Assignment 4:
Reference systems and transformations

Questions

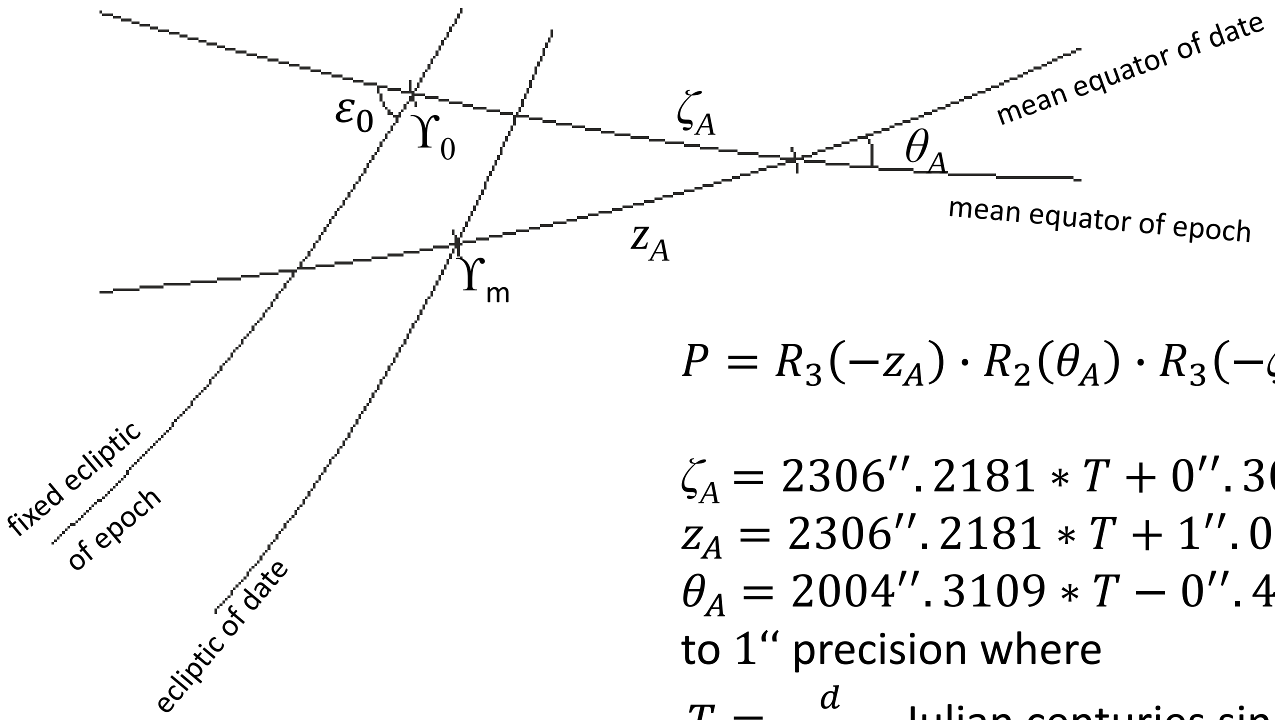
Dr. Robert Heinkelmann

Exercise 1

Write code that computes the precession matrix based on the NEWCOMB precession angles, $P = R_3(-z_A)\, R_2(\theta_A)\, R_3(-\zeta_A)$, where R denotes a 3D-rotation matrix.

