AA528: HW5 P14.3

Berit Syltebo

March 2023

Problem 14.3

Two 10-orbit simulations were conducted using the exact nonlinear relative equations of motion given in Equation 1, and the linearized Clohessy-Wiltshire (CW) Equations given in both rectilinear (Equation 2) and curvilinear (Equation 3) coordinates. The first simulation uses an eccentricity of zero while the second using an eccentricity of $e = 0.0032364$ which is the maximum eccentricity before the difference between the rectilinear CW and nonlinear sims exceed 1 kilometer.

$$\begin{aligned}
\ddot{x} - 2\dot{f}\left(\dot{y} - y\frac{\dot{r}_c}{r_c}\right) - x\dot{f}^2 - \frac{\mu}{r_c^2} &= -\frac{\mu}{r_d^3}(r_c + x) \\
\ddot{y} + 2\dot{f}\left(\dot{x} - x\frac{\dot{r}_c}{r_c}\right) - y\dot{f}^2 &= -\frac{\mu}{r_d^3}y \\
\ddot{z} &= -\frac{\mu}{r_d^3}z
\end{aligned} \tag{1}$$

$$\begin{aligned}
\ddot{x} - 2n\dot{y} - 3n^2 x &= 0 \\
\ddot{y} + 2n\dot{x} &= 0 \\
\ddot{z} + n^2 z &= 0
\end{aligned} \tag{2}$$

$$\begin{aligned}
\delta\ddot{r} - 2nr_c\delta\dot{\theta} - 3n^2\delta r &= 0 \\
r_c\delta\ddot{\theta} + 2n\delta\dot{r} &= 0 \\
\ddot{z} + n^2 z &= 0
\end{aligned} \tag{3}$$

The following two plots (Figures 1 and 2) compare the outputs of the three methods described above. There are no surprises; all behave very well with the assumption that the chief and deputy masses have nearly circular orbits. There is very little deviation in all three axes and any provide accurate models for the relative motion.

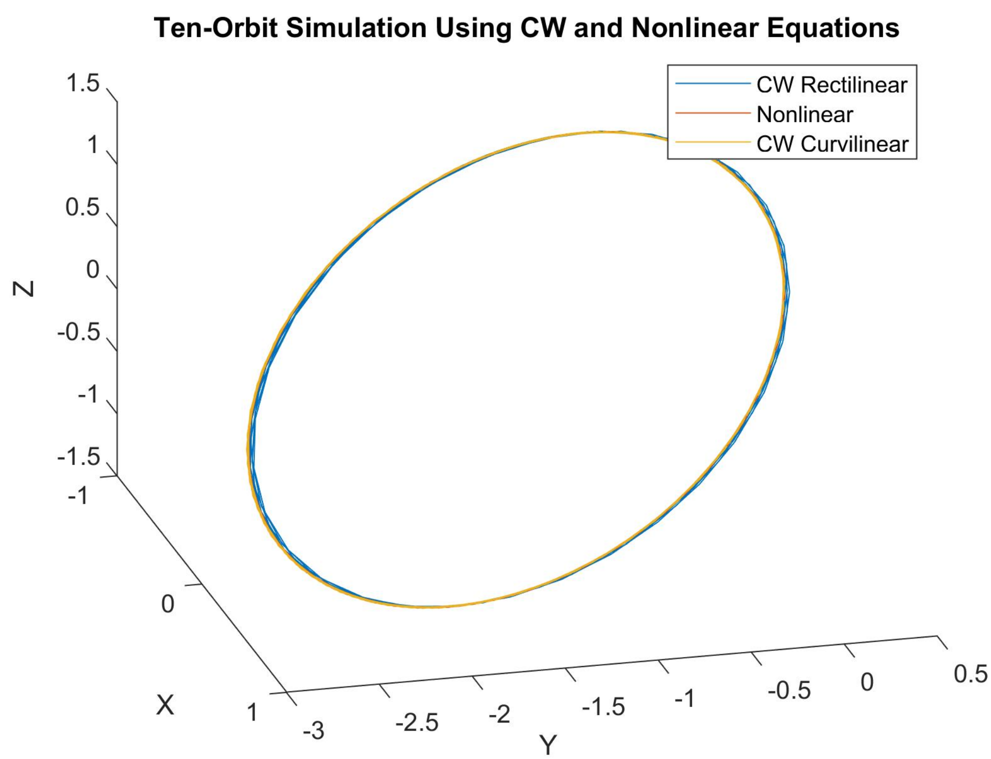

Figure 1: 10-orbit simulation using nonlinear and CW equations with zero eccentricity

