

```

% Spring 2024 AER E 351 Homework 06 Problem 1.)6.11.)
% Matthew Mehrstens
clear; clc; close all;

%% Given
m_L = 1000; % [kg]
deltav_T = 8.5; % [km/s]
deltav_T = deltav_T * 1000; % [m/s]
c = [3500,3800,4100]; % [m/s]
epsilon = [0.10,0.12,0.09]; % []
gravity = 9.80665; % [m/s^2]
t = 90; % [s]

options = optimset("Display","off");

syms m_1 m_2 m_3;

%% Equations
f = @(epsilon,Z) ...
    sum(log(1 - epsilon) + log(Z) - log(1 - epsilon .* Z)); % []
g = @(deltav_T,c,Z) deltav_T - sum(c .* log(Z)); % [m/s]
g_gravity = @(deltav_T,c,Z,gravity,t) ...
    g(deltav_T, c, Z) + gravity * t; % [m/s]

f_alt = @(m) sum(m); % [kg]
g_alt = @(deltav_T,c,epsilon,m,m_L) ...
    c(1) * log((sum(m) + m_L) ...
        / (epsilon(1) * m(1) + sum(m(2:3)) + m_L)) ...
    + c(2) * log((sum(m(2:3)) + m_L) ...
        / (epsilon(2) * m(2) + m(3) + m_L)) ...
    + c(3) * log((m(3) + m_L) / (epsilon(3) * m(3) + m_L)) ...
    - deltav_T; % [velocity]

g_gravity_alt = @(deltav_T,c,epsilon,m,m_L,gravity,t) ...
    g_alt(deltav_T,c,epsilon,m,m_L) - gravity * t; % [velocity]

m_fn = @(Z,epsilon,m_1,m_2,m_3,m_L) ...
    solve(Z(1) ...
        == (m_1 + m_2 + m_3 + m_L) / (epsilon(1) * m_1 + m_2 + m_3 + m_L), ...
        Z(2) == (m_2 + m_3 + m_L) / (epsilon(2) * m_2 + m_3 + m_L), ...
        Z(3) == (m_3 + m_L) / (epsilon(3) * m_3 + m_L));

%% Calculations
Z = fmincon(@(Z) f(epsilon, Z),ones(1,3),[],[],[],[],[],[], ...
    @(X) rocket_constraint(X,@(Z) g(deltav_T,c,Z)),options); % []

m_0 = m_L * exp(f(epsilon,Z)); % [kg]

S_a = m_fn(Z,epsilon,m_1,m_2,m_3,m_L);
m_a = double([S_a.m_1,S_a.m_2,S_a.m_3]); % [kg]

m_a_alt = fmincon(@(m) f_alt(m),ones(1,3) * 1000,[],[],[],[],[],[], ...
    @(X) rocket_constraint(X,@(m) g_alt(deltav_T,c,epsilon,m,m_L)), ...
    options); % [kg]

m_0_alt = sum(m_a_alt) + m_L; % [kg]

Z_gravity = fmincon(@(Z) f(epsilon,Z),ones(1,3),[],[],[],[],[],[], ...

```

```

    @(X) rocket_constraint(X,@(Z) g_gravity(deltav_T,c,X,gravity,t)), ...
    options); % []

m_0_gravity = m_L * exp(f(epsilon, Z_gravity)); % [kg]

S_b = m_fn(Z_gravity,epsilon,m_1,m_2,m_3,m_L);
m_b = double([S_b.m_1,S_b.m_2,S_b.m_3]); % [kg]

m_b_alt = fmincon(@(m) f_alt(m),ones(1,3) * 1500,[],[],[],[],[],[], ...
    @(X) rocket_constraint(X, ...
    @(m) g_gravity_alt(deltav_T,c,epsilon,m,m_L,gravity,t)), ...
    options); % [kg]

m_0_gravity_alt = sum(m_b_alt) + m_L; % [kg]

%% Output
fprintf( ...
    "Problem 1.)6.11.)\n" + ...
    "a.) optimizing for Z_1, Z_2, and Z_3\n" + ...
    "m_0 = %g kg\n" + ...
    "m_1 = %g kg\n" + ...
    "m_2 = %g kg\n" + ...
    "m_3 = %g kg\n\n" + ...
    "a.) optimizing for m_1, m_2, and m_3\n" + ...
    "m_0 = %g kg\n" + ...
    "m_1 = %g kg\n" + ...
    "m_2 = %g kg\n" + ...
    "m_3 = %g kg\n\n" + ...
    "b.) optimizing for Z_1, Z_2, and Z_3\n" + ...
    "m_0 = %g kg\n" + ...
    "m_1 = %g kg\n" + ...
    "m_2 = %g kg\n" + ...
    "m_3 = %g kg\n\n" + ...
    "b.) optimizing for m_1, m_2, and m_3\n" + ...
    "m_0 = %g kg\n" + ...
    "m_1 = %g kg\n" + ...
    "m_2 = %g kg\n" + ...
    "m_3 = %g kg\n", ...
    m_0,m_a,m_0_alt,m_a_alt,m_0_gravity,m_b,m_0_gravity_alt,m_b_alt);

```