15/09/2017

EMTH171: Case Study 1

Newton’s Method for vehicle power

Daniel Wadsworth-dwa94

[dwa94@uclive.ac.nz](mailto:dwa94@uclive.ac.nz)

28208782

Jaime sequeira- jfs73

[jfs73@uclive.ac.nz](mailto:jfs73@uclive.ac.nz)

42408823

**Exercise 1**

**Introduction**

In exercise 1, an electric motor was required to pull a load on a sled with a combined mass of 1000 kilograms up an incline of 45 degrees to the horizontal. The power output by the motor is best modeled using the equation

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where α and β are constants measuring 1W.s2/rad2 and 314.16rad/s respectively. ω is the rotational speed of the shaft in the motor (rad/s). This equation was modeled with the graph below to show the relationship between the shaft speed and the power output from the motor.

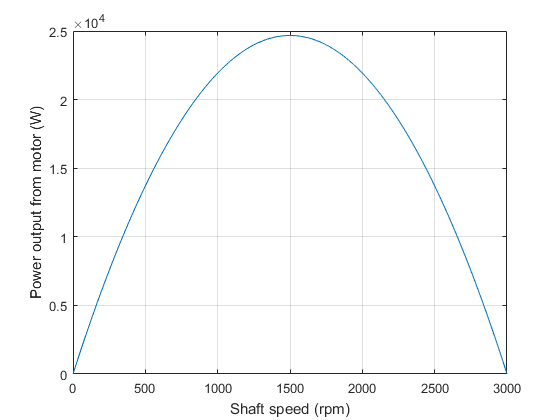


Figure 1: Motor output power versus shaft speed

A gear box was connected to the motor to with a ratio of 20:1. This idea was used to create an equation of the tangential speed of the pulley.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Where v is the tangential speed of the pulley (m/s), R is the radius of the pulley (m), ω is the rotational speed of the motor (rad/s) and rGB is the ratio of the gearbox. By rearranging this equation, an equation with the rotational speed of the motor being the dependent variable was formed.

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

This equation was substituted into equation (1) which meant that the tangential speed of the pulley (the speed of the sled) was the only unknown variable. This created the equation

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

The sled system had two forces to overcome to be able to travel at a constant speed: its weight force and friction force acting against the movement of the sled. The power required to overcome the weight force is described using the following equation:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Where m is the combined mass of the sled and the load inside it (kg), g is the gravity constant (9.81 m/s2), θ is the angle of the incline (rad) and v is the current velocity of the sled (m/s).

The power required to overcome the friction force acting against the movement of the system is described using the following equation:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

where cf is the friction coefficient and the other variables are the same as stated in equation (5).

When the sled is moving at a constant speed, the power output by the electric motor must be equal to the power demanded by the sled. Therefore, the following equation is formed to describe this situation:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

This was rearranged to create a quadratic equation in terms of the velocity of the sled.

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

The function, f(v), evaluates to zero when the power output by the motor is equal to the power demanded by the sled. Since this is a quadratic equation, the roots are needed to find these certain velocities. An estimate of when the power output is matched by the power demanded could be found by analyzing the graph below to find where the line for the power output and line for the power demanded intersect.

