Table of Contents

AERO 535 - Assignment 1

Carter Briggs, Cole Helsel, Hunter Sagerer

```
close all;  clear;  clc;
```

Direct Ascent Calculations

Files Included:

```
function [m0DA] = directAscentPayload()
% Function for finding the payload mass in a direct ascent trajectory

mf = 11900; % CSM dry mass
deltaV = (0.9 + 2.2 + 2.5 + 0.9)*10^3; % deltaV of CSM, DA
Isp = 311; % ISP of Aerozine-50
g0 = 9.81;

m0DA = mf/exp(-deltaV/(Isp*g0));

% fprintf('Payload Mass of Direct Ascent: %0.0f kg\n',m0DA);

end
```

```
m0DA = directAscentPayload();
fprintf('LV Payload Mass of Direct Ascent: %0.0f kg
\n',round(m0DA,4,'significant'));
```

Lunar Rendezvous Calculations

Files Included:

```
function [m0LOR] = LORPayload()
% Function for finding the payload mass in a direct ascent trajectory

Isp = 311;
g0 = 9.81;

mf1 = 11900;
deltaV1 = 0.9*10^3;
```

```
mf2 = 4280;
deltaV2 = (0.1 + 2.2 + 2.5)*10^3;

deltaV3 = 0.9*10^3;

m0LOR1 = mf1/exp(-deltaV1/(Isp*g0));
m0LOR2 = (mf2)/exp(-deltaV2/(Isp*g0));
m0LOR3 = (m0LOR2 + m0LOR1)/exp(-deltaV3/(Isp*g0));

% fprintf('Payload Mass of LOR1: %0.3f kg\n',m0LOR1);
% fprintf('Payload Mass of LOR2: %0.3f kg\n',m0LOR2);
% fprintf('Payload Mass of LOR3: %0.0f kg\n',m0LOR3);

m0LOR = m0LOR3;

end


m0LOR = LORPayload();
fprintf('LV Payload Mass of LOR: %0.0f kg\n\n',round(m0LOR,4,'significant'));
```

Stage Masses

Files Included:

```
function [stageMasses,deltaVs] = stageMasses(mL)
% Function to calculate the mass per stage of the
% direct ascent launch vehicle

syms a

g0 = 9.81;
deltaV = 10500;

Isp = [283,311,421];
Epsilon = [0.05,0.07,0.19];

for i = 1:3
    R(i) = (a.*Isp(i).*g0 + 1)./(a.*Isp(i).*g0.*Epsilon(i));
    eqn(i) = Isp(i).*g0.*log(R(i));
end

EQN = eqn(1) + eqn(2) + eqn(3) == deltaV;

alpha = double(vpasolve(EQN,a));

R = (alpha.*Isp.*g0 + 1)./(alpha.*Isp.*g0.*Epsilon);
G = (1-R.*Epsilon)./(R-1);

m03 = mL*((1+G(3))/G(3));
m02 = m03*((1+G(2))/G(2));
m01 = m02*((1+G(1))/G(1));
```

```
stageMasses = [m01-m02,m02-m03,m03-mL];

dv1 = Isp(1)*g0*log((alpha*Isp(1)*g0 + 1)/(alpha*Isp(1)*g0*Epsilon(1)));
dv2 = Isp(2)*g0*log((alpha*Isp(2)*g0 + 1)/(alpha*Isp(2)*g0*Epsilon(2)));
dv3 = Isp(3)*g0*log((alpha*Isp(3)*g0 + 1)/(alpha*Isp(3)*g0*Epsilon(3)));

deltaVs = [dv1,dv2,dv3];
totaldV = sum(deltaVs);

end


[mDA, dvDA] = stageMasses(m0DA);
[mLOR, dvLOR] = stageMasses(m0LOR);

for i = 1:3
    fprintf('Direct Ascent Stage %d Mass: %0.3f x 10^6 kg
\n',i,round(mDA(i)/1E6,4,'significant'));
end
fprintf('Direct Ascent Total Mass: %0.3f x 10^6 kg\n
\n',round((sum(mDA)+m0DA)/1E6,4,'significant'));

for i = 1:3
    fprintf('LOR Stage %d Mass: %0.3f x 10^6 kg
\n',i,round(mLOR(i)/1E6,4,'significant'));
end
fprintf('LOR Total Mass: %0.3f x 10^6 kg\n
\n',round((sum(mLOR)+m0LOR)/1E6,4,'significant'));
```

