## Table of Contents

# Joshua Oates - AERO 215 - Fall 2021 - space wrap up

```
close all;
clear all;
clc;

u = 3.986004418 * (10^5); %km^3/s^2, (mu) earth

%for part 3
isp = 316; % isp of C208 in s
g = 9.81; % m/s^2
mf = 12568;  % final mass of C208 in kg

% for propellent: 1000 kg per cubic meter
% 1m/1000kg
propConv = 1/1000;
```

# Part 1

Use the following values to generate COEs for the Cargo Dragon-2 spacecraft C208 and the ISS.

```
% Cargo Dragon-2 C208 R= -406.663, -4186.877, -5059.146 (km)
% Cargo Dragon-2 C208 V = 7.386, -2.178, 1.1889 (km/sec)
% ISS R = 5648.682, -2337.321, 2943.766 (km)
% ISS V = -0.208, 5.799, 5.008 (km/sec)


% Prepare a COEs struct to be calculated
CD.name = "Cargo Dragon 2 C208";
CD.Rv = [-406.663, -4186.877, -5059.146]; %km
CD.Vv = [7.386, -2.178, 1.1889]; %km/sec
ISS.name = "International Space Station";
ISS.Rv = [5648.682, -2337.321, 2943.766]; %km
ISS.Vv = [-0.208, 5.799, 5.008]; %km/sec

% Calculate COEs
[CD.a,CD.e,CD.nu,CD.i,CD.raan,CD.aop,CD.T,CD.E] =
 COEsOatesJoshua(CD.Rv,CD.Vv);
[ISS.a,ISS.e,ISS.nu,ISS.i,ISS.raan,ISS.aop,ISS.T,ISS.E] =
 COEsOatesJoshua(ISS.Rv,ISS.Vv);
```

```
% Calculate Ra = Radius at apoapsis as magnitude in km
CD.Ra = CD.a * ( 1 + CD.e);
ISS.Ra = ISS.a * ( 1 + ISS.e);

% Calculate Rp = Radius at periapsis as magnitude in km
% Rp + Ra = 2 * a
ISS.Rp = (2 * ISS.a) - ISS.Ra;
CD.Rp = (2 * CD.a) - CD.Ra;

% Outputs:

disp("Part 1: COEs, apogee, and perigee for ISS and C208");
disp(" ");

dispCOEsOatesJoshua(CD,1,1);
dispCOEsOatesJoshua(ISS,1,1);

disp(CD.name + " Radius of");
disp ("  -apogee:  " + CD.Ra + " km");
disp ("  -perigee: " + CD.Rp + " km");
disp(" ");

disp(ISS.name + " Radius of");
disp ("  -apogee:  " + ISS.Ra + " km");
disp ("  -perigee: " + ISS.Rp + " km");

% Analysis:

% The perigee radius of C208 is 6574 km compared to ISS' 6783 km.
% This is a difference of 209 km.

% The apogee radius of C208 is 6613 km compared to ISS' 6787 km.
% This is a difference of 175 km.

%The closesest approach (C208 apogee to ISS perigee) is 171 km.

% While both orbits are fairly circular, ISS is much more so.
% This means that C208 will be closest to ISS at its apogee and is 38
 km
% closer than when it is at perigee.

% For this reason, it would make sense for the rendezvous to happen
 closer
% to the initial apogee for C208 since apogee is much closer to the
% destination orbit than perigee.

% The inclination for C208 is 51.6 degrees and so is the ISS's.
% This was done so that a plane change is not a necissary manuver to
% rendezvous. Avoiding this manuver will save the spacecraft fuel.
% The inclination was probably made very specifically for this reason
 at
% the time of the launch.
```

```
Part 1: COEs, apogee, and perigee for ISS and C208
```

```
------- Cargo Dragon 2 C208 -------
Semi-major axis       : 6593.4981 km
Eccentricity          : 0.002887
True anomaly          : 42.9263 degrees
Inclination           : 51.6477 degrees
RAAN                  : 336.5533 degrees
Argument of periapsis : 238.4059 degrees
Period                : 5328.2527 seconds
Specific Mech. Energy : -30.2268 km^2/s^2

------- International Space Station -------
Semi-major axis       : 6785.5425 km
Eccentricity          : 0.00026781
True anomaly          : 73.1592 degrees
Inclination           : 51.647 degrees
RAAN                  : 315.1239 degrees
Argument of periapsis : 320.4313 degrees
Period                : 5562.7282 seconds
Specific Mech. Energy : -29.3713 km^2/s^2

Cargo Dragon 2 C208 Radius of
  -apogee:  6612.5336 km
  -perigee: 6574.4626 km

International Space Station Radius of
  -apogee:  6787.3597 km
  -perigee: 6783.7252 km
```

# Part 2

Calculate dV to circularize CD at apogee

```
% Calculate V at Ra for CD
CD.VRa = sqrt( ( CD.E + ( u/CD.Ra) ) *2 );

% Find E for an orbit where Ra = COEs1.Ra and e = 0. ie circularize
CD_C.Ra = CD.Ra; % C208's orbit once circular

% Find E of circular
CD_C.E = (-u) / ( 2* CD_C.Ra );

% Use E to find Vm ( magnitude V ) for circular orbit ( Orbit C )
CD_C.VRa = sqrt( ( CD_C.E + ( u/CD_C.Ra) ) *2 );

% Use V of orbit 1 and V of orbit C to find dV of circularizing burn
dV = abs( CD_C.VRa - norm(CD.VRa) ) * 1000; %m/s

% This test funtion sets up an orbit that has perpendicular R and V
 vectors
% and can be used to show that dV is correct, by calculating that e is
 near
% 0.
% [a,e,nu,i,raan,aop,T,E] = COEsOatesJoshua([CD.Ra,0,0],
[0,0,norm(CD.VRa)+dV]);

% Output:


disp("____________________________________________________________________");
disp(" ");
disp("Part 2: dV for circularization");
disp(" ");
disp("For circularization, " + CD.name + " will need a delta V of: " +
 dV +" m/s");
```

```
____________________________________________________________________

Part 2: dV for circularization

For circularization, Cargo Dragon 2 C208 will need a delta V of:
 11.2155 m/s
```

# Part 3

find the amount of fuel used by C208 during circularization

```
% use:
% dV = isp * g * ln ( mi / mf )
% dV / ( isp * g ) = ln ( mi / mf )
% exp ( dV / ( isp * g ) ) = mi / mf
% mf * exp ( dV / ( isp * g ) ) = mi

mi = mf * exp ( dV / ( isp * g ) );
dm = mf - mi;

% this function can check that the mass use is correct by showing dV2
 = dV
% dV2 = isp * g * log ( mi / mf );

% Output:

disp("____________________________________________________________________");
disp(" ");
disp("Part 3: amount of propellent for circularization");
disp(" ");
disp("For circularization, " + CD.name + " will need a propellent mass
 of: " + abs(dm) +" kg");
disp("For circularization, " + CD.name + " will need a propellent
 volume of: " + abs(dm) * propConv +" m^3");

% propellent volume of 0.045553 m^3 is approximately 1.6 ft^3 for only
 11.2 m/s dV
```

```
____________________________________________________________________

Part 3: amount of propellent for circularization
```

```
For circularization, Cargo Dragon 2 C208 will need a propellent mass
 of: 45.5526 kg
For circularization, Cargo Dragon 2 C208 will need a propellent volume
 of: 0.045553 m^3
```

*Published with MATLAB® R2021a*