Table of Contents


```
% Homework 1
% Aero 452
% Liam Hood
function Aero_557_HW1()

clear ; close all ; clc ;
pt = 'Problem number %u \n \n' ;
```

1

```
fprintf( pt , 1 )
% far distance, 10 periods
HW1_P1()
fprintf( ' \n' )
```

```
Problem number 1

-------Gauss-----------
The position is 17705.179022 km
-6964.287452 km
-12774.235639 km
-10089.151021 km

The velocity is 5.989709 km/s
1.546611 km/s
-5.083867 km/s
-2.763857 km/s

    "h"                         "67179.6441"    "km^2/s"
    "inclination"               "34.8033173"    "degrees"
    "eccentricity"              "0.86028923"    "   "
    "RAAN"                      "335.360427"    "degrees"
    "argument of perigee"       "151.973012"    "degrees"
    "true anomaly"              "114.774615"    "degrees"
    "semi-major axis"           "43563.9995"    "km"
    "radius of periapsis"       "6086.35993"    "km"
    "radius of apoapsis"        "81041.6391"    "km"

-------Double-R Iteration-----------
The position is 10603.887018 km
5947.440788 km
5729.581440 km
```

```
6651.485885 km

The velocity is 6.487724 km/s
-4.348063 km/s
4.574956 km/s
1.501564 km/s

    "h"                        "68013.5424"     "km^2/s"
    "inclination"              "39.9731744"     "degrees"
    "eccentricity"             "0.191319342"    "  "
    "RAAN"                     "330.030298"     "degrees"
    "argument of perigee"      "17.1040435"     "degrees"
    "true anomaly"             "60.4239653"     "degrees"
    "semi-major axis"          "12046.1494"     "km"
    "radius of periapsis"      "9741.48806"     "km"
    "radius of apoapsis"       "14350.8108"     "km"

The gauss method is not as accurate as the double-r method
```

2

```
fprintf( pt , 2 )
% circular point in time
HW1_P2()
fprintf( ' \n' )
```

```
Problem number 2

-------------Minimum Energy Solution---------------------
Velocity at position 1 is 3.569547 km/s
2.047408 km/s
2.924002 km/s
0.000000 km/s
    "h"                        "46624.1995"     "km^2/s"
    "inclination"              "0"              "degrees"
    "eccentricity"             "0.700207531"    "  "
    "RAAN"                     "0"              "degrees"
    "argument of perigee"      "0"              "degrees"
    "true anomaly"             "159.999999"     "degrees"
    "semi-major axis"          "10699.4839"     "km"
    "radius of periapsis"      "3207.62468"     "km"
    "radius of apoapsis"       "18191.343"      "km"

-------------Izzo/Gooding Solution-----------------------
Velocity at position 1 is 3.569588 km/s
2.058911 km/s
2.915964 km/s
0.000000 km/s
Velocity at position 2 is 3.569588 km/s
-3.451562 km/s
0.910315 km/s
0.000000 km/s
```

```
    "h"                         "46496.0336"      "km^2/s"
    "inclination"               "0"               "degrees"
    "eccentricity"              "0.702205826"     "  "
    "RAAN"                      "0"               "degrees"
    "argument of perigee"       "0"               "degrees"
    "true anomaly"              "159.999999"      "degrees"
    "semi-major axis"           "10699.5681"      "km"
    "radius of periapsis"       "3186.26905"      "km"
    "radius of apoapsis"        "18212.8672"      "km"

-------------Gauss Solution-----------------------------
Velocity at position 1 is 3.569613 km/s
2.058939 km/s
2.915974 km/s
0.000000 km/s
Velocity at position 2 is 3.569613 km/s
-3.451591 km/s
0.910305 km/s
0.000000 km/s
    "h"                         "46496.1994"     "km^2/s"
    "inclination"               "0"              "degrees"
    "eccentricity"              "0.70220498"     "  "
    "RAAN"                      "0"              "degrees"
    "argument of perigee"       "0"              "degrees"
    "true anomaly"              "159.99961"      "degrees"
    "semi-major axis"           "10699.6194"     "km"
    "radius of periapsis"       "3186.29336"     "km"
    "radius of apoapsis"        "18212.9453"     "km"

-------------Universal Variable Solution-----------------
Velocity at position 1 is 3.569588 km/s
2.058911 km/s
2.915963 km/s
0.000000 km/s
Velocity at position 2 is 3.569588 km/s
-3.451563 km/s
0.910315 km/s
0.000000 km/s
    "h"                         "46496.0275"      "km^2/s"
    "inclination"               "0"               "degrees"
    "eccentricity"              "0.702205921"     "  "
    "RAAN"                      "0"               "degrees"
    "argument of perigee"       "0"               "degrees"
    "true anomaly"              "159.999999"      "degrees"
    "semi-major axis"           "10699.5681"      "km"
    "radius of periapsis"       "3186.26805"      "km"
    "radius of apoapsis"        "18212.8682"      "km"

-------------Battin Solution----------------------------
Velocity at position 1 is 3.569588 km/s
2.058911 km/s
2.915964 km/s
0.000000 km/s
Velocity at position 2 is 3.569588 km/s
```

```
-3.451562 km/s
0.910315 km/s
0.000000 km/s
    "h"                        "46496.0334"     "km^2/s"
    "inclination"              "0"              "degrees"
    "eccentricity"             "0.702205827"    "  "
    "RAAN"                     "0"              "degrees"
    "argument of perigee"      "0"              "degrees"
    "true anomaly"             "160"            "degrees"
    "semi-major axis"          "10699.5681"     "km"
    "radius of periapsis"      "3186.26903"     "km"
    "radius of apoapsis"       "18212.8671"     "km"

The minimum energy method differs the most from the other methods
because it is not constrained to the same time of transfer as the
other methods. Battin method is most similar to Izzo's solution
with the difference on the order of 10^-7. Gauss is on the order
of 10^-4, and Universal Variable is on the order of 10^-5.The
largest difference for all is in the angular momentum. The
differences appear because each method uses different series
approximations and variables to iterate on.
```