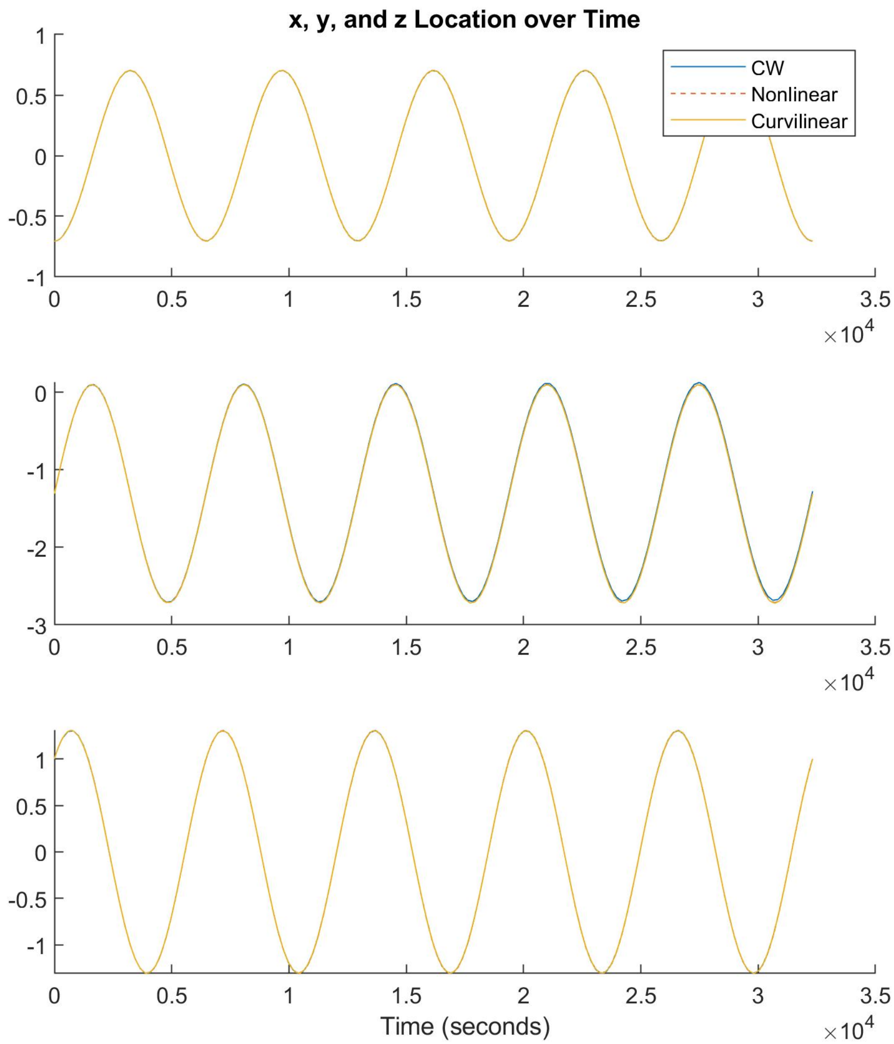

Figure 2: 10-orbit location over time across the three simulations with an eccentricity of zero

Using a non-zero eccentricity is much more exciting. There are some very clear inaccuracies between the nonlinear and CW representations. Most notably, the curvilinear estimation deviates a lot from the "exact" simulation from the nonlinear equations. The rectilinear behaves much better which is why it was chosen to find the critical eccentricity rather than the curvilinear.

Also interestingly, the nonlinear simulation shows drift that doesn't appear using the CW equations (this is difficult to see in the provided plots but is slightly visible). Knowing that we wouldn't expect drift with the chief and deputy having the same orbital periods and neglecting gravitational perturbations, there may be some inaccuracies in the solver used for the simulations: MATLAB's `ode89()`. This is speculation but there may be inaccuracies due to the complexity of the simulation and the step size used.

