3BP Propagator

```
% Gravitational parameters:
muSun = 1.327124400189e11; % km^3/s^2
muEarth = 3.9860044188e5; % km^3/s^2
muMoon = 4.90486959e3; % km^3/s^2

% Initial states:
r0_Earth = [1.085654661170196E+07 -1.516246665530904E+08
 -9.740893969707191E+03]';
v0_Earth = [2.924778860543529E+01  2.018192849080797
 1.110912062926461E-03]';
X0_Earth = [r0_Earth; v0_Earth];
r0_Moon  = [1.059481977955440E+07 -1.513568853278044E+08
 -1.365246557713300E+04]';
v0_Moon  = [2.845478370962656E+01  1.335568444795943
 -9.403977943926289E-02]';
X0_Moon  = [r0_Moon; v0_Moon];
r0_Sun   = [3.891963415068314E+04  5.550923454368106E+04
 -6.493728399841038E+03]';
v0_Sun   = [8.758026516243883E-03  2.695060513075030E-03
 -1.589335583160812E-04]';
X0_Sun   = [r0_Sun; v0_Sun];

tspan = [0 1];

yEarth = X0_Earth';
yMoon = X0_Moon';
ySun = X0_Sun';
year = 31536000;
day = 60*60*24;
xEarth = zeros(6,1);
xEarth = X0_Earth;
xMoon = zeros(6,1);
xMoon = X0_Moon;
xSun = zeros(6,1);
xSun = X0_Sun;
earthOut = zeros(year, 6);
moonOut = zeros(year, 6);
sunOut = zeros(year, 6);
bigMu = muSun + muEarth + muMoon;

for n = 1:day:year

    for m = 1:6
        earthOut(n,m) = yEarth( end, m);
        moonOut(n,m) = yMoon( end, m);
        sunOut(n,m) = ySun( end, m);
    end

    [tEarth, yEarth] = ode45(@propagate_3BP, tspan, xEarth, [], xSun,
 muSun, xMoon, muMoon);
```

```
    [tMoon, yMoon] = ode45(@propagate_3BP, tspan, xMoon, [], xSun,
 muSun, xEarth, muEarth);
    [tSun, ySun] = ode45(@propagate_3BP, tspan, xSun, [], xEarth,
 muEarth, xMoon, muMoon);

    REarth = [yEarth(end,1) yEarth(end,2) yEarth(end,3)]';
    RMoon = [yMoon(end,1) yMoon(end,2) yMoon(end,3)]';
    RSun = [ySun(end,1) ySun(end,2), ySun(end,3)]';

    % Update:
    tspan = [n n+1];
    r_baryOld = (1/bigMu) * (muSun * xSun(1:3,1) + muEarth *
 xEarth(1:3,1) + muMoon * xMoon(1:3,1));
    r_baryNew = (1/bigMu) * (muSun * RSun + muEarth * REarth + muMoon
 * RMoon);
    vBary = (r_baryNew - r_baryOld)/day;
    rEarth_curr = REarth - r_baryNew;
    rMoon_curr = RMoon - r_baryNew;
    rSun_curr = RSun - r_baryNew;
    vEarth_curr = yEarth(end, 4:6)' - vBary;
    vMoon_curr = yMoon(end, 4:6)' - vBary;
    vSun_curr = ySun(end, 4:6)' - vBary;

    for o = 1:6
        if (o < 4)
            xEarth(o,1) = rEarth_curr(o,1);
            xMoon(o,1) = rMoon_curr(o, 1);
            xSun(o,1) = rSun_curr( o, 1);
        else
            xEarth(o,1) = vEarth_curr( o - 3, 1);
            xMoon(o,1) = vMoon_curr( o - 3, 1);
            xSun(o,1) = vSun_curr( o - 3, 1);
        end
    end

end

figure(1);
plot3(earthOut(:,1), earthOut(:,2), earthOut(:,3));
hold on;
plot3(moonOut(:,1), moonOut(:,2), moonOut(:,3));
hold on;
plot3(sunOut(:,1), sunOut(:,2), sunOut(:,3));
title({'3BP Propagator Solution', 'By: Tom West'});
```