Rocket Sizing

Files Included:

```
function [T, mdot, tB, n, D] = rocketSizing(m0, mStages, dv)

% Constants
ThrustRP1 = 6770e3;
ThrustLH2VA = 1033e3;
engThrust = [ThrustRP1,ThrustLH2VA, ThrustLH2VA];
Isp = [263,421,421];
TW(1) = 1.2;
TW(2) = 0.7;
TW(3) = 0.5;
g0 = 9.81;
engDiam = [3.1,2.7,2.7];
structCoef = [0.05,0.07,0.19];

% Rocket Sizing of the Direct Ascent Configuration
m(1) = mStages(1) + mStages(2) + mStages(3) + m0;
m(2) = mStages(2) + mStages(3) + m0;
m(3) = mStages(3) + m0;
```

```
% Ignition Thrusts
T = m.*g0.*TW;

% Thrust to Weight to Determine Engine # DA
n(1) = ceil((T(1)/ThrustRP1));
n(2) = ceil((T(2)/ThrustLH2VA));
n(3) = ceil((T(3)/ThrustLH2VA));

% Update thrust to be the actual value from number of engines
T = n.*engThrust;

% Flow Rate
mdot = T./(Isp.*g0);

% Propellant Mass from structural coefficient
% mp = m.*(1-exp(-dv./(g0.*Isp)));
ms = mStages.*structCoef;
mp = mStages-ms;

% Burn time
tB = mp./mdot;

% Engine diameter based on engine number
for i = 1:3
    switch n(i)
        case 1
            D(i) = 1*engDiam(i);
        case 2
            D(i) = 2*engDiam(i);
        case 5
            D(i) = (1+sqrt(2*(1+(1/sqrt(5)))))*engDiam(i);
        case 10
            D(i) = 3.813*engDiam(i);
    end
end

end


[TDA, mdotDA, tbDA, nDA, DDA] = rocketSizing(m0DA, mDA, dvDA);
[TLOR, mdotLOR, tbLOR, nLOR, DLOR] = rocketSizing(m0LOR, mLOR, dvLOR);

fprintf('Arc\tStg\tThrust [kN]\tFlow Rt [kg/s]\t tB [s] \tn\tdiam [m]\n');
for i = 1:3
    fprintf('DA \t %d \t %d \t\t %0.2f \t\t %0.2f \t%d\t%0.2f
\n',i,TDA(i)/1000,mdotDA(i), tbDA(i),nDA(i),DDA(i));
    fprintf('LOR\t %d \t %d \t\t %0.2f \t\t %0.2f \t%d\t%0.2f
\n',i,TLOR(i)/1000,mdotLOR(i), tbLOR(i),nLOR(i),DLOR(i));
end
```