Exercise 1

Precession, numerical expressions based on J2000.0 (i.e. FK5)
Method based on Newcomb parameters (prior to 2003)
ζ_A, θ_A, z_A: precession angles



$P = R_3(-z_A) \cdot R_2(\theta_A) \cdot R_3(-\zeta_A)$ with

$\zeta_A = 2306''.2181 * T + 0''.30188 * T^2$
$z_A = 2306''.2181 * T + 1''.09468 * T^2$
$\theta_A = 2004''.3109 * T - 0''.42665 * T^2$
to 1“ precision where

$T = \frac{d}{36525}$ Julian centuries since J2000.0
$d = JD - 2451545.0$ Julian days
$JD = \cdots$ (Julian date see assignment#2)

Exercise 1

Write code that computes the precession matrix based on the NEWCOMB precession angles, $P = R_3(-z_A)\ R_2(\theta_A)\ R_3(-\zeta_A)$, where R denotes a 3D-rotation matrix.

prec.m | a4e1.mlx

```
function [P] = prec(jd)
% PREC computes the 3x3 precession matrix P.
%     Precision of computation is around one second of arc.
%     usage:
%     [P] = precang(jd)
%     input:
%     jd: UT1 time as Julian Date (days.dec)
%     output:
%     P (3x3) precession matrix
%     function files:
%     dms2deg
%     rot3d
```

The difference between UT1 and UTC can be neglected for this exercise!

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

E.g. with a MatLab ***live script***

1) coordinates of a fictitious celestial object at the vernal equinox (0; 0):
$\alpha = 0;\ \delta = 0$

2) Celestial to Cartesian coordinate conversion
$\alpha, \delta \rightarrow \boldsymbol{x_{i0}}$?

option: use your function (assignment #3)
```
[xi0(1),xi0(2),xi0(3)] = plr2xyz(delta,alpha,1);%(rad,rad,m)-->(m,m,m)
```
Attention! Mind the order of appearance is delta, alpha, rad to agree with phi, lambda, rad

or think a bit to get the Cartesian coordinates `[xi0(1) xi0(2) xi0(3)] = [1 0 0];`

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

3) during the years 1990 to 2030
with annual resolution

```
yyyy=1990:2030
mm=1; dd=1; ut1=12; minute=0; second=0;
```

this choice of date and time is favorable because with this choice the precession should pass through zero for `yyyy=2000` J2000.0 = 2000-01-01 12:00:00 (control point)

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
   where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
   function gre2jd (assignment #2)

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
function gre2jd (assignment #2)
b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
   where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
   function gre2jd (assignment #2)
b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)
c) multiply the precession matrix with the Cartesian coordinate vector at J2000.0
   $\boldsymbol{x}_{\boldsymbol{iP}}(JD) = P(JD) \cdot \boldsymbol{x}_{\boldsymbol{i0}}$; as $P \in \mathbb{R}^{3\times3}$, it must be $\boldsymbol{x}_{\boldsymbol{i0}} \in \mathbb{R}^{3\times1}$.
   If in your case $\boldsymbol{x}_{\boldsymbol{i0}} \in \mathbb{R}^{1\times3}$, you should transpose $\boldsymbol{x}_{\boldsymbol{i0}}$ before the matrix multiplication.

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
function gre2jd (assignment #2)

b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)

c) multiply the precession matrix with the Cartesian coordinate vector at J2000.0
$\boldsymbol{x_{iP}}(JD) = P(JD) \cdot \boldsymbol{x_{i0}}$; as $P \in \mathbb{R}^{3\times3}$, it must be $\boldsymbol{x_{i0}} \in \mathbb{R}^{3\times1}$.
If in your case $\boldsymbol{x_{i0}} \in \mathbb{R}^{1\times3}$, you should transpose $\boldsymbol{x_{i0}}$ before the matrix multiplication.

d) convert the resulting precessed Cartesian coordinate vector $\boldsymbol{x_{iP}}$ back to celestial coordinates function xyz2plr (assignment #3). Attention! Again mind the order of appearance of spherical coordinates.

