```
function [ydot] = NumericJ2Prop(t,Y,mu, J2, Re)
% J2 IS 7th TERM IN STATE!!!!!
% the input Y is a column with the beginning being the states and the end
% being the STM
%--- propagate the state
ydot = zeros(6,1);

% velocity maps to itself
ydot(1:3,1) = Y(4:6,1);

% assign states
x = Y(1);
y = Y(2);
z = Y(3);

% compute r
r = sqrt(x^2 + y^2 + z^2);

% accerlation due to J2 perturbation
apertx = -((mu*x)/(r^3))*(1-J2*(3/2)*(Re/r)^2*(5*(z/r)^2-1));
aperty = -((mu*y)/(r^3))*(1-J2*(3/2)*(Re/r)^2*(5*(z/r)^2-1));
apertz = -((mu*z)/(r^3))*(1-J2*(3/2)*(Re/r)^2*(5*(z/r)^2-3));

% How the accelerations map to output
ydot(4) = apertx;
ydot(5) = aperty;
ydot(6) = apertz;

if length(Y) == 7
    % if including J2
    J2partial = 0;
    ydot(7) = J2partial;
else
    % do nothing
end

% If only wanting to propagate state then dont worry about this part
if length(Y) < 8
    % No worries, just propagating the state
else
    % going to propagate the STM too
    %--- construct A matrix to be evaluated for each new state

    % Initialize A with zeros to make easier to fill in
    A = zeros(7,7);

    % velocity states map to themselves
    A(1:3, 4:6) = eye(3,3);

    A(4,1) = -(2*mu*(x^2 + y^2 + z^2)^3 - 6*mu*x^2*(x^2 + y^2 + z^2)^2 + 3*J2*Re^2*mu*(x^2 + y^2 + z^2)^2 + 105*J2*Re^2*mu*x^2*z^2 - 15*J2*Re^2*mu*x^2*(x^2 + y^2 +

    A(5,1) = (6*mu*x*y*(x^2 + y^2 + z^2)^2 - 105*J2*Re^2*mu*x*y*z^2 + 15*J2*Re^2*mu*x*y*(x^2 + y^2 + z^2))/(2*(x^2 + y^2 + z^2)^(9/2));

    A(6,1) = (6*mu*x*z*(x^2 + y^2 + z^2)^2 - 105*J2*Re^2*mu*x*z^3 + 45*J2*Re^2*mu*x*z*(x^2 + y^2 + z^2))/(2*(x^2 + y^2 + z^2)^(9/2));

    A(4,2) = (6*mu*x*z*(x^2 + y^2 + z^2)^2 - 105*J2*Re^2*mu*x*z^3 + 45*J2*Re^2*mu*x*z*(x^2 + y^2 + z^2))/(2*(x^2 + y^2 + z^2)^(9/2));

    A(5,2) = -(2*mu*(x^2 + y^2 + z^2)^3 - 6*mu*y^2*(x^2 + y^2 + z^2)^2 + 3*J2*Re^2*mu*(x^2 + y^2 + z^2)^2 + 105*J2*Re^2*mu*y^2*z^2 - 15*J2*Re^2*mu*y^2*(x^2 + y^2 +

    A(6,2) = -(2*mu*(x^2 + y^2 + z^2)^3 - 6*mu*y^2*(x^2 + y^2 + z^2)^2 + 3*J2*Re^2*mu*(x^2 + y^2 + z^2)^2 + 105*J2*Re^2*mu*y^2*z^2 - 15*J2*Re^2*mu*y^2*(x^2 + y^2 +

    A(4,3) = (6*mu*x*z*(x^2 + y^2 + z^2)^2 - 105*J2*Re^2*mu*x*z^3 + 45*J2*Re^2*mu*x*z*(x^2 + y^2 + z^2))/(2*(x^2 + y^2 + z^2)^(9/2));

    A(5,3) = (6*mu*y*z*(x^2 + y^2 + z^2)^2 - 105*J2*Re^2*mu*y*z^3 + 45*J2*Re^2*mu*y*z*(x^2 + y^2 + z^2))/(2*(x^2 + y^2 + z^2)^(9/2));

    A(6,3) = -(2*mu*(x^2 + y^2 + z^2)^3 - 6*mu*z^2*(x^2 + y^2 + z^2)^2 + 9*J2*Re^2*mu*(x^2 + y^2 + z^2)^2 + 105*J2*Re^2*mu*z^4 - 90*J2*Re^2*mu*z^2*(x^2 + y^2 + z^2)

    % partials WRT J2
    A(4,7) = -(3*Re^2*mu*x*(x^2 + y^2 - 4*z^2))/(2*(x^2 + y^2 + z^2)^(7/2));

    A(5,7) = -(3*Re^2*mu*y*(x^2 + y^2 - 4*z^2))/(2*(x^2 + y^2 + z^2)^(7/2));

    A(6,7) = -(3*Re^2*mu*z*(3*x^2 + 3*y^2 - 2*z^2))/(2*(x^2 + y^2 + z^2)^(7/2));

    % Phi is the end of the Y column vector
    phiCol = Y(8:end);

    % reshape to be matrix
    phi = reshape(phiCol, [7,7]);

    % STM propagation
    phiDot = A * phi;

    % The state is the first part and phi is the second
    ydot(8:56) = reshape(phiDot, [49, 1]);

end

end
```

```
Not enough input arguments.

Error in Utility.NumericJ2Prop (line 9)
ydot(1:3,1) = Y(4:6,1);
```

Published with MATLAB® R2020b