Flight Simulation

Files Included:

```
% Based off of LaunchODE.m provided on Canvas

% verticalLaunch and fixedPitch are boolean values, where vertical launch
% does a gravity turn and fixedPitch does a fixed pitch maneuver
function [xdot] = saturnVODE(t,x,mdot,T_SL,T_vac,targetPitch)

% Set constants
Re = 6371e3;    % m, Radius of earth
g0=9.81;        % m/s^2, gravitational acceleration

% Split up x vector for easy interpretation
m = x(1);   % kg, masss
v = x(2);   % m/s, velocity
r = x(3);   % m, distance from center of Earth
psi = x(4); % rad, flight angle
phi = x(5); % rad, true anomaly

% Atmospheric Pressure Interpolation from Appendix
h_ref = 1e3*[0:9 10:5:25 30:10:80];
P_ref = 1e4*[10.13 8.988 7.95 7.012 6.166 5.405 4.722 4.111 3.565 3.08 2.65...
    1.211 0.5529 0.2549 0.1197 0.0287 0.007978 0.002196 0.00052 0.00011];
Patm = interp1(h_ref,P_ref,r-Re,'linear',0);

% Calculate thrust at new altitude
Thrust = T_vac - Patm/P_ref(1)*(T_vac-T_SL);

% Assumptions made
D = 0;      % Drag
L = 0;      % Lift
AoA = 0;    % Angle of Attack

% Change in mass
dmdt = -mdot;

% Change in velocity
if ~isnan(targetPitch)
    dvdt = -g0*sin(psi) + (Thrust/m)*cos(targetPitch-psi) - (D/m);
else
    dvdt = -g0*sin(psi) + (Thrust/m) - (D/m);
end

% Change in position
drdt = v*sin(psi);

% Change in Flight Angle
% If targetPitch is not NaN, in a constant pitch section
if ~isnan(targetPitch)
    dpsidt = -(g0/v - v/r)*cos(psi)+Thrust/(v*m)*sin(targetPitch-psi);
else
    dpsidt = -(g0/v - v/r)*cos(psi);
end

if isnan(dpsidt)
```

```
    dpsidt = 0;
end

% Change in True Anomoly
dphidt = v/r*cos(psi);

% Construct xdot
xdot = [dmdt;dvdt;drdt;dpsidt;dphidt];


end


options = odeset('AbsTol',1e-10,'RelTol',1e-12);

% Constants
Re = 6371e3;    % m, Radius of earth
g0 = 9.81;      % m/s^2, gravitational acceleration
T_F1_SL = 6770e3;
T_F1_Vac = 7770e3;
T_J2_SL = 486.2e3;
T_J2_Vac = 1033e3;

% Vertical Lift-off
x0 = [sum(mLOR)+m0LOR;
      0;
      Re;
      pi/2;
      0];
tstart = 0;
tstep = 0.01;
tend = 12;
timeS1 = tstart:tstep:tend;
[T1,X1] = ode45(@(t,x)
 saturnVODE(t,x,mdotLOR(1),5*T_F1_SL,5*T_F1_Vac,pi/2),...
   timeS1,x0,options);
% [T1,X1] = ode45(@(t,x) launchODE(t,x,mdotLOR(1),5*T_F1_SL,5*T_F1_Vac...
%     ,0,true,false,0),timeS1,x0,options);

% Pitch Over, Gravity Turn
x0 = X1(end,:);
x0(4) = deg2rad(89);
tstart = tend;
tend = tbLOR(1);
timeS2 = tstart:tstep:tend;
[T2,X2] = ode45(@(t,x) saturnVODE(t,x,mdotLOR(1),5*T_F1_SL,5*T_F1_Vac,NaN),...
   timeS2,x0,options);
% [T2,X2] = ode45(@(t,x) launchODE(t,x,mdotLOR(1),5*T_F1_SL,5*T_F1_Vac...
%     ,0,false,false,0),timeS2,x0,options);

% 2nd stage constant pitch
cp = 17.99335;    % Constant pitch angle, degrees
x0 = X2(end,:);
x0(1) = mLOR(2) + mLOR(3) + m0LOR;
```

```
tstart = tend;
tend = tbLOR(2)+tend;
timeS3 = tstart:tstep:tend;
[T3,X3] = ode45(@(t,x) saturnVODE(t,x,mdotLOR(2),5*T_J2_SL,5*T_J2_Vac,...
    deg2rad(cp)),timeS3,x0,options);
% [T3,X3] = ode45(@(t,x) launchODE(t,x,mdotLOR(2),5*T_J2_SL,5*T_J2_Vac...
%     ,0,false,true,deg2rad(cp)),timeS3,x0,options);

% 3rd stage constant pitch
x0 = X3(end,:);
x0(1) = mLOR(3) + m0LOR;
tstart = tend;
tend = tbLOR(3)+tend;
timeS4 = tstart:tstep:tend;
[T4,X4] = ode45(@(t,x) saturnVODE(t,x,mdotLOR(3),T_J2_SL,T_J2_Vac,...
    deg2rad(cp)),timeS4,x0,options);
% [T4,X4] = ode45(@(t,x) launchODE(t,x,mdotLOR(3),T_J2_SL,T_J2_Vac...
%     ,0,false,true,deg2rad(cp)),timeS4,x0,options);
% Correct to end at 7300 km/s
T4 = T4(X4(:,2)<=7300);
X4 = X4(X4(:,2)<=7300,:);
tend = max(T4);
timeS4 = tstart:tstep:tend;

% % Coast
% x0 = X4(end,:);
% tstart = tend;
% tend = tend + 500;
% timeS5 = tstart:tstep:tend;
% [T5,X5] = ode45(@(t,x) saturnVODE(t,x,0,0,0,0),timeS5,x0,options);

% Combine
T = [T1;T2;T3;T4];
X = [X1;X2;X3;X4];

% Post process
mass = X(:,1);
vel = X(:,2);
r = X(:,3);
psi = X(:,4);
phi = X(:,5);