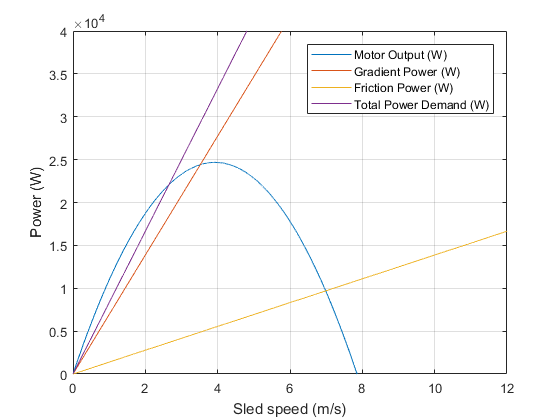
****

Figure 2: Motor power and power demanded as a function of v

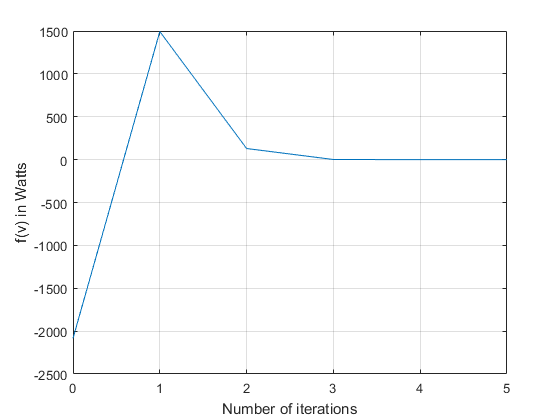
Newton’s method of finding roots of equations was used using Matlab to find at what velocity the two lines intersected. Newton’s method iteratively uses the equation

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

where xn+1 is the next guess for the velocity of the sled, xn is the current velocity, f(xn) is the value of the function evaluated with the current velocity and f’(xn) is the derivative of the function evaluated with the current velocity. The derivative used in the equation was the derivative of equation 8 which is as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

**Results and Analysis**

After five iterations of Newton’s method using Matlab, the root of the equation was found to be 2.6514m/s. This was achieved by using an initial guess of 2.000m/s. This meant the sled had to move at a constant speed of 2.6514m/s up the slope for the power demanded by the sled to be equal to the power output by the motor. A plot comparing f(v) with every iteration of Newton’s method can be seen below in Figure (3). However, some discrepancies were discovered when different initial guesses were used in Newton’s method. Since the movement of the sled is described with a quadratic equation, this equation provides two roots. Hence, with an initial guess using a velocity of zero or lower, the root is calculated to be zero. This is a valid root as at this velocity there is no power being output by the motor or power being demanded by the sled. However, the logical solution to this problem is that the sled travels at a constant speed of 2.6514m/s.

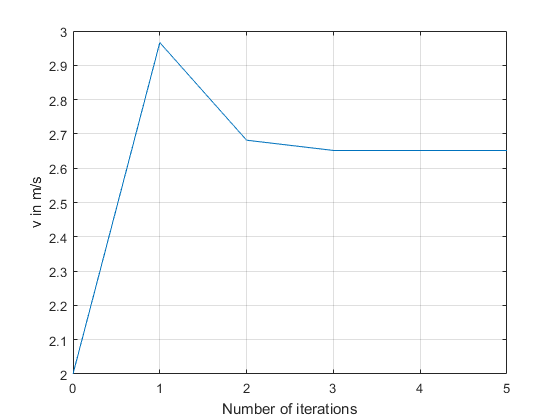


Figure 4: A plot of the sum of the power output and demanded (f(v)) against the velocity

**Conclusions**

Figure 3: A plot of v against the iteration number (convergence plot).

The aim of this Exercise was to find what velocity a sled must climb up an incline for its velocity to remain constant. This occurred when the power demanded by the sled was equal to the power output by the electric motor which was pulling it up. Newton’s method of finding roots to quadratic equations using Matlab provided two real roots to the equation: 0m/s and 2.6514m/s. While both solutions were valid, 2.6514m/s was the solution chosen as at 0m/s, the sled does not move up the incline at all.

**Exercise 2**

**Introduction**

When designing a system that requires mechanical energy it is important to know if the intended motor is going to be able to supply enough power to system to do the work in an acceptable time frame. This report uses Newton’s method (Equation (9)) to find vehicle top speed and acceleration of a typical family car, determined by the engine power output and the demand for power. This case covers the demand for power required to maintain speed against resistive forces of friction and aerodynamic drag, and the power required to accelerate a typical family road car on the open road in no wind conditions. The maximum engine speed of a typical family car is generally limited at 7000rpm to reduce risk of breakage and excessive wear as shown in figure 2.1.