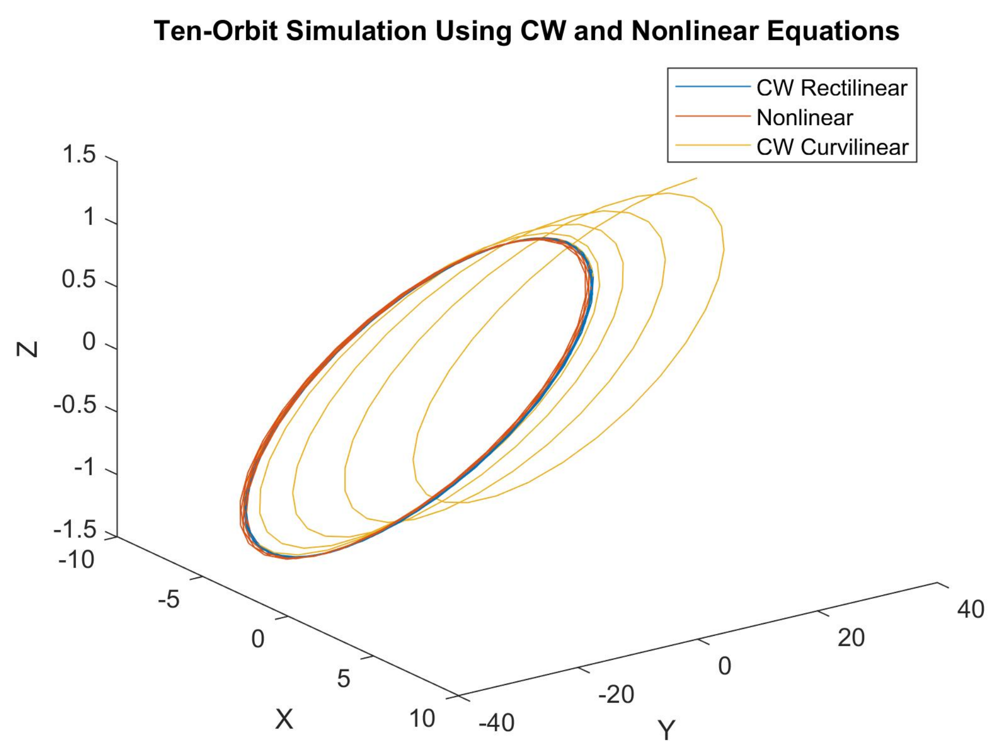

Figure 3: 10-orbit simulation using nonlinear and CW equations with critical eccentricity