alt = r-Re;
range = Re.*phi;
fprintf('\n\nFinal Vel: %0.2f km/s\n',vel(end)/1000);
fprintf('Final Alt: %0.3f km\n',alt(end)/1000);

% Acceleration Calculation
% acc = (diff(vel))/tstep;
% acc(end+1) = 0;
r_d = vel.*sin(psi);
r_dd = (diff(r_d))/tstep;
r_dd(end+1) = 0;
phi_d = vel./r.*cos(psi);
phi_dd = (diff(phi_d))/tstep;
```

```
phi_dd(end+1) = 0;
acc = sqrt((r_dd-r.*phi_d.^2).^2+(r.*phi_dd+2.*r_d.*phi_d).^2);
[~,i] = max(acc);
acc(i) = acc(i-1);

% Time and Axis Calcs
tpitch = max(timeS1);
tmeco = max(timeS2);
tseco = max(timeS3);
rpitch = range(T==tpitch);
rpitch = rpitch(1)/1000;
rmeco = range(T==tmeco)/1000;
rseco = range(T==tseco)/1000;

% Plot altitude vs. time
figure();
plot(T,alt/1000,'-b','linewidth',2);
hold on;
plot([tpitch,tpitch],[0,1.2*max(alt/1000)],'--k','linewidth',2);
plot([tmeco,tmeco],[0,1.2*max(alt/1000)],'--k','linewidth',2);
plot([tseco,tseco],[0,1.2*max(alt/1000)],'--k','linewidth',2);
% ht = text(5,0.05*max(alt/1000),'Vertical Liftoff','Fontname','times');
% set(ht,'rotation',90);
% text(17,0.15*max(alt/1000),'Start Gravity Turn','Fontname','times');
% text(30,0.8*max(alt/1000),'Stage 1 Gravity Turn','Fontname','times');
% text(140,0.15*max(alt/1000),'Stage 1 MECO','Fontname','times');
% text(140,0.1*max(alt/1000),'Stage 2 Ignition','Fontname','times');
% text(140,0.05*max(alt/1000),'Turn to Constant Pitch','Fontname','times');
% text(300,0.8*max(alt/1000),'Stage 2 Constant Pitch','Fontname','times');
% text(525,0.1*max(alt/1000),'Stage 2 MECO','Fontname','times');
% text(525,0.05*max(alt/1000),'Stage 3 Ignition','Fontname','times');
% text(540,0.8*max(alt/1000),'Stage 3 Constant Pitch','Fontname','times');
hold off;
xlabel('Time [s]');
ylabel('Altitude [km]');
xticks(0:60:max(T));
xlim([0,max(T)]);
ylim([0,1.1*max(alt/1000)]);
set(gca,'fontname','times');

set(gcf, 'Position', get(0, 'Screensize').*0.5 + [200 200 0 0]);
saveas(gcf, [pwd '/Figures/1.png'])

% Plot altitude vs. range
figure();
plot(range/1000,alt/1000,'-b','linewidth',2);
hold on;
plot([rpitch,rpitch],[0,1.2*max(alt/1000)],'--k','linewidth',2);
plot([rmeco,rmeco],[0,1.2*max(alt/1000)],'--k','linewidth',2);
plot([rseco,rseco],[0,1.2*max(alt/1000)],'--k','linewidth',2);
% text(50,0.8*max(alt/1000),'Stage 1','Fontname','times');
% text(400,0.8*max(alt/1000),'Stage 2','Fontname','times');
% text(1600,0.8*max(alt/1000),'Stage 3','Fontname','times');
hold off;
```

```
xlabel('Range [km]');
ylabel('Altitude [km]');
xlim([0,max(range./1000)]);
ylim([0,1.1*max(alt/1000)]);
set(gca,'fontname','times');

set(gcf, 'Position', get(0, 'Screensize').*0.5 + [200 200 0 0]);
saveas(gcf, [pwd '/Figures/2.png'])

% Plot acceleration vs. time
figure();
plot(T,acc/g0,'-b','linewidth',2);
hold on;
plot([tpitch,tpitch],[0,1.2*max(acc/g0)],'--k','linewidth',2);
plot([tmeco,tmeco],[0,1.2*max(acc/g0)],'--k','linewidth',2);
plot([tseco,tseco],[0,1.2*max(acc/g0)],'--k','linewidth',2);
% text(30,0.8*max(acc/g0),'Stage 1','Fontname','times');
% text(300,0.8*max(acc/g0),'Stage 2','Fontname','times');
% text(540,0.8*max(acc/g0),'Stage 3','Fontname','times');
hold off;
xlabel('Time [s]');
ylabel('Acceleration [Gs]');
xticks(0:60:max(T));
xlim([0,max(T)]);
ylim([0,1.1*max(acc/g0)]);
set(gca,'fontname','times');

set(gcf, 'Position', get(0, 'Screensize').*0.5 + [200 200 0 0]);
saveas(gcf, [pwd '/Figures/3.png'])

% Print Con Ops
fprintf('\nConcept of Operations\n');
fprintf('Mission Time\tEvent\n');
fprintf('%0.2f s\t\t\tLiftoff\n',min(timeS1));
fprintf('%0.2f s\t\t\tPitch Maneuver\n',min(timeS2));
fprintf('%0.2f s\t\tMECO\n',max(timeS2));
fprintf('%0.2f s\t\t2nd Stage Ignition\n',min(timeS3));
fprintf('%0.2f s\t\tPitch to %0.2f deg\n',min(timeS3),cp);
fprintf('%0.2f s\t\tSECO\n',max(timeS3));
fprintf('%0.2f s\t\t3rd Stage Ignition\n',min(timeS4));
fprintf('%0.2f s\t\t3rd Stage Cutoff\n',max(timeS4));
```

Published with MATLAB® R2022a