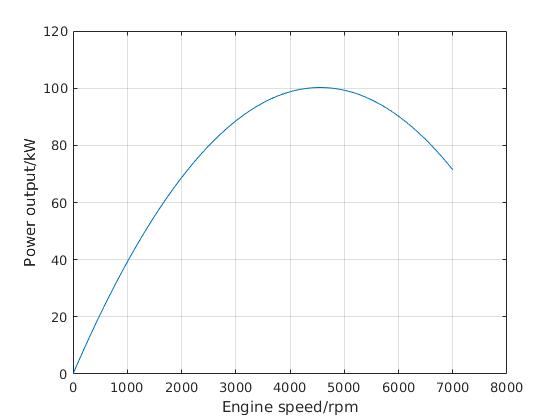


Figure 2.1: Power output of engine as a function of speed

Modelling the power output of a petrol engine as a function of speed can be shown using the quadratic equation below:

|  |  |  |
| --- | --- | --- |
|  | ω - β | (2.1) |

Where α and β are positive constants with units of and. ω is the engine speed in the SI units of . To obtain the more usual rpm the below equation was used

(2.2)

To show the above engine speed in the more intuitive vehicle road speed requires information on the gearing of the vehicle. The engine (ω rad/s) turns the gearbox and the final drive each changing the shaft speed. The wheel turns at the rotational speed of .

For every turn of the gearbox output shaft the crankshaft turns times. For every turn of the wheel the gearbox ouput shaft turns times. Thus the wheel rotational speed is given by

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

Which relates the rotational wheel speed to the SI units m/s. We can find that the road speed V can be described as

|  |  |  |
| --- | --- | --- |
|  | = | (2.4) |

is the number of turns of the wheel per second. The road speed for a given engine speed is shown by the equation

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

The power output of the engine as a function of road speed is given by

|  |  |
| --- | --- |
|  | (2.6) |

Where K is a constant (unit of meters) and derived as follows:

|  |  |
| --- | --- |
|  | (2.7) |

The car will experiencing forces resisting movement, in this case study forces considered are aerodynamic drag, rolling resistance, and gravity relevant if there is a gradient.

A vehicle, as with any object moving through a fluid is affected by aerodynamic drag. If insufficient power is supplied to the vehicle the drag force will cause a reduction in speed. The power required to overcome drag is the equation of drag force multiplied by velocity.

|  |  |  |
| --- | --- | --- |
|  |  | (2.8) |

Where is the drag coefficient given for this case study, A is frontal area of vehicle, ρ is density of air and V is speed of the vehicle.

Traveling uphill the vehicle the power consumed is equal to the potential energy gained by climbing. This power is shown using the following formula

|  |  |  |
| --- | --- | --- |
|  |  | (2.9) |

Where θ is the angle to the horizontal of the road in radians.

Rolling resistance is the rubber deforming when contacting the ground and going over bumps and protrusions. Power to overcome rolling resistance is calculated from the rolling resistance force multiplied by velocity.

|  |  |  |
| --- | --- | --- |
|  |  | (2.10) |

Where is the given rolling resistance value of an ordinary car tire on asphalt.

If the engine power supplied is greater than the power dissipated by resistive forces, the vehicle accelerates at a rate calculated by the following equation.

|  |  |  |
| --- | --- | --- |
|  |  | (2.11) |

The power balance is the sum of the total power demand equal to the power output. To put this into a form useful to the Newton’s method the power output is subtracted from both sides resulting in the following equation.

|  |  |  |
| --- | --- | --- |
|  | = 0 | (2.12) |

The derivative to the equation above is as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (2.13) |

