Test 2:

Algorithm

This program uses the Euler Method in calculating the terminal velocity of a bike ride given three parameters; initial velocity, time interval, and density of medium.

The main equation used was:

Where:

– Velocity after time interval

– Velocity before time interval

– Force applied through the ride,

– Mass of the system,

– Drag Coefficient,

– Density of the Medium, for air, for fog.

– Area,

– Time interval, either or .

Depending on the part of the program, the values used in the code will depend on the conditions given. Since the equation is one that is constantly reiterating, the program will mainly be a single loop.

Code

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

#include <stdio.h>

using namespace std;

// This is the function that will be used to calculate the terminal velocity.

// The values that can be changed are the initial velocity, the time interval, and the density.

void TestBicycle(long double Velocity, long double dt, long double rho)

{

// Using the constants given.

long double T = 400, Force = 50, C = .5, A = .33, M = 70, Tint = 1, V = 0;

// Opening the file that will contain all the data that will be used.

ofstream Test;

Test.open("Bicycle.txt");

// Fixing the outbut numbers.

Test << fixed << setprecision(6);

// Creating the header.

Test << "Time (sec) Velocity (m/s)" << endl;

Test << 0.000000 << " " << Velocity << endl;

// The velocity initial will be stored for reasons.

V = Velocity;

// Creating the loop to calculate the program via Euler Method.

while (Tint <= (T / dt))

{

// This will take the velocity initial and update it each time the loop runs according to the Euler Method.

Velocity = Velocity + (Force / M) \* dt- (C \* rho \* A \* pow(Velocity, 2)) / (2 \* M)\*dt;

// Displaying the time in seconds and the velocity in meters per second

Test << Tint \* dt << " " << Velocity << endl;

// This will display the final velocity of the bike given the initial changing paramaters.

if (Tint == T / dt)

{

cout << "The final velocity for the following quantities:" << endl;

cout << "Initial velocity: " << V << " meters per second" << endl;

cout << "Time interval: " << dt << " seconds" << endl;

cout << "Density of medium: " << rho << " Kilograms per meter cubed" << endl;

cout << "Is: " << Velocity << " meters per second." << endl;

cout << "A file containing the data for the above values has been created."<< endl << endl;

}

// Increasing the counter.

Tint++;

}

// Closing the file.

Test.close();

};

int main()

{

// This is part a. of the test.

TestBicycle(0, .1, 1.225);

rename("Bicycle.txt", "Initial Velocity = 0, Time Interval = .1, Density = 1.225.txt");

// This is part b.

// This is for time interval of .5 seconds.

TestBicycle(0, .5, 1.225);

rename("Bicycle.txt", "Initial Velocity = 0, Time Interval = .5, Density = 1.225.txt");

// This is for time interval of .01 seconds.

TestBicycle(0, .01, 1.225);

rename("Bicycle.txt", "Initial Velocity = 0, Time Interval = .01, Density = 1.225.txt");

// This is for time interval of .001 seconds.

TestBicycle(0, .001, 1.225);

rename("Bicycle.txt", "Initial Velocity = 0, Time Interval = .001, Density = 1.225.txt");

// This is part c., the terminal velocity was 22.242827 meters per second, then the velocity of 25 meters per second is greater than the terminal velocity.

TestBicycle(25, .1, 1.225);

rename("Bicycle.txt", "Initial Velocity = 25, Time Interval = .1, Density = 1.225.txt");

// Assuming that the density is slightly higher than the density of dry air, I get that the density would be 1.425.

TestBicycle(0, .1, 1.425);

rename("Bicycle.txt", "Initial Velocity = 0, Time Interval = .1, Density = 1.425.txt");

// End the program.

return 0;

}

How to Run the Code

This code is written in C++ so in order to run it, the g++ compiler should be used. This compiler should already be in Omega. The file extension that seemed to work best is the .C extension. This code will display the final velocity and let the user know that a file for the given values had been created.

Results and Analysis

1. Write a code to calculate the final velocity with the above equation after 400 seconds. Consider and . Show the terminal velocity by presenting the data plot.
2. Examine the effect of the time steps on the final velocity after 400 seconds. Consider three different time steps, and . Provide all the plots.

Plot for

Plot for

Plot for

It can be seen that changing the time interval to make it either bigger, in the case, or smaller, as in the other two cases does not really change the final velocity, though especially in the case it made the program run slowly.

1. Redo part a) with initial velocity more than the terminal velocity. For this part use the time steps which you think is accurate enough from part b). Discuss the results from the plot.

Plot for

Since the force that is being applied is not enough to keep the bicycle going at a speed higher than the terminal velocity, the velocity of the bike will quickly drop to the terminal velocity. I let the since what I saw from part b), it comes to show that having a lower time step won’t really affect the final value and the curve.

1. Consider a scenario where the rider suddenly entered in a foggy road. How would you modify the equation (3) to take this into account? Write down the modified equation with your explanation, and find the terminal velocity. If you need to change any input values as given in (2), provide your justification.

Plot of velocity where

All the values are the same from part a) except for the density of the medium. The density value is slightly higher since fog is slightly denser than dry air. I assumed that the medium in was fog so the algorithm used did not change, but the force of drag increased, leading to a lower terminal velocity. The equation remained:

Conclusion

What I learned from the following parts:

1. The terminal velocity graph looked close to the same as that of the original bike problem. The only difference is that it could now starts from zero due to the equation changes.
2. As long as the time interval is below one percent of the total time interval, the interval should be fine. The other intervals don’t really improve the graphs.
3. Since the force needs to be enough to keep the bike moving, having a velocity higher than the force can support will not be able to hold, so the bike will reach equilibrium.
4. Since fog is denser than air, the terminal velocity in fog is lower than that in air.

Lastly, the Euler Method is good with approximating linear phenomena such as this one.