3.20

Contents

Initialize

```
clear; clc;
```

Define variables and constants

```
global mu
mu = 398600;        % km^3/s^2
```

Initial conditions

```
R0 = [20000 -105000 -19000];        % km
V0 = [0.9 -3.4 -1.5];               % km/s
t = 2*3600;                         % s
```

do algortihm (from appendix D.16) and output results

```
[R, V] = rv_from_r0v0(R0, V0, t);
fprintf("Final position: R=(%g, %g, %g) km", R(1), R(2), R(3));
fprintf("\nFinal velocity: V=(%g, %g, %g) km/s", V(1), V(2), V(3));
```

```
Final position: R=(26337.8, -128752, -29655.9) km
Final velocity: V=(0.862796, -3.2116, -1.46129) km/s
```

Published with MATLAB® R2021a

4.2

Contents

Initialize

```
clear; clc;
```

Define variables and constants

```
Re = 6378;          % km
global mu
mu = 398600;        % km^3/s^2
```

given initial conditions

```
h = 500;            % km
a0 = 300;           % deg
d0 = -20;           % deg
```

determine initial location and velocity

```
R0 = (6378 + h)*[cosd(d0)*cosd(a0) cosd(d0)*sind(a0) sind(d0)];     % km
V0 = [0 0 10];                                                      % km/s
t = 1800;                                                           % s
```

use algorithm from appendix again

```
[R, V] = rv_from_r0v0(R0, V0, t);
```

use algortihm 4.1

get these vars

```
r = norm(R);        % km
l = R(1)/r;
m = R(2)/r;
n = R(3)/r;

% calc and return del
d = asind(n);
fprintf("del: %g deg", d);
```

```
% check value of m
fprintf("\nm: %g", m);

% b/c m>0
a = acosd(l/cosd(d));
fprintf("\nalpha: %g deg", a);
```

```
del: 63.7473 deg
m: 0.38307
alpha: 120 deg
```

Published with MATLAB® R2021a

4.3

Contents

Initialize

```
clear; clc;
```

Define variables and constants

```
mu = 398600;        % km^3/s^2
```

Initial conditions

```
R = [2500 16000 4000];     % km
V = [-3 -1 5];             % km/s
```

run matlab script from appendix

```
coe = coe_from_sv(R, V, mu);

% outputs
% coe = [h e RA incl w TA a]
fprintf("e=%g", coe(2));
fprintf("\nh=%g km^2/s", coe(1));
fprintf("\ni=%g deg", rad2deg(coe(4)));
fprintf("\nOmega=%g deg", rad2deg(coe(3)));
fprintf("\nw=%g deg", rad2deg(coe(5)));
fprintf("\ntheta=%g deg", rad2deg(coe(6)));
```

```
e=0.465759
h=98623 km^2/s
i=62.5256 deg
Omega=73.7398 deg
w=22.0805 deg
theta=353.6 deg
```

Published with MATLAB® R2021a

4.6

Contents

Initialize

```
clear; clc;
```

given vals

```
R = [-6000 -1000 -5000];        % km
e = [0.4 0.5 0.6];
```

determine magnitudes

```
r = norm(R);
em = norm(e);
```

calculate true anomaly

approaching perigee so:

```
theta = 360 - acosd(dot(R, e)/(em*r));
fprintf("true anomaly: %g deg", theta);
```

```
true anomaly: 211.361 deg
```

Published with MATLAB® R2021a