GFZ Helmholtz Centre POTSDAM

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
function gre2jd (assignment #2)
b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)
c) multiply the precession matrix with the Cartesian coordinate vector at J2000.0
$\boldsymbol{x_{iP}}(JD) = P(JD) \cdot \boldsymbol{x_{i0}}$; as $P \in \mathbb{R}^{3\times3}$, it must be $\boldsymbol{x_{i0}} \in \mathbb{R}^{3\times1}$.
If in your case $\boldsymbol{x_{i0}} \in \mathbb{R}^{1\times3}$, you should transpose $\boldsymbol{x_{i0}}$ before the matrix multiplication.
d) convert the resulting precessed Cartesian coordinate vector $\boldsymbol{x_{iP}}$ back to celestial coordinates
function xyz2plr (assignment #3)
e) make sure alpha and delta are in the right units, i.e. alpha (hour angle), delta (degree)

Exercise 1

Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
   where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
   function gre2jd (assignment #2)
b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)
c) multiply the precession matrix with the Cartesian coordinate vector at J2000.0
   $\boldsymbol{x_{iP}}(JD) = P(JD) \cdot \boldsymbol{x_{i0}}$; as $P \in \mathbb{R}^{3\times3}$, it must be $\boldsymbol{x_{i0}} \in \mathbb{R}^{3\times1}$.
   If in your case $\boldsymbol{x_{i0}} \in \mathbb{R}^{1\times3}$, you should transpose $\boldsymbol{x_{i0}}$ before the matrix multiplication.
d) convert the resulting precessed Cartesian coordinate vector $\boldsymbol{x_{iP}}$ back to celestial coordinates
   function xyz2plr (assignment #3)
e) make sure alpha and delta are in the right units, i.e. alpha (hour angle), delta (degree)
f) store the results for each yyyy

Exercise 1

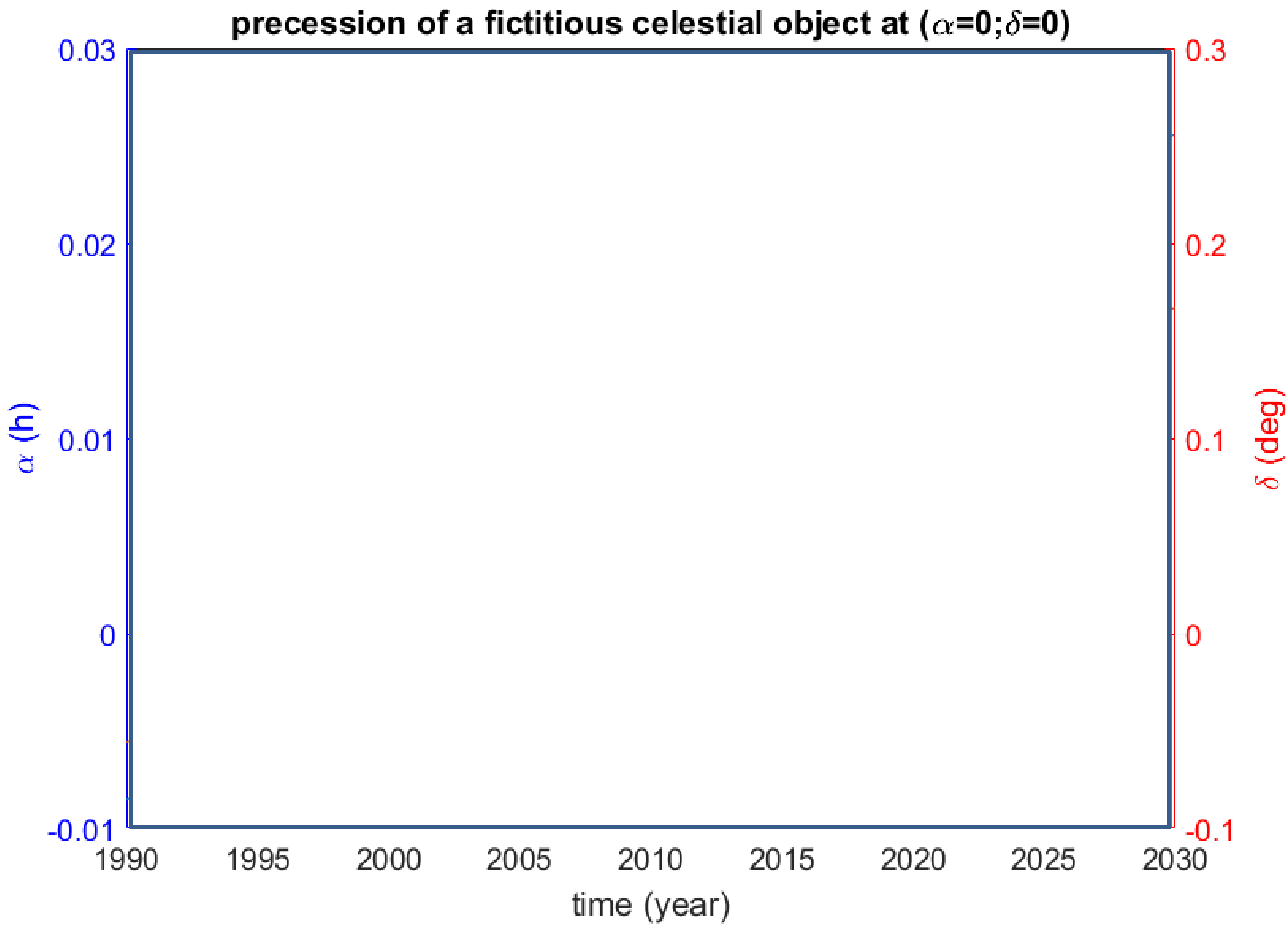
Evaluate your code by calculating the variation of the coordinates of a fictitious celestial object at the vernal equinox (0; 0) due to precession during the years 1990 to 2030.

