

Homework 3b Code

Cameron Lane

CJL3282

QUESTION 1:

```
clear
```

```
clc
```

```
month = 2;  
day = 19;  
year = 2022;  
hour = 5;  
minute = 59;  
second = 0;  
UTC = [year month day hour minute second];  
x_p = 0.016768;  
y_p = 0.357565;  
deltaUT1 = -.110914;  
dX = 0.000066;  
dY = -0.000049;  
X = 438.4313;  
Y = 4.391260;  
s = -0.006758;
```

```
UT1 = UTC;  
UT1(4) = UTC(4) - 1;  
UT1(5) = UTC(5) + 59;  
UT1(6) = UTC(6) + deltaUT1 + 60;  
JulianDateUT1 = convertJD( UT1 );
```

```
thetaERA = 2 * pi() * (0.7790572732640 + 1.00273781191135448 * (JulianDateUT1 -  
2451545.0));
```

```
TAI = UTC;  
TAI(6) = UTC(6) + 37;  
TT = TAI;  
TT(5) = TAI(5) + 1;  
TT(6) = 9.184;  
JulianDateTT = convertJD( TT );  
CenturiesTT = Ttt( JulianDateTT );  
x_p = arcSecRad( x_p );  
y_p = arcSecRad( y_p );  
stilt = -0.000047 * CenturiesTT;
```

```

stilt = arcSecRad( stilt );
w_R3 = R3(stilt);
w_R2 = R2(x_p);
w_R1 = R1(y_p);
W = w_R3 * w_R2 * w_R1
R = R3(-thetaERA)
s = arcSecRad( s );
X = arcSecRad( X );
Y = arcSecRad( Y );
dX = arcSecRad( dX );
dY = arcSecRad( dY );
X = X + dX;
Y = Y + dY;
a = 0.5 + 0.125*(X^2 + Y^2);
pn = [1-a*Y^2 -a*X*Y X; -a*X*Y 1-a*Y^2 Y; -X -Y 1-a*(X^2 + Y^2)];
pn_R3 = R3(s);
PN = pn * pn_R3
Q = PN * R * W

```

QUESTION 2:

```

clear
clc
format long g
timeUT1 = [2022 2 19 5 58 59];
timeUTC = [2022 2 19 5 59 0];
EOP = [0.041075 0.337800 -.200316 0.000146 -0.000045 396.8722 -1.481681 -0.000904];
ritrf = [-742.845 -5463.244 3196.066];
thetaGMST = UT1ThetaGMST( timeUT1 );
thetaGMST = thetaGMST * pi / 180
q1 = R3(-thetaGMST
q2 = Qitrf( timeUTC, EOP )
r1 = q1 * ritrf
r2 = q2 * ritrf
error = norm(r1 - r2)

```

FUNCTIONS:

```

function [ radian ] = arcSecRad( arcSec )
radian = arcSec * pi() / 648000;
end

```

```

function [ JulianDate ] = convertJD( time )
JulianDate = 367*time(1) - floor(7 / 4* (time(1) + floor((time(2) + 9)/12))) +
floor(275*time(2)/9) + time(3) + 1721013.5 + (1/24)*(time(4) + (1/60)*(time(5) + time(6)/60))

```

End

```
function [ rotatedR1 ] = R1( angle )
rotatedR1 = [1 0 0 ; 0 cos(angle) sin(angle); 0 -sin(angle) cos(angle)];
end
```

```
function [ rotatedR2 ] = R2( angle )
rotatedR2 = [cos(angle) 0 -sin(angle); 0 1 0; sin(angle) 0 cos(angle)];
end
```

```
function [ rotatedR3 ] = R3( angle )
rotatedR3 = [cos(angle) sin(angle) 0; -sin(angle) cos(angle) 0; 0 0 1];
end
```

```
function [ thetaGMST ] = UT1ThetaGMST( UT1 )
JulianDateUT1 = convertJD( UT1 );
CenturiesUT1 = Tut1(JulianDateUT1);
thetaGMST = 67310.54841 + (876600*3600 + 8640184.812866)*CenturiesUT1 +
0.093194*CenturiesUT1^2 - 6.2* 10^(-6)*CenturiesUT1^3;
thetaGMST = thetaGMST * 15 / 3600;
overlap = floor(thetaGMST / 360);
if thetaGMST > 360
    thetaGMST = thetaGMST - (360 * overlap);
end
end
```

```
function [T,Y] = propCartODE45(rv_matrix, time, mu)
odeoptions = odeset('RelTol',1e-10,'AbsTol',1e-20);
[T,Y] = ode45( @(t,y) dy(t, y, mu), time, rv_matrix, odeoptions );
```

```
function dy = dy(t,y, mu)
y1to3 = norm(y(1:3));
dy4 = -mu * y(1) / (y1to3)^3;
dy5 = -mu * y(2) / (y1to3)^3;
dy6 = -mu * y(3) / (y1to3)^3;

dy = [ y(4); y(5); y(6); dy4; dy5; dy6];
end
end
```

```
function [ seconds ] = JDSeconds( inputTime )
seconds = inputTime * 84600;
end
```

```
function [ UTC ] = TTUTC( TTSec )
deltaAT = 37;
```

```
TAI = TTSec - 32.184;  
UTC = TAI - deltaAT;  
end
```

```
function [ CenturiesTT ] = Ttt( JulianTT )  
CenturiesTT = (JulianTT - 2451545) / 36525;  
end
```

```
function [ CenturiesUT1 ] = Tut1( JulianUT1 )  
CenturiesUT1 = (JulianUT1 - 2451545) / 36525;  
end
```

```
function [Q] = Qitr( UTC, EOP )  
x_p = EOP(1);  
y_p = EOP(2);  
deltaUT1 = EOP(3);  
dX = EOP(4);  
dY = EOP(5);  
X = EOP(6);  
Y = EOP(7);  
s = EOP(8);  
UT1 = UTC;  
UT1(4) = UTC(4) - 1;  
UT1(5) = UTC(5) + 59;  
UT1(6) = UTC(6) + deltaUT1 + 60  
JulianDateUT1 = convertJD( UT1 );
```

```
thetaERA = 2 * pi() * (0.7790572732640 + 1.00273781191135448 * (JulianDateUT1 -  
2451545.0));
```

```
TAI = UTC;  
TAI(6) = UTC(6) + 37;  
TT = TAI;  
TT(6) = TAI(6) + 32.184;  
JulianDateTT = convertJD( TT );  
CenturiesTT = Ttt( JulianDateTT );  
x_p = arcSecRad( x_p );  
y_p = arcSecRad( y_p );  
stilt = -0.000047 * CenturiesTT;  
stilt = arcSecRad( stilt );  
w_R3 = R3(stilt);  
w_R2 = R2(x_p);  
w_R1 = R1(y_p);  
W = w_R3 * w_R2 * w_R1;  
R = R3(-thetaERA);
```

```

s = arcSecRad( s );
X = arcSecRad( X );
Y = arcSecRad( Y );
dX = arcSecRad( dX );
dY = arcSecRad( dY );
X = X + dX;
Y = Y + dY;
a = 0.5 + 0.125*(X^2 + Y^2);
pn = [1-a*Y^2 -a*X*Y X; -a*X*Y 1-a*Y^2 Y; -X -Y 1-a*(X^2 + Y^2)];
pn_R3 = R3(s);
PN = pn * pn_R3;
Q = PN * R * W;
end

```