**Results and Analysis**

**Test Case 1: No gradient and no acceleration.**

The MATLAB script in the appendix was used to solve for the speed (V). With an initial guess of 50m/s, a top speed of 54.88 m/s was acquired in 3 iterations which is equal to approximately 4500 rpm as can be seen in Figure (2.1). The initial guess was chosen from the graphical representation of the problem in Figure (2.2). If an initial guess of 26 or below was chosen the Newton method found the root to be zero. Although it is possible that this is a correct root for the quadratic equation, it is of little use for this case study so can be ignored. An initial guess of 27m/s converged on a root of close to -282m/s. This behavior was not predicted, almost nonsensical and likely occurred due to the function being close to zero or the derivative being close to undefined. When 29m/s was used as an initial guess, a root of 54.88m/s was found. Although this seems to be consistent with an accurate guess some strange behavior was observed. The second iteration found the root to be well over 1000 m/s which required 12 iterations to converge on the final root as opposed to 3 for an initial guess of 50m/s. When looking at the graph, 29m/s is clearly a poor choice of initial guess so this behavior is not of much significance to this case study. When the initial guess was 40 or above the expected root of approximately 55m/s was found, consistent with Figure 2.2 below. This is where the lines for the output power and power demanded intersected.

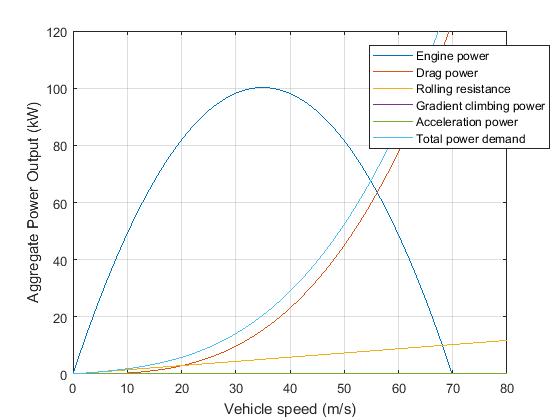


Figure 2.2: Power output as a function of speed, Case 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | v(m/s) | f(v) watts | | Initial guess | 50 | -2926.1 | | 1st | 55.501 | 4178 | | 2nd | 54.891 | 53 | | 3rd | 54.883 | 0 |   Figure 2.3: Table comparing the current velocity with the sum of the power demanded and output (f(v)) for case 1. | Figure 2.4: Convergence plot of v against the number of iterations. | Figure 2.5: Sum of power output and demanded (f(v)) against the number of iterations. |

**Case 2: Gradient of 10 degrees, no acceleration.**

From viewing Figure 2.3 an initial guess of 30 was chosen that converged on a root of 32.39 m/s. As expected this is below the result of case 1 due to the vehicle losing energy proportional to the gain in gravitational energy. An initial guess of below 15 converged on a root of zero. Although mathematically correct it is of little use to the case study. With an initial guess Newton’s Method found a trivial root of -260m/s. This is likely to be due to either the function or the derivative of the function being close to zero.

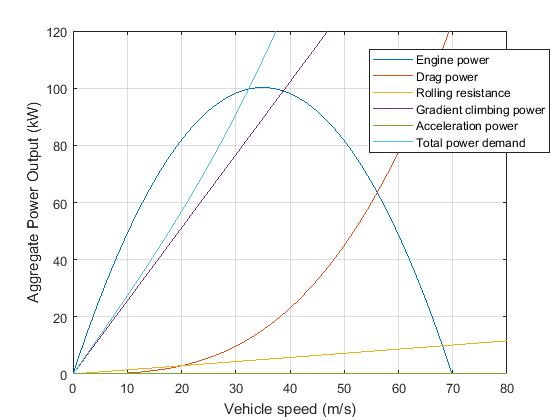


Figure 2.6: Power output as a function of speed, case 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | v(m/s) | f(v) watts | | Initial guess | 30 | -7497.1 | | 1st | 32.6208 | 792.8 | | 2nd | 32.3922 | 6.1 | | 3rd | 32.3904 | 0 |   Figure 2.7: Table comparing the current velocity with the sum of the power demanded and output (f(v)) for case 2. | Figure 2.8: Convergence plot of v against the number of iterations. | Figure 2.9: Sum of power output and demanded (f(v)) against the number of iterations. |