The computation steps involve

a) compute JD for each `yyyy=1990:2030`
   where `mm=1; dd=1; ut1=12; minute=0; second=0;` is stationary
   function gre2jd (assignment #2)
b) compute the time-dependent precession matrix $P(JD)$ (function prec created before)
c) multiply the precession matrix with the Cartesian coordinate vector at J2000.0
   $\boldsymbol{x}_{\boldsymbol{iP}}(JD) = P(JD) \cdot \boldsymbol{x}_{\boldsymbol{i0}}$; as $P \in \mathbb{R}^{3\times3}$, it must be $\boldsymbol{x}_{\boldsymbol{i0}} \in \mathbb{R}^{3\times1}$.
   If in your case $\boldsymbol{x}_{\boldsymbol{i0}} \in \mathbb{R}^{1\times3}$, you should transpose $\boldsymbol{x}_{\boldsymbol{i0}}$ before the matrix multiplication.
d) convert the resulting precessed Cartesian coordinate vector $\boldsymbol{x}_{\boldsymbol{iP}}$ back to celestial coordinates
   function xyz2plr (assignment #3)
e) make sure alpha and delta are in the right units, i.e. alpha (hour angle), delta (degree)
f) store the results for each yyyy
g) plot alpha / delta vs. time (year)

Exercise 1

Plot the celestial coordinates vs. time in years. Example plot with `plotyy.`



Exercise 2

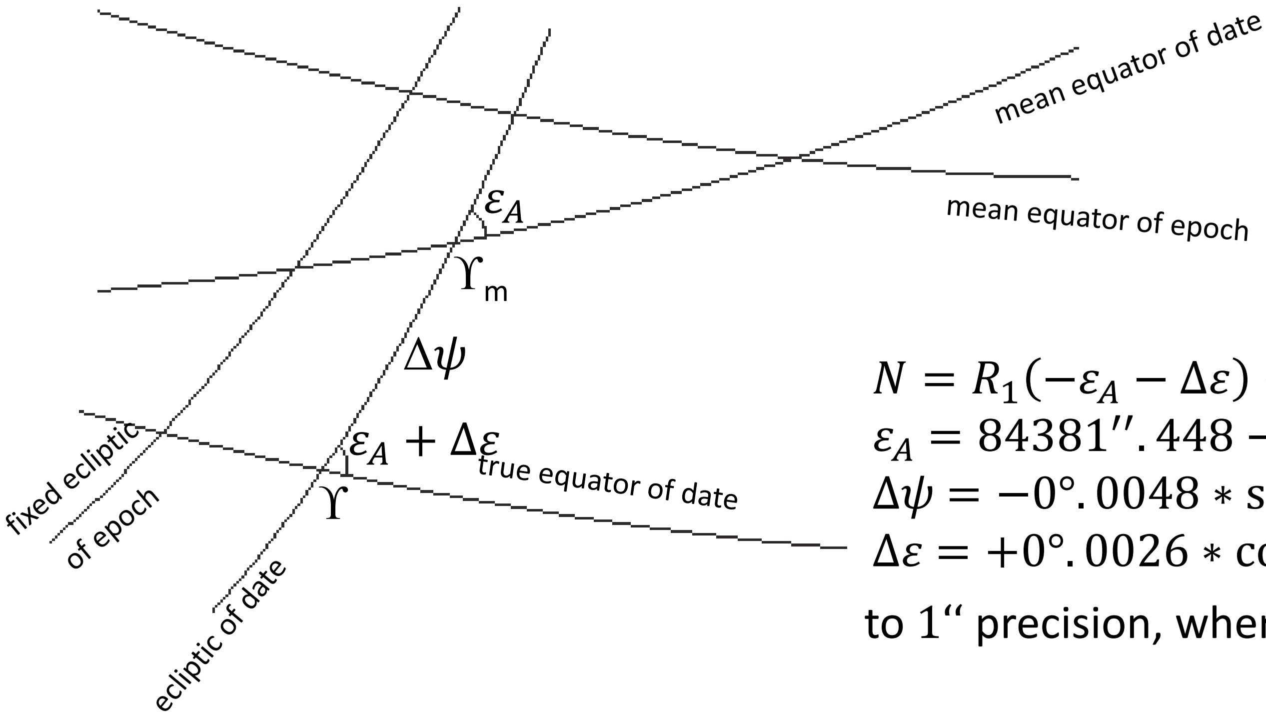
Write code that computes the nutation matrix ...

works in analogy to exercise#1 but with monthly resolution

Exercise 2

Nutations, numerical expressions based on J2000.0 (i.e. FK5)