FUNCTIONS:

```
function XDot = propagate_3BP(t, Xi, Xj, muj, Xk, muk)

% Compute distances between attracting bodies:
R_ij = [(Xj(1,1) - Xi(1,1)) (Xj(2,1) - Xi(2,1)) (Xj(3,1) - Xi(3,1))]';
r_ij = norm(R_ij);
R_ik = [(Xk(1,1) - Xi(1,1)) (Xk(2,1) - Xi(2,1)) (Xk(3,1) - Xi(3,1))]';
```

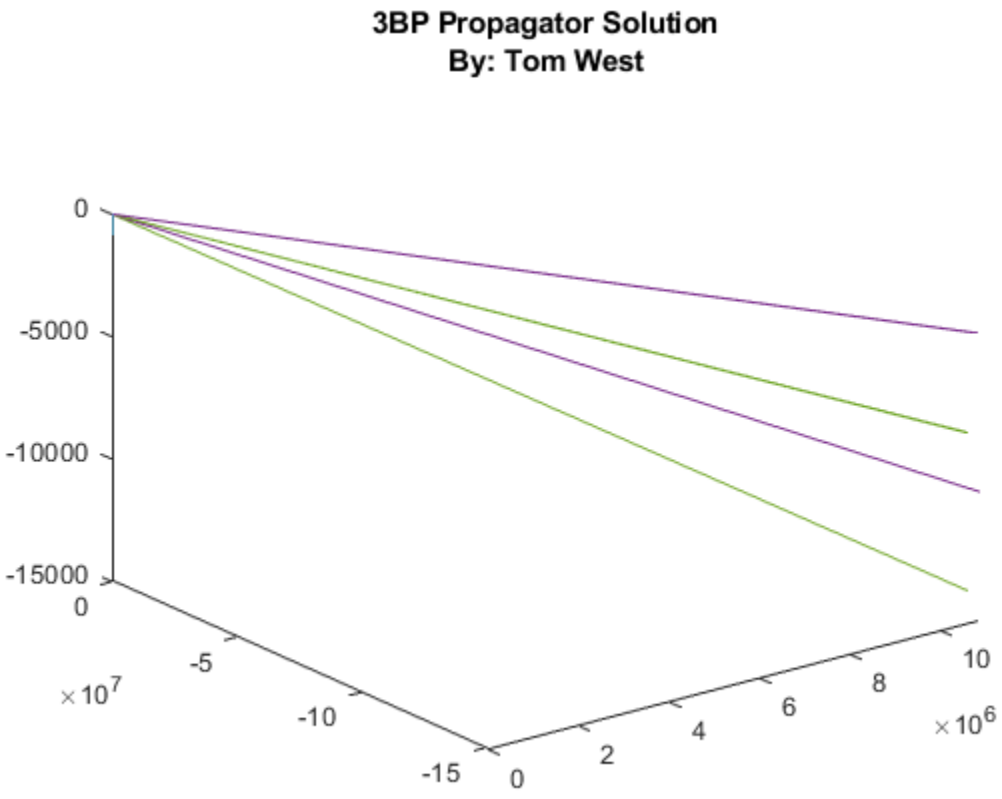
```
r_ik = norm(R_ik);

% Gravitational acceleration:

% Body j:
ax_ij = -(muj/(r_ij^3)) .* R_ij(1,1);
ay_ij = -(muj/(r_ij^3)) .* R_ij(2,1);
az_ij = -(muj/(r_ij^3)) .* R_ij(3,1);
% Body k:
ax_ik = -(muk/(r_ik^3)) .* R_ik(1,1);
ay_ik = -(muk/(r_ik^3)) .* R_ik(2,1);
az_ik = -(muk/(r_ik^3)) .* R_ik(3,1);
% Superposition:
ax = ax_ij + ax_ik;
ay = ay_ij + ay_ik;
az = az_ij + az_ik;

% State derivative
XDot = [Xi(4,1) Xi(5,1) Xi(6,1) ax ay az]';

end
```



Published with MATLAB® R2020b