**Case 3: -5 degree gradient, no acceleration.**

As this case contains a negative gradient a higher final speed is expected as the gain in gravitational potential energy is added to the final speed. From looking at the graphical representation a good initial guess of 60m/s was chosen, this converged on the root of 65.12m/s. Much faster than the two previous cases as expected. An initial guess of 31 and below found the root of zero, which can be ignored in this case. An initial guess between 32 and 35 found the nonsense root of -293m/s. An initial guess of above 35 found the expected behavior of 65m/s.

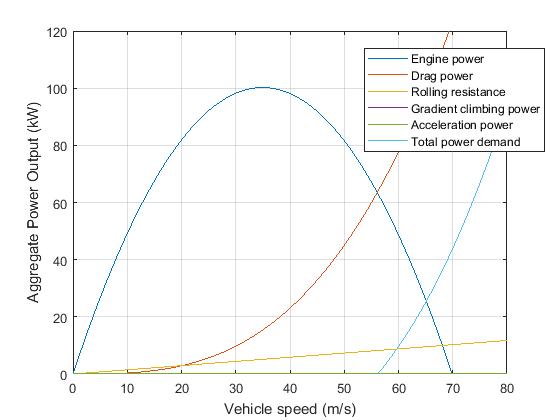


Figure 2.10: Power output as a function of speed, case 3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | v(m/s) | f(v) watts | | Initial guess | 60 | -390.85 | | 1st | 65.692 | 482.7 | | 2nd | 65.129 | 4.8 | | 3rd | 65.124 | 0 |   Figure 2.11: Table comparing the current velocity with the sum of the power demanded and output (f(v)) for case 3. | Figure 2.12: Convergence plot of v against the number of iterations | Figure 2.13: Sum of power output and demanded (f(v)) against the number of iterations. |

**Case 4: No gradient, 1m/s2 acceleration.**

From looking at the graph a reasonable initial guess of 40m/s was chosen. At this initial value a root of 42m/s was found. It makes sense that the final velocity is less than case1 as some power is being distributed to the acceleration of the vehicle. It is less obvious how this case relates to a gradient of 10 degrees, that it is faster does not seem to be unusual.

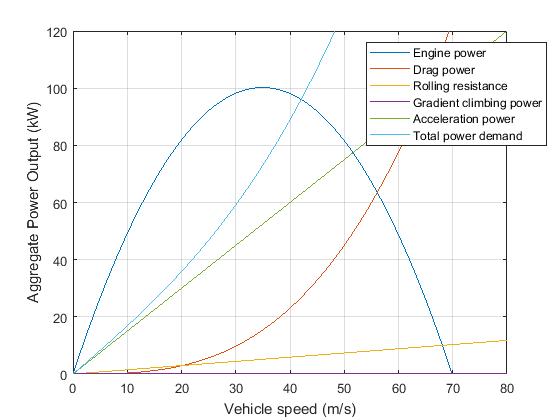
An initial value of 21 m/s gives the nonsense value of -270m/s. An initial guess of 22m/s gives the unusual second guess of 1100m/s, and takes 12 iterations to find the root.

Figure 2.14: Power output as a function of speed, case 4.

