
John Clouse IMD HW 3

Table of Contents

Initialize	1
Problem 1	1
Problem 2	1
Problem 3	3
Problem 4	3

Initialize

```
clearvars -except hw_pub function_list
```

```
CelestialConstants
```

Problem 1

```
JD_depart = 2454085.5;
JD_arrive = JD_depart + 830;
[r_p1, v_p1] = MeeusEphemeris(Mars, JD_depart, Sun);
[r_p2, v_p2] = MeeusEphemeris(Jupiter, JD_arrive, Sun);

fprintf('r_mars:\n');disp(r_p1); fprintf('\b\b km\n\n')
fprintf('r_jupiter:\n');disp(r_p2); fprintf('\b\b km\n\n')

r_mars:
    1.0e+08 *

    -1.2817
    -1.9059
    -0.0084 km

r_jupiter:
    1.0e+08 *

    4.8338
    -5.8746
    -0.0838 km
```

Problem 2

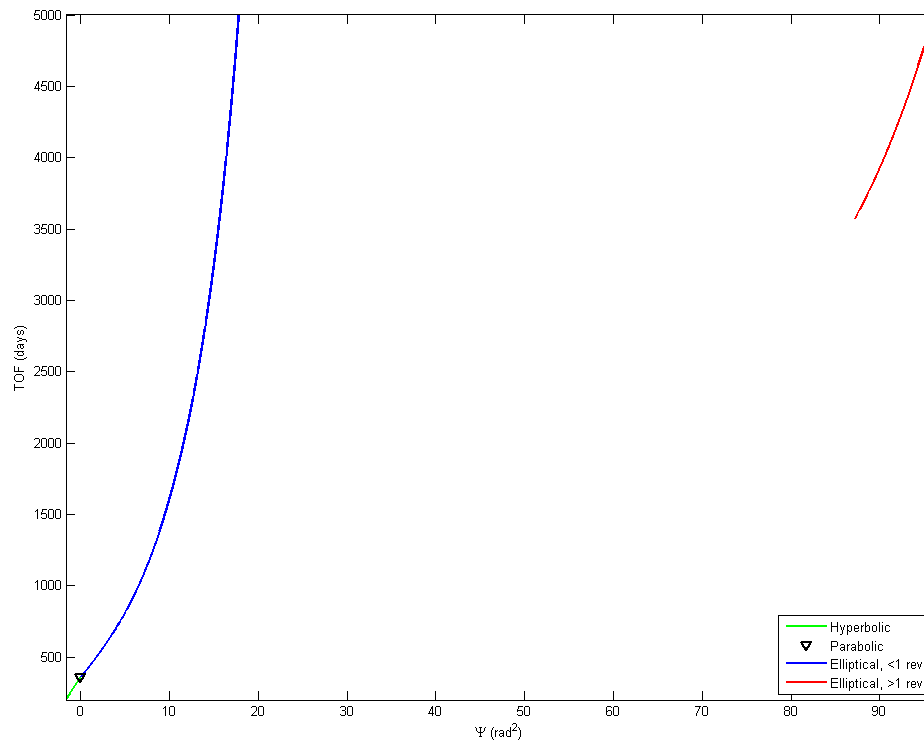
```
day_store = 200:5000;
psi_store = zeros(1,length(day_store));
% get psi out of Lambert
for ii = 1:length(day_store)
    [~, ~, psi_store(ii)] = lambert(r_p1, r_p2, day_store(ii)*24*3600, 1, Sun);
end
```

```
% Find the initial psi value that yields a multi-rev solution
psi_multi_rev = psi_store(end);
psi_out = psi_store(end);
while psi_out <= psi_store(end)+1
    psi_multi_rev = psi_multi_rev+10;
    [~, ~, psi_out] = lambert(r_p1, r_p2, day_store(end)*24*3600, 1, Sun,psi_multi_rev);
end

% Run multi-rev Lambert
psi_multirev_store = zeros(1,length(day_store));
for ii = length(day_store):-1:1
    [~, ~, psi_multirev_store(ii)] = lambert(r_p1, r_p2, day_store(ii)*24*3600, 1, Sun,psi_multi_rev);
end

figure('Position', hw_pub.figPosn)
ell_plot = plot(psi_store, day_store,'LineWidth',hw_pub.lineWidth);
hold on
hyp_plot = plot(psi_store(psi_store < 0), day_store(psi_store < 0),'g',...
    'LineWidth',hw_pub.lineWidth);
para_plot = plot(max(psi_store(psi_store < 0)), ...
    max(day_store(psi_store < 0)), 'kv', 'LineWidth',hw_pub.lineWidth);
mr_indices = psi_multirev_store > psi_store +1;
mr_plot = plot(psi_multirev_store(mr_indices), day_store(mr_indices),'r',...
    'LineWidth',hw_pub.lineWidth);

ylabel('TOF (days)')
xlabel('\Psi (rad^2)')
xlim([min(psi_store) max(psi_multirev_store)]);
ylim([day_store(1) day_store(end)]);
legend([hyp_plot para_plot ell_plot mr_plot], 'Hyperbolic', 'Parabolic', ...
    'Elliptical, <1 rev', 'Elliptical, >1 rev', 'Location', 'SouthEast');
```



Problem 3

I added an argument to make the initial guess for Psi to be larger. I also had to implement a bailout mechanism, as sometimes it would get stuck on some other local minimum and not converge.

Problem 4

```
min_days = min(day_store(mr_indices));  
fprintf('Minimum time for multi-rev transfer is %d days.\n', min_days);  
fprintf('Psi = %.3f rad^2\n', min(psi_multirev_store(mr_indices)));
```

```
Minimum time for multi-rev transfer is 3564 days.  
Psi = 87.116 rad^2
```

Published with MATLAB® R2013b