m=68.1;

g=9.81;

cd=.25;

v=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t);

plot(t,v)

hold

ylabel('Velocity(m/s)');

xlabel('Time(s)');

title('(Pranav) Heuns Jumper Velocity');

stpsiz =[.5 .025 .01];

for cntr=1:3

t2 = 0:stpsiz(cntr):8;

v2 = zeros(1,length(t2));

v2(1)=0;

for ii=1:length(t2)-1

predictor = v2(ii)+(g-(cd/m)\*(v2(ii)^2))\*stpsiz(cntr);

ypredicslope =g-(cd/m)\*(predictor^2);

slope =g-(cd/m)\*(v2(ii)^2);

avgslope = (slope+ypredicslope)/2;

v2(ii+1)=v2(ii)+(avgslope)\*stpsiz(cntr);

end

plot(t2,v2)

end

legend('Actual(11.1)','step size =.5','step size =.025' , 'step size =.01')