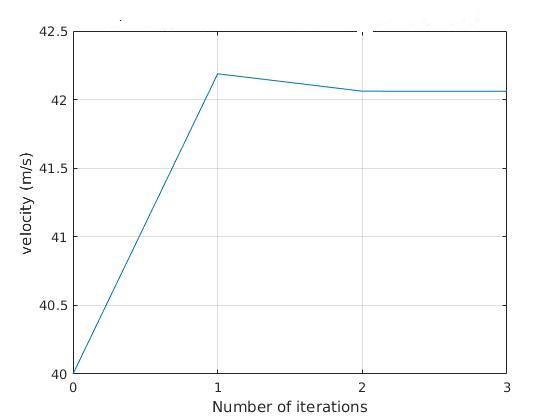
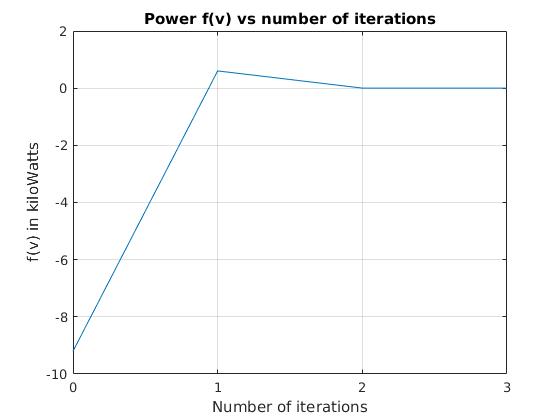


Figure 2.17: Sum of power output and demanded (f(v)) against the number of iterations.

|  |  |  |
| --- | --- | --- |
|  | v(m/s) | f(v) watts |
| Initial guess | 40 | -920.23 |
| 1st | 42.188 | 603.7 |
| 2nd | 42.061 | 2.1 |
| 3rd | 42.060 | 0 |

Figure 2.15: Table comparing the current velocity with the sum of the power demanded and output (f(v)) for case 4.

Figure 2.16: Convergence plot of v against the number of iterations.

**Conclusions**

The maximum engine power output was found to be 100kW at approximately 4500 rpm found iteratively using Newton's Method. Case1 (no gradient) with an initial root of 50m/s converged to 54.8832m/s in 3 iterations. Case2 (10degree gradient) with initial root of 30m/s converged on 32.3904m/s in 3 iterations. Case3 (-5 degree gradient) with initial root of 60m/s converged to the final velocity of 65.1241m/s after its third iteration. Case 4 (1ms-2 acceleration) converged to 42.0609m/s after an initial guess of 40m/s. All these speeds seem reasonable for a typical family car in each situation. All initial guesses were found from graphical observation of power output vs vehicle speed. Poor initial guesses had varied effects on the Newton method to find the roots.

Initial guess too low resulted in the real, but not very useful, root of 0 to be found. Initial guess where either the function or the derivative where close to zero resulted in unwanted behavior such as roots well below -200m/s. As well as not expecting a negative root it is very unlikely that a factory typical family car could set a world land speed record while in reverse gear. Other poor choice of initial guess may have resulted in converging on the correct root, but have very inaccurate iterations and converge after many more iterations than required.

**Appendix**

% EMTH171 Case Study 1

% Exercise 1: Electric motor pulling a sled

% Daniel Wadsworth and Jaime Sequeira

clear; clc; close all

% Set up variables, constants and power equations

theta = pi/4; % Angle of incline (Radians)

gravConst = 9.81; % Gravity constant (m/s^2)

rGB = 20; % Gearbox ratio

alphaCons = 1; % Alpha constant for finding power output of motor (Ws^2/rad^2)

betaCons = 100\*pi; % Beta constant for finding power output of motor (rad/s)

radiusPulley = 0.5; % Radius of the pulley (m)

massCombined = 1000; % Mass of the sled and load (kg)

cFriction = 0.2; % Coefficient of friction used

% Function handles to deal with amount of power demanded and output by the system (all in watts)

outputEngine = @(v) alphaCons\*((betaCons\*rGB\*v/radiusPulley)-...

(rGB^2\*v.^2/radiusPulley^2));

weightPower = @(v) massCombined\*gravConst\*sin(theta)\*v;

frictionPower = @(v) cFriction\*massCombined\*gravConst\*cos(theta)\*v;

totalPowerDemanded = @(v) weightPower(v) + frictionPower(v);

f = @(v) totalPowerDemanded(v) - outputEngine(v);

% Setting anonymous functions for derivatives of the functions above

derivOutput = @(v) alphaCons\*((betaCons\*rGB/radiusPulley)-...