$\Delta\varepsilon$: nutation in obliquity, $\Delta\psi$: nutation in longitude, ε_A: obliquity of the ecliptic at date



$$N = R_1(-\varepsilon_A - \Delta\varepsilon) \cdot R_3(-\Delta\psi) \cdot R_1(\varepsilon_A)$$
$$\varepsilon_A = 84381''.448 - 46''.8150 * T$$
$$\Delta\psi = -0°.0048 * \sin(f_1) - 0°.0004 * \sin(f_2)$$
$$\Delta\varepsilon = +0°.0026 * \cos(f_1) - 0°.0002 * \cos(f_2)$$

to 1“ precision, where

$$f_1 = 125° - 0°.05295 * d$$
$$f_2 = 200°.9 + 1°.97129 * d$$

$d = T * 36525$: time difference to J2000.0

(T see exercise#1)

Exercise 2

Evaluate your code by calculating the variation of the fictitious celestial object at the vernal equinox due to nutation during the years 1990 to 2030 (monthly resolution).

1) coordinates of a fictitious celestial object at the vernal equinox (0; 0):

$\alpha = 0;\ \delta = 0$

2) Celestial to Cartesian coordinate conversion

$\alpha, \delta \rightarrow \boldsymbol{x}_{i0}$

3) during the years 1990 to 2030

monthly resolution

nested loops for

`yyyy=1990:2030`

`mm=1:12;`

where `dd=1; ut1=12; minute=0; second=0;` is stationary

Exercise 3

Download the polar motion time series compatible with the IAU1980 theory during 2021. Write a function that outputs the polar motion matrix $W = R_2(-x_p) \cdot R_1(-y_p)$ for a given time. Use linear interpolation between the diurnal polar motion values.