Work

```
function HW1_P1()
    % Compare State and COES from double-r and gauss (extended and
    % Discuss difference and thoughts on each of the methods
    mu = 398600 ;
    coesName = [ "h" ; "inclination" ; "eccentricity" ; "RAAN"
; "argument of perigee" ; "true anomaly" ; "semi-major axis"
; "radius of periapsis" ; "radius of apoapsis" ] ;
    coesUnits = [ "km^2/s" ; "degrees" ; "  " ; "degrees"
; "degrees" ; "degrees" ; "km" ; "km" ; "km" ] ;
    d2s = 24*60*60 ;

    time = [ 11 , 30 , 0 ; 11 , 50 , 0 ; 12 , 0 , 0 ] ;
    rtasc = [ -33.0588410 ; 55.0931551 ; 98.7739537 ] ;
    decli = [ -7.2056382 ; 36.5731946 ; 31.1314513 ] ;
    lat = 40 ;
    long = -110 ;
    alt = 2000 ;
    llasite = [ lat , long , alt ] ;
    day = [ 2010 , 8 , 20 ] ;

    UT =  [ [ day ; day ; day ] , time ] ;
    R = lla2eci( [ llasite ; llasite ; llasite ] , UT )*1e-3 ;

    qhat = zeros(3) ;
    for ii = 1:3
        qhat(1,ii) = cosd( decli(ii) )*cosd( rtasc(ii) ) ;
        qhat(2,ii) = cosd( decli(ii) )*sind( rtasc(ii) ) ;
        qhat(3,ii) = sind( decli(ii) ) ;
```

```
        end

        jt = juliandate( UT ) ;
        tau1 = ( jt(1) - jt(2) )*d2s ;
        tau3 = ( jt(3) - jt(2) )*d2s ;

        [ r2_g , v2_g ] = GaussIOD( qhat(:,1) , qhat(:,2) ,
qhat(:,3) , R(1,:)' , R(2,:)' , R(3,:)' , tau1 , tau3 , mu ) ;
        COES_g = state2coes_display( [ r2_g ; v2_g ] , mu ) ;
        fprintf( '-------Gauss-----------\n' )
        fprintf( 'The position is %f km \n' , norm( r2_g ))
        fprintf( '%f km \n' , r2_g )
        fprintf( '\n' )
        fprintf( 'The velocity is %f km/s \n' , norm( v2_g ))
        fprintf( '%f km/s \n' , v2_g)
        fprintf( '\n' )
        coesOut_g = [ coesName , COES_g' , coesUnits ] ;
        disp( coesOut_g ) ;

        [ r2vec_dr , v2vec_dr ] = DoubleR( qhat(:,1) , qhat(:,2) ,
qhat(:,3) , R(1,:)' , R(2,:)' , R(3,:)' , tau1 , tau3 ) ;
        COES_dr = state2coes_display( [ r2vec_dr ; v2vec_dr ] , mu ) ;
        fprintf( '-------Double-R Iteration-----------\n' )
        fprintf( 'The position is %f km \n' , norm( r2vec_dr ))
        fprintf( '%f km \n' , r2vec_dr )
        fprintf( '\n' )
        fprintf( 'The velocity is %f km/s \n' , norm( v2vec_dr ))
        fprintf( '%f km/s \n' , v2vec_dr)
        fprintf( '\n' )
        coesOut_dr = [ coesName , COES_dr' , coesUnits ] ;
        disp( coesOut_dr ) ;

        fprintf( 'The gauss method is not as accurate as the double-r
method \n' )
    end

    function HW1_P2()
        % Using universal variable method, gauss’ method, Izzo/Gooding
method (from MATLAB central)
        % and minimum energy, find and compare the velocity vector for
the orbit give two positions, a
        % difference in time, and going the short way around.
        mu = 398600 ;
        coesName = [ "h" ; "inclination" ; "eccentricity" ; "RAAN"
; "argument of perigee" ; "true anomaly" ; "semi-major axis"
; "radius of periapsis" ; "radius of apoapsis" ] ;
        coesUnits = [ "km^2/s" ; "degrees" ; "  " ; "degrees"
; "degrees" ; "degrees" ; "km" ; "km" ; "km" ] ;
        d2s = 24*60*60 ;

        r1 = [ 15945.34 ; 0 ; 0 ] ;
        r2 = [ 12214.83899 ; 10249.46731 ; 0 ] ;
        dtsec = 76*60 ;
        dtday = dtsec/d2s ;
```

```
        tm = "short" ;

        [ v1_min , amin , emin , tmin , tp ] = Lambert_MinEnergy( r1 ,
r2 , mu ) ;
        fprintf( '-------------Minimum Energy