(2\*rGB^2\*v/radiusPulley^2));

derivWeight = @(v) massCombined\*gravConst\*sin(theta);

derivFriction = @(v) cFriction\*massCombined\*gravConst\*cos(theta);

d = @(v) derivWeight(v) + derivFriction(v) - derivOutput(v);

% Plotting graph for engine power output against engine speed (rpm)

degreesToRadians = pi/30; % Convert angles to SI radians

enginePower = @(omega) (alphaCons\*betaCons\*omega)-(alphaCons\*omega.^2);

omega = 0:3000;

plot(omega,enginePower(omega\*degreesToRadians))

xlabel('Shaft speed (rpm)'); ylabel('Power output from motor (W)');grid on

% Plotting graph for all powers in w against v in m/s

v=0:0.001:12; % Creating data for the x axis

figure(2), plot(v,outputEngine(v),v,weightPower(v), ...

v,frictionPower(v),v,totalPowerDemanded(v))

legend('Motor Output (W)', 'Gradient Power (W)', 'Friction Power (W)', 'Total Power Demand (W)')

ylim([0 40000]);

xlabel('Sled speed (m/s)');ylabel('Power (W)');grid on

% Iterating to find the roots using Newton's method

N = 50; % Max number of iterations

root=2; % The initial guess

velocityArray(1) = root; % Create array for v used

tol = 1\* 10 ^(-4); % Tolerance for iterations

residualArray(1) = abs(0 - f(root));

differenceArray = zeros(1, N);

for x=1:N

root = root - (f(root)/d(root)); % Newton's method formula

velocityArray(x+1) = root;

residualArray(x) = abs(0 - f(x));

differenceArray(x) = abs(velocityArray(x+1) - velocityArray(x));

if abs(f(root)) < tol & abs(velocityArray(x+1) -...

velocityArray(x)) < tol % Checks if root is found

break

end

end

% Plotting convergence (v against f(v))

figure(3)

xAxis = 0:1:length(velocityArray)-1;

plot(xAxis,velocityArray)

xlabel('Number of iterations'), ylabel('v in m/s')

grid on; xticks([0:(length(velocityArray))])

% Plotting power against the number of iterations

figure(4)

xAxis = 0:1:length(velocityArray)-1;

plot(xAxis,f(velocityArray)),

xlabel('Number of iterations'), ylabel('f(v) in Watts')

grid on; xticks([0:length(velocityArray)])

% Case study part two

% Car motor power

clc; clear; close all;

% Initilizing variables (Units)

mass = 1500; %(kg)

drag\_coefficient = 0.30; %unitless

frontal\_area = 2.0; %(m^2)

rolling\_resistance = 0.010; %unitless

wheel\_radius = 0.205; %(m)

alpha = 420; %(ws/rad)

beta = 0.440; %(ws^2/rad^2)

drive\_ratio = 3.50; %unitless

gear\_ratio = 0.80; %unitless

gravity = 9.81; %(m/s^2)

air\_density = 1.2; %(kg/m^3)

K = wheel\_radius / (drive\_ratio \* gear\_ratio); %(m)

% Variables that change for each case

degrees2radians = 2\*pi/360; % constant for unit conversion

theta = 0; %(rad)- initial gradient angle

theta = theta \* degrees2radians;% puts theta in desired unit of radians

acceleration = 1; % (m/s^2)initial value of acceleration

% Information for plotting first graph

to\_rpm = 60 / (2\*pi); % Converts m/s into rpm

watt2kW = 1/1000;

omega = 1:733;

engine\_power\_rpm = ((alpha .\* omega) - (beta .\* omega.^2));

figure(1)

plot(omega\*to\_rpm, engine\_power\_rpm\*watt2kW)

ylim([0,120])

xlim([0,8000])

