Homework 6 Code

Cameron Lane

CJL3282

QUESTION 2:

```
clear
clc
radiusearth = 6378.1363;
muearth = 398600.4415;
j2 = 0.0010826267;
j3 = -0.0000025327;
omegaearth = 7.2921158553 * 10^(-5);
rtestvector = [6092.032, 2487.062, 2487.062];
aj2test = a_j2(rtestvector, muearth, j2, radiusearth);
aj3test = a_j3(rtestvector, muearth, j3, radiusearth);
aptest = apert(rtestvector, muearth, j2, j3, radiusearth)
rvector = [100, 5000, -6000];
aj2 = a_j2(rvector, muearth, j2, radiusearth);
aj3 = a_j3(rvector, muearth, j3, radiusearth);
ap = apert(rvector, muearth, j2, j3, radiusearth)
```

QUESTION 3:

```
clear
clc
radiusearth = 6378.1363;
mueearth = 398600.4415;
j2 = 0.0010826267;
j3 = -0.0000025327;
omegaearth = 7.2921158553 * 10^(-5);
rvec = [2781.000 5318.000 -5629.000];
rsunvec = [2379260.000 148079334.000 -1009936.000];
am = 0.0001;
cr = 1.5;
gamma = 1;
rscsvec = rsunvec - rvec;
asrp = asrp(rscsvec, gamma, cr, am )
```

QUSTION 4:

```
clear
cl
altvector = 1:1:1000;
for j = 1:1:1000
    rho(j) = getDensityParams(j);
end
figure(1)
semilogx(rho, altvector)
ylabel('Altitude in km')
xlabel('Density in kg/km3')
title('Atmospheric Density vs. Altitude')
```

QUESTION 5:

```
clear
clc


radiusearth = 6378.1363;
muearth = 398600.4415;
musun = 1.327e11;
j2 = 0.0010826267;
j3 = -0.0000025327;
omegaearth = 7.2921158553 * 10^(-5);

a = 6800;
e = 0.005;
i = 71;
i = dR( i );
Omega = 300;
Omega = dR( Omega );
omega = 78;
omega = dR( omega );
nu = 0;

startTime = 2458200.5;
startingSecond = startTime * 86400;
endTime = startTime+1;
totalSeconds = 86400;
endSecond = endTime*86400;
time = 0:60:86400;

elements = [a e i Omega omega nu];
```

```
rv = OrbitalElementToCartesian(elements, muearth);
rvmatrix = [rv(1) rv(2) rv(3); rv(4) rv(5) rv(6)];


for it = 1:1:1000
    inf = getDensityParams(it);
    h0 = inf(2);
    rho0 = inf(1);
    H = inf(3);
end

rho0 = rho0 / (10^(-9));

[T, Y] = prop45AllForces(rv, time, muearth, musun, j2, j3, radiusearth, h0, rho0, H);

[T2,Y2] = propCartODE45(rv, time, muearth);

Y(1441, :)

difference = Y - Y2;

time = time';
time = time / 60  / 60 ;
figure(1)
subplot(3,1,1)
plot(time, difference(:, 1))
ylim([-1010 1010])
xlabel('Time in hours')
ylabel('Distance in km')
title('Difference in 2 Body and High Fidelity Propogator in i-direction')
subplot(3,1,2)
plot(time, difference(:, 2))
xlabel('Hours')
ylabel('km')
title('j-direction')
subplot(3,1,3)
plot(time, difference(:, 3))
xlabel('Hours')
ylabel('km')
title('k-direction')
```