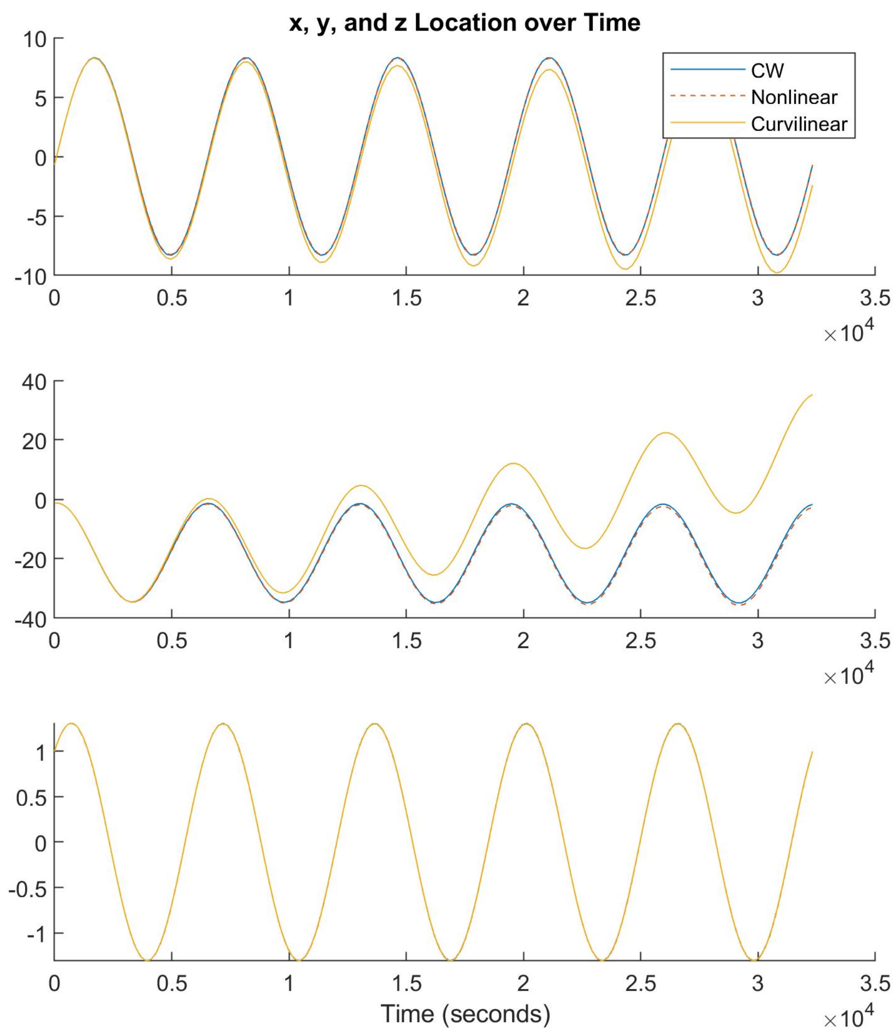

Figure 4: 10-orbit location over time across the three simulations with a critical eccentricity

```
clear all; close all; clc;


%% Inputs

a = 7500;
e = 0.0032364;
% e = 0.00000001;
i = 45 * pi/180;
Omega = 20 * pi/180;
omega = 30 * pi/180;
M0 = 20 * pi/180;

i_d = i + 0.01*pi/180;
e_d = e + 0.0001;
omega_d = omega - 0.01*pi/180;


%% Calculation of Initial Conditions

[r_c,v_c] = keplerian2ijk(a*1000,e,i*180/pi,Omega*180/pi,...
     omega*180/pi,M0*180/pi);
[r_d,v_d] = keplerian2ijk(a*1000,e_d,i_d*180/pi,Omega*180/pi,...
    omega_d*180/pi,M0*180/pi);

E_pre = 0:0.000001:2*pi;
tolerance = 10^(-6);
for j = 1:length(E_pre)
    term2 = E_pre(j) - e*sin(E_pre(j));
    if abs(M0-term2) <= tolerance
        E = E_pre(j);
        break
    end
end

f = 2 * atan(sqrt((1+e)/(1-e)) * tan(E/2));

r_c = r_c / 1000; v_c = v_c / 1000;
r_d = r_d / 1000; v_d = v_d / 1000;

t1 = r_c/norm(r_c);
t3 = cross(r_c,v_c)/norm(cross(r_c,v_c));
t2 = cross(t3,t1);

ON = [t1';t2';t3'];

rho = ON * (r_d - r_c);
rho_i = r_d - r_c;
rho_dot_i = v_d - v_c;
```

```
f_dot = norm(cross(r_c,v_c)) / norm(r_c)^2;
om = [0;0;f_dot];
v_c = ON * v_c - cross(om,ON*r_c);
v_d = ON * v_d - cross(om,ON*r_d);

rho_dot = ON * rho_dot_i - cross(om,ON*rho_i);
rho_dot(1) = v_c(1);

r_c = ON * r_c;
r_d = ON * r_d;

ICs = [norm(r_c);norm(v_c);f;f_dot;rho;rho_dot];

% Radial

dr = sqrt((norm(r_c)+rho(1))^2 + rho(2)^2) - norm(r_c);
dth = atan(rho(2)/(norm(r_c)+rho(1)));
z = rho(3);
drd = ((norm(r_c)+rho(1))*(norm(v_c)+rho_dot(1))+...
    rho(2)*rho_dot(2))/(norm(r_c)+norm(v_c)) - norm(v_c);
sd = norm(v_c)*atan(rho(2)/(norm(r_c)+rho(1)))-...
    norm(r_c)/(norm(r_c)+dr)^2*(rho(2)*(norm(v_c)+rho_dot(1))...
    -rho_dot(2)*(norm(r_c)+rho(1)));
s = norm(r_c)*dth;
dthd = sd/norm(r_c)-s*norm(v_c)/norm(r_c)^2;
zd = rho_dot(3);

ICsrad = [dr;dth;z;drd;dthd;zd;norm(r_c);norm(v_c);f;f_dot];


%% Simulation

G = 6.674 * 10^(-20); % km^3/kg-s^2
m1 = 5.97219 * 10^(24); % kg
mu = G * m1;

P = 2*pi*sqrt(a^3/mu);
tspan = [0 5*P];
x0 = [rho;rho_dot];
[t,x] = ode89(@CW,tspan,x0);
[ts,xs] = ode89(@instant,tspan,ICs);
[tr,xr] = ode89(@CWr,tspan,ICsrad);