xlabel('Engine speed/rpm')

ylabel('Power output/kW')

grid on

%function handles (all units in watts):

power\_to\_overcome\_drag = @(v) drag\_coefficient \* frontal\_area ./ 2 .\* air\_density .\* v.^3;

gravity\_component = @(v) mass .\* gravity .\* sin(theta).\* v;

power\_to\_overcome\_rr = @(v) rolling\_resistance .\* mass .\* gravity .\* cos(theta) .\* v;

max\_speed = @(v) mass .\* acceleration .\* v;

engine\_power = @(v) ((alpha .\* v ./ K) - (beta .\* v.^2 ./ K .^ 2));

power\_to\_overcome\_gradient = @(v) mass \* gravity \* sin(theta) \* v;

acceleration\_power = @(v) mass \* acceleration \* v;

total\_power\_demand = @(v) gravity\_component(v) + power\_to\_overcome\_drag(v) + power\_to\_overcome\_rr(v) + acceleration\_power(v);

% information for plotting power vs speed

v = 0:100;

figure(2)

v\_kph = v \* 3.6; % converts m/s to km/h

watt2kW = 1/1000; % conversion of watts to kilowatts

plot(v, engine\_power(v)\*watt2kW, ...

v, power\_to\_overcome\_drag(v)\*watt2kW, ...

v, power\_to\_overcome\_rr(v)\*watt2kW, ...

v, power\_to\_overcome\_gradient(v)\*watt2kW, ...

v, acceleration\_power(v)\*watt2kW, ...

v, total\_power\_demand(v)\*watt2kW)

legend('Engine power','Drag power','Rolling resistance', ...

'Gradient climbing power','Acceleration power','Total power demand')

xlabel('Vehicle speed (m/s)')

ylabel('Aggregate Power Output (kW)')

ylim([0,120])

xlim([0,80])

grid on

%total power is:

f = @(v) power\_to\_overcome\_drag(v) + gravity\_component(v) + power\_to\_overcome\_rr(v)+ max\_speed(v) - engine\_power(v);

% derivatives function handles

d\_power\_to\_overcome\_drag = @(v) 3 .\* drag\_coefficient .\* frontal\_area ./ 2 .\* air\_density .\* v.^2;

d\_gravity\_component = @(v) mass .\* gravity .\* sin(theta);

d\_power\_to\_overcome\_rr = @(v) rolling\_resistance .\* mass .\* gravity .\* cos(theta);

d\_max\_speed = @(v) mass .\* acceleration;

d\_engine\_power = @(v) ((alpha ./ K) - (2 .\* beta .\* v ./ K .^ 2));

d = @(v) d\_power\_to\_overcome\_drag(v) + d\_gravity\_component(v) + d\_power\_to\_overcome\_rr(v) ...

+ d\_max\_speed(v) - d\_engine\_power(v);

%initilizing variabels for Newtons Method

N = 20; %max number of iterations

x = 21; %set up initial guess

tol = 1\*exp(-4);

%initilizing arrays for Newtons Method

roots\_array = zeros();

error\_array = zeros();

change\_array = zeros();

roots\_array(1) = x;

error\_array(1) = abs(f(x));

%iterations of Newtons Method

for ii = 1: N

x = x - f(x)/d(x);

roots\_array(ii+1) = x;

error\_array(ii+1) = abs(f(x));

change\_array(ii) = abs(x - roots\_array(ii));

if error\_array(ii+1) < tol && change\_array(ii) < tol

break

end

end

%plot of approximate roots

figure(3)

plot(0:ii, roots\_array)

xlabel('Number of iterations')

xticks(0: length(roots\_array))

ylabel('velocity (m/s)')

grid on

%plot of power

figure(4)

plot(0:ii, f(roots\_array))

xlabel('Number of iterations')

ylabel('f(v) in Watts')

xticks(0: length(roots\_array))

ylim = 100;

grid on

roots\_array

f(roots\_array)