Exercise 3

Download the polar motion time series compatible with the IAU1980 theory during 2021.

If not all values for 2021 are available, no problem work with what is available.

Use linear interpolation between the diurnal polar motion values.

Find the two adjacent diurnal values, e.g. for 2021-06-11

2021 6 11 59376 0.174245 0.435720 -0.1810931 -0.0008734 -0.108643 -0.010282 ...

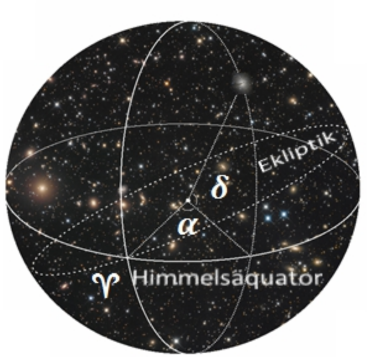
2021 6 12 59377 0.175463 0.435402 -0.1802017 -0.0009192 -0.108645 -0.010266 ...

and interpolate between the values (given at 00:00) linearly using the fraction of day.
Make sure you have the angles in the correct units

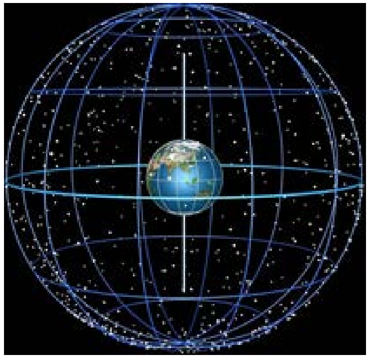
The MatLab command `interp1` could be helpful in this respect.