FUNCTIONS:

```
function [ rGCRF ] = j2000GCRF( R_J2000 )

deltaAlphaNot = 0.0146;
xiNot = -0.16617;
etaNot = -0.0068192;
deltaAlphaNot = deltaAlphaNot * pi() / 648000;
xiNot = xiNot * pi() / 648000;
etaNot = etaNot * pi() / 648000;

B_1 = R3(-deltaAlphaNot);
B_2 = R2(-xiNot);
B_3 = R1(etaNot);

B = B_1 * B_2 * B_3;

rGCRF = B * R_J2000';


end


function [ T, Y ] = prop45AllForces( rv, time, muEarth, muSun, j2, j3, re, h_not, rho_not, H)
odeoptions = odeset('RelTol',1e-10,'AbsTol',1e-20);
[T,Y] = ode45( @(t,y) dy4(t, y, muEarth, muSun, j2, j3, re, h_not, rho_not, H), time, rv,
odeoptions);
end

function [ RGCRF ] = earthrel( JD )

%JD = 2451545.0; %TDB
TJD = Ttt( JD );
muEarth = 398600.4415;
muSun = 1.327e11;


a = 1.000001018;
e = 0.01670862 - 0.000042037*TJD - 0.0000001236*TJD^2 + 0.00000000004*TJD^3;
i = 0.00000000 + 0.0130546  *TJD - 0.00000931  *TJD^2 - 0.000000034  *TJD^3;
Omega= 174.873174 - 0.2410908*TJD + 0.00004067  *TJD^2 - 0.000001327  *TJD^3;
omegatilde=102.937348+0.3225557*TJD + 0.00015026  *TJD^2 + 0.000000478  *TJD^3;
lambdaM=100.466449+35999.3728519*TJD- 0.00000568*TJD^2 + 0.000000000  *TJD^3;

a = aukm(a);
i = dR(i);
Omega = dR(Omega);
```

```
omegatilde = dR(omegatilde);
lambdaM = dR(lambdaM);
M = lambdaM - omegatilde;
omega = omegatilde - Omega;

EccentricA = AnomalyPropogation(M, e);
nu = 2*atan2( sqrt(1 + e) * tan(EccentricA / 2), sqrt(1-e));
elements = [a e i Omega omega nu];

obliquityE = 23.439279 - 0.0130102*TJD - 5.086e-8*TJD^2 +5.565e-7*TJD^3 + 1.6e-10*TJD^4 + 1.21e-11*TJD^5;
obliquityE = dR(obliquityE); %convert it to radians

RV5 = OrbitalElementToCartesian(elements, muSun);
RV5 = [RV5(1) RV5(2) RV5(3); RV5(4) RV5(5) RV5(6)];
RV = R1(-obliquityE) * RV5';
RV = RV';
R = RV(1,1:3);

%Convert from J2000 frame to GCRF

RGCRF  = j2000GCRF( R );

end

function [out] = getDensityParams( altitude )
%
% function [rho0,h0,H] = getDensityParams( altitude )
% ------------------------------------------------------------------
%
% Description:
%
%  Get the density model parameters for the 1976 Standard Atmosphere
%  exponential model
%
% Inputs:
%
%  altitude - Altitude above the Earth's surface in kilometers
%
% Outputs:
%
%  rho0 - Nominal density in kg/m^3
%  h0   - Base altitude (not radius!) in km
%  H    - Scale height in km
%
% Assumptions/References:
```

```
%
%  Vallado and McClain, Third Edition, P. 564
%
% Dependencies:
%
%  None
%
% Modification History:
%
%   18jan17     Brandon A. Jones      original version
%
% ---------------------------------------------------------------------
% Copyright University of Texas at Austin, 2017
%
if altitude > 1000
    rho0 = 3.019e-15;
    h0 = 1000;
    H = 268;
    out = [rho0,h0,H];
elseif altitude > 900
    rho0 = 5.245e-15;
    h0 = 900;
    H = 181.05;
    out = [rho0,h0,H];
elseif altitude > 800
    rho0 = 1.170e-14;
    h0 = 800;
    H = 124.64;
    out = [rho0,h0,H];
elseif altitude > 700
    rho0 = 3.614e-14;
    h0 = 700;
    H = 88.667;
    out = [rho0,h0,H];
elseif altitude > 600
    rho0 = 1.454e-13;
    h0 = 600;
    H = 71.835;
    out = [rho0,h0,H];
elseif altitude > 500
    rho0 = 6.967e-13;
    h0 = 500;
    H = 63.822;
    out = [rho0,h0,H];
elseif altitude > 450
    rho0 = 1.585e-12;
    h0 = 450;
    H = 60.828;
    out = [rho0,h0,H];
elseif altitude > 400
    rho0 = 3.725e-12;
    h0 = 400;
    H = 58.515;
    out = [rho0,h0,H];
elseif altitude > 350
    rho0 = 9.518e-12;
    h0 = 350;
    H = 53.298;
    out = [rho0,h0,H];
elseif altitude > 300
    rho0 = 2.418e-11;
    h0 = 300;
    H = 53.628;
    out = [rho0,h0,H];
elseif altitude > 250
    rho0 = 7.248e-11;
    h0 = 250;
    H = 45.546;
    out = [rho0,h0,H];
elseif altitude > 200
    rho0 = 2.789e-10;
    h0 = 200;
    H = 37.105;
    out = [rho0,h0,H];
elseif altitude > 180
    rho0 = 5.464e-10;
    h0 = 180;
    H = 29.740;
    out = [rho0,h0,H];
elseif altitude > 150
    rho0 = 2.070e-9;
    h0 = 150;
    H = 22.523;
    out = [rho0,h0,H];
elseif altitude > 140
    rho0 = 3.845e-9;
    h0 = 140;
    H = 16.149;
    out = [rho0,h0,H];
elseif altitude > 130
    rho0 = 8.484e-9;
    h0 = 130;
    H = 12.636;
    out = [rho0,h0,H];
elseif altitude > 120
    rho0 = 2.438e-8;
    h0 = 120;
    H = 9.473;
    out = [rho0,h0,H];
elseif altitude > 110
    rho0 = 9.661e-8;
    h0 = 110;
    H = 7.263;
    out = [rho0,h0,H];
elseif altitude > 100
    rho0 = 5.297e-7;
    h0 = 100;
    H = 5.877;
    out = [rho0,h0,H];
elseif altitude > 90
    rho0 = 3.396e-6;
    h0 = 90;
    H = 5.382;
    out = [rho0,h0,H];
elseif altitude > 80
    rho0 = 1.905e-5;
    h0 = 80;
    H = 5.799;
    out = [rho0,h0,H];
elseif altitude > 70
    rho0 = 8.770e-5;
    h0 = 70;
    H = 6.549;
    out = [rho0,h0,H];
elseif altitude > 60
    rho0 = 3.206e-4;
    h0 = 60;
    H = 7.714;
    out = [rho0,h0,H];
elseif altitude > 50
    rho0 = 1.057e-3;
    h0 = 50;
    H = 8.382;
    out = [rho0,h0,H];
elseif altitude > 40
    rho0 = 3.972e-3;
    h0 = 40;
    H = 7.554;
    out = [rho0,h0,H];
elseif altitude > 30
    rho0 = 1.774e-2;
    h0 = 30;
    H = 6.682;
    out = [rho0,h0,H];
elseif altitude > 25
    rho0 = 3.899e-2;
    h0 = 25;
    H = 6.349;
    out = [rho0,h0,H];
else
    rho0 = 1.225;
    h0 = 0;
    H = 7.249;
    out = [rho0,h0,H];
end
end
```