Contents

Initialize

```
clear; clc;
```

Problem 6.34

Define variables and constants

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

Initial conditions

```
R0 = [0 16000 0];       % km
e2 = 0.5;
h2 = (mu*R0(2)*(1+e2))^(0.5);
Vb = h2/R0(2);          % km/s
V0 = [0 0 Vb];       % km/s
t = 60*60;             % s
```

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

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

```
Final position vec: R=(0, 7829.06, 18496.8) km
Final velocity vec: V=(0, -3.75299, 3.62618) km/s
```

setup for lambert

```
r1 = [10000 0 0];               % km
r2 = R;                         % km
dt = 60*60;                     % s
string = "pro";
```

use lambert function (alogrithm 5.2) from book to get v at each position

```
[v1, v2] = lambert(r1, r2, dt, string);

% output
fprintf("\nV1 velocity vec: V=(%g, %g, %g) km/s", v1(1), v1(2), v1(3));
fprintf("\nV2 velocity vec: V=(%g, %g, %g) km/s", v2(1), v2(2), v2(3));
```

```
V1 velocity vec: V=(0.948538, 3.13609, 7.40927) km/s
V2 velocity vec: V=(-4.0057, 1.20499, 2.84689) km/s
```

determine delta v req.

velocity at a (circular orbit):

```
vA = [0 (mu/norm(r1))^(0.5) 0];     % km/s

% subtract from V1/Va3
dVA = v1 - vA;                  % km/s (vector)
dV = norm(dVA);                 % km/s
fprintf("\ndV reqd: %g km/s", dV);
```

```
dV reqd: 8.11744 km/s
```

Published with MATLAB® R2021a