Exercise 4

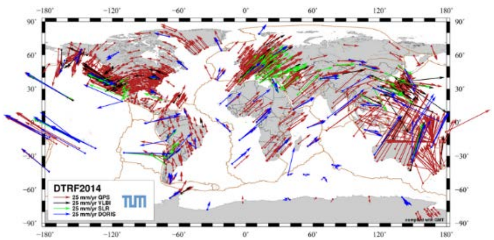
Barycentric celestial



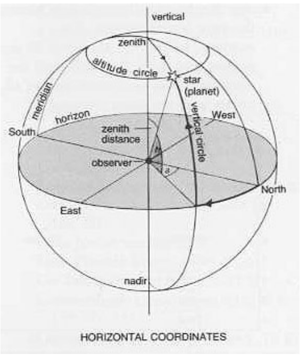
Geocentric celestial



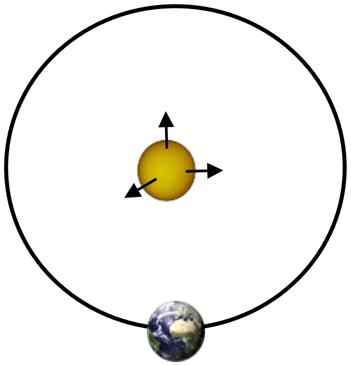
Geocentric terrestrial



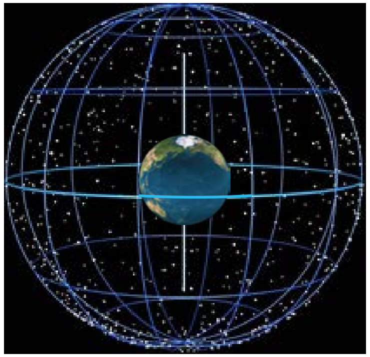
Local horizon



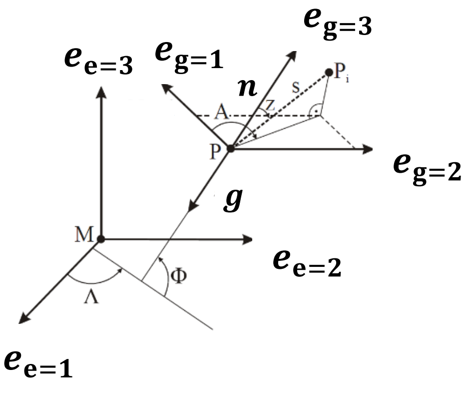
relativistic transformation



Earth orientation



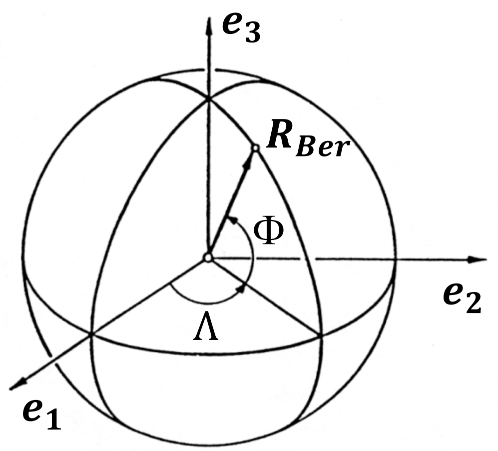
global → local orientation

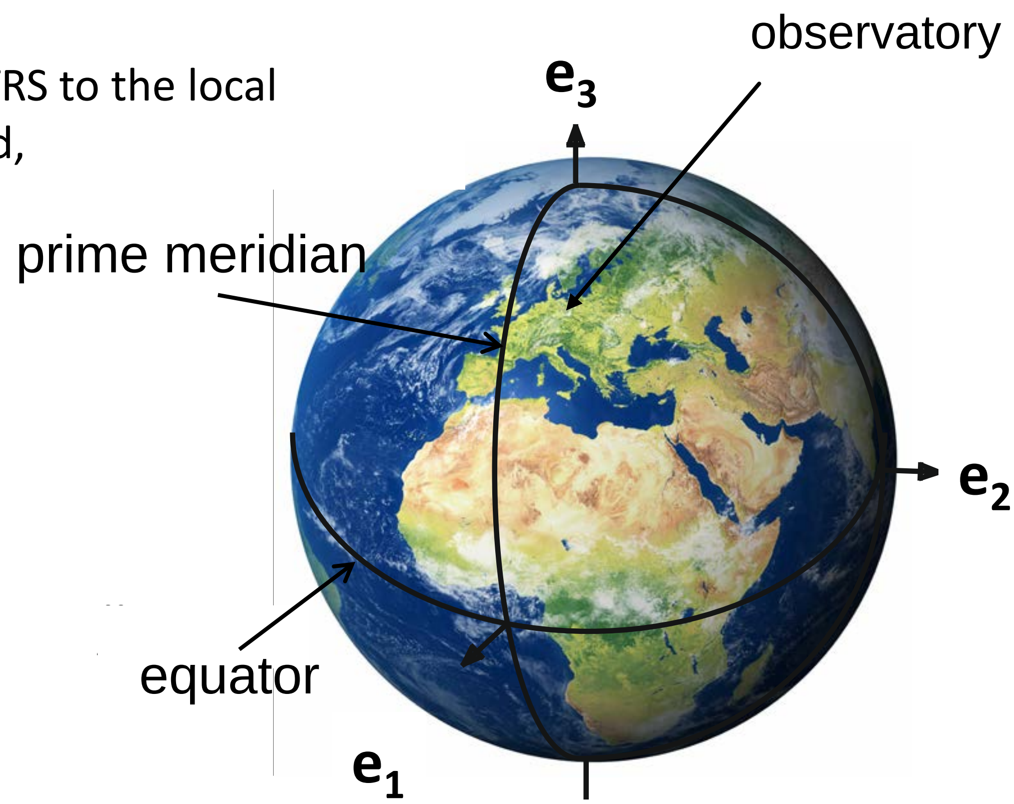


Transformation from ITRS to local horizon

After transformation from GCRS to ITRS, the position of the celestial objects is given in ITRS.

In a next step we need to transform from ITRS to the local horizon system of an observatory on ground, for which we know the astronomical longitude Λ and astronomical latitude Φ.