Solution---------------------\n' )
        fprintf( 'Velocity at position 1 is %f km/s \n' ,
norm( v1_min ) )
        fprintf( '%f km/s \n' , v1_min )
        COES_min = state2coes_display( [ r1 ; v1_min ] , mu ) ;
        coesOut_min = [ coesName , COES_min' , coesUnits ] ;
        disp( coesOut_min ) ;

        [ v1_izzo , v2_izzo ] = Lambert_Izzo( r1 , r2 , dtsec , 1 ,
mu ) ;
        fprintf( '-------------Izzo/Gooding
Solution----------------------\n' )
        fprintf( 'Velocity at position 1 is %f km/s \n' ,
norm( v1_izzo ) )
        fprintf( '%f km/s \n' , v1_izzo )
        fprintf( 'Velocity at position 2 is %f km/s \n' ,
norm( v2_izzo ) )
        fprintf( '%f km/s \n' , v2_izzo )
        COES_izzo1 = state2coes_display( [ r1 ; v1_izzo ] , mu ) ;
        coesOut_izzo1 = [ coesName , COES_izzo1' , coesUnits ] ;
        disp( coesOut_izzo1 ) ;

        [ v1_gauss , v2_gauss ] = Lambert_Gauss( r1 , r2 , dtsec , 1 ,
mu ) ;
        fprintf( '-------------Gauss
Solution-----------------------------\n' )
        fprintf( 'Velocity at position 1 is %f km/s \n' ,
norm( v1_gauss ) )
        fprintf( '%f km/s \n' , v1_gauss )
        fprintf( 'Velocity at position 2 is %f km/s \n' ,
norm( v2_gauss ) )
        fprintf( '%f km/s \n' , v2_gauss )
        COES_gauss = state2coes_display( [ r1 ; v1_gauss ] , mu ) ;
        coesOut_gauss = [ coesName , COES_gauss' , coesUnits ] ;
        disp( coesOut_gauss ) ;

        [ v1_uv , v2_uv ] = Lambert_UV( r1 , r2 , dtsec , 1 , mu ) ;
        fprintf( '-------------Universal Variable
Solution----------------\n' )
        fprintf( 'Velocity at position 1 is %f km/s \n' ,
norm( v1_uv ) )
        fprintf( '%f km/s \n' , v1_uv )
        fprintf( 'Velocity at position 2 is %f km/s \n' ,
norm( v2_uv ) )
        fprintf( '%f km/s \n' , v2_uv )
        COES_uv = state2coes_display( [ r1 ; v1_uv ] , mu ) ;
        coesOut_uv = [ coesName , COES_uv' , coesUnits ] ;
        disp( coesOut_uv ) ;
```

```
        [ v1_bat , v2_bat ] = Lambert_Battin( r1 , r2 , dtsec , 1 ,
 mu ) ;
        fprintf( '-------------Battin
 Solution-----------------------------\n' )
        fprintf( 'Velocity at position 1 is %f km/s \n' ,
 norm( v1_bat ) )
        fprintf( '%f km/s \n' , v1_bat )
        fprintf( 'Velocity at position 2 is %f km/s \n' ,
 norm( v2_bat ) )
        fprintf( '%f km/s \n' , v2_bat )
        COES_bat = state2coes_display( [ r1 ; v1_bat ] , mu ) ;
        coesOut_bat = [ coesName , COES_bat' , coesUnits ] ;
        disp( coesOut_bat ) ;

COES_Dif_bat = ( COES_izzo1 - COES_bat )./COES_izzo1 .* 100 ;
COES_Dif_gauss = ( COES_izzo1 - COES_gauss )./COES_izzo1 .* 100 ;
COES_Dif_uv = ( COES_izzo1 - COES_uv )./COES_izzo1 .* 100 ;

fprintf( 'The minimum energy method differs the most from the other
 methods \n' )
fprintf( 'because it is not constrained to the same time of transfer
 as the \n' )
fprintf( 'other methods. Battin method is most similar to Izzo''s
 solution \n' )
fprintf( 'with the difference on the order of 10^-7. Gauss is on the
 order \n' )
fprintf( 'of 10^-4, and Universal Variable is on the order of
 10^-5.The \n' )
fprintf( 'largest difference for all is in the angular momentum. The
 \n' )
fprintf( 'differences appear because each method uses different series
 \n' )
fprintf( 'approximations and variables to iterate on. \n' )

    end
```

Functions

```
end
```

Published with MATLAB® R2019a