xrad = [];
yrad = [];
for i = 1:length(xr)
    xrad(i) = xr(i,7)*(cos(xr(i,2)) - 1) + xr(i,1)*cos(xr(i,2));
    yrad(i) = sin(xr(i,2))*(xr(i,7) + xr(i,1));
end
zrad = xr(:,3);
```

```
figure(1)
hold on
plot3(x(:,1),x(:,2),x(:,3))
plot3(xs(:,5),xs(:,6),xs(:,7))
plot3(xrad,yrad,zrad)
xlabel('X')
ylabel('Y')
zlabel('Z')
legend('CW Rectilinear','Nonlinear','CW Curvilinear')
title('Ten-Orbit Simulation Using CW and Nonlinear Equations')

figure(2)
subplot(3,1,1)
hold on
plot(t,x(:,1))
plot(ts,xs(:,5),'--')
plot(tr,xrad)
legend('CW','Nonlinear','Curvilinear')
title('x, y, and z Location over Time')
subplot(3,1,2)
hold on
plot(t,x(:,2))
plot(ts,xs(:,6),'--')
plot(tr,yrad)
subplot(3,1,3)
hold on
plot(t,x(:,3))
plot(ts,xs(:,7),'--')
plot(tr,zrad)
xlabel('Time (seconds)')

diff = norm(x(end,1:3)-xs(end,5:7));


%% Functions

function dxdt = instant(~,in)

G = 6.674 * 10^(-20); % km^3/kg-s^2
m1 = 5.97219 * 10^(24); % kg
mu = G * m1;

x = in(5);
y = in(6);
z = in(7);
xdot = in(8);
ydot = in(9);
zdot = in(10);
rc = in(1);
```

```
vc = in(2);
% f = in(3);
fdot = in(4);

rd = sqrt((rc+x)^2 + y^2 + z^2);
fddot = -2 * vc/rc * fdot;
rcddot = rc*fdot^2 - mu/rc^2;

xddot = 2*fdot*(ydot-y*vc/rc) + x*fdot^2 + mu/rc^2 - mu/rd^3*(rc+x);
yddot = -2*fdot*(xdot-x*vc/rc) + y*fdot^2 - mu/rd^3*y;
zddot = -mu/rd^3*z;

dxdt = [vc;rcddot;fdot;fddot;xdot;ydot;zdot;xddot;yddot;zddot];

end


function dxdt = CWr(~,in)

G = 6.674 * 10^(-20); % km^3/kg-s^2
m1 = 5.97219 * 10^(24); % kg
mu = G * m1;

a = 7500;
n = sqrt(mu/a^3);

dr = in(1);
% dth = in(2);
z = in(3);
drd = in(4);
dthd = in(5);
zd = in(6);
rc = in(7);
vc = in(8);
% f = in(9);
fdot = in(10);

rcdd = rc*fdot^2-mu/rc^2;
fddot = -2 * vc/rc * fdot;

drdd = 2*n*rc*dthd + 3*n^2*dr;
dthdd = -2*n*drd/rc;
zdd = -n^2*z;

dxdt = [drd;dthd;zd;drdd;dthdd;zdd;vc;rcdd;fdot;fddot];

end


function dxdt = CW(~,in)
```

```
G = 6.674 * 10^(-20); % km^3/kg-s^2
m1 = 5.97219 * 10^(24); % kg
mu = G * m1;

a = 7500;
n = sqrt(mu/a^3);

x = in(1);
% y = in(2);
z = in(3);
xdot = in(4);
ydot = in(5);
zdot = in(6);

xddot = 2*n*ydot + 3*n^2*x;
yddot = -2*n*xdot;
zddot = -n^2*z;

dxdt = [xdot;ydot;zdot;xddot;yddot;zddot];

end
```