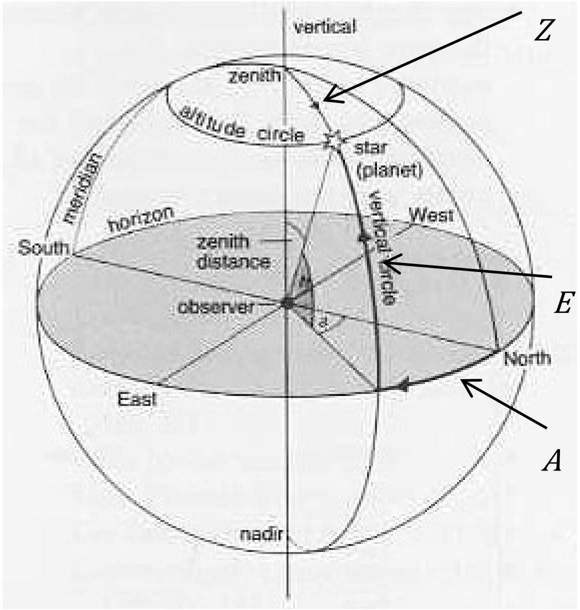


Local topocentric / horizon system

<u>Origin:</u> at the reference point of the instrument
(Φ, Λ, H), astronomical latitude Φ, astronomical longitude Λ, and physical height H

<u>Scale</u>
Proper time of observer (close to TT)

<u>Orientation, principle plain:</u>
The principle plain "**horizon**" is the plain normal to the vertical (in opposition to the local plumb line). The first axis points to the local **North** that is the intersection of the orthogonal projection of the Earth mean rotation axis on the principle plain. The third axis points along the **vertical** ("zenith") and the second axis completes a left-handed orthonormal basis (local **East**).



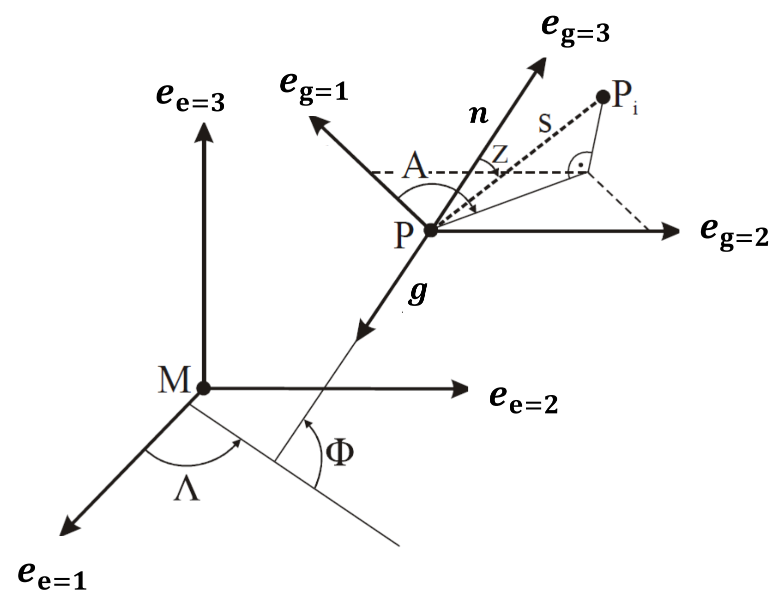
<u>Coordinates</u>:
Azimuth angle: $A \in [0; 360[$ positively counted clockwise from the north direction and
Elevation angle: $E \in [0; 90]$ or **Zenith Distance**: $Z = 90° - E$ referring to the local zenith.

<u>Orientation stability</u>: The orientation stability is practically maintained as long as the instrument is not shifted, tilted or replaced.

Transformation between the terrestrial system and the local horizon system

The true vertical (zenith direction) depends on the local plumb line $\boldsymbol{g}/\|\boldsymbol{g}\|$ and thus on the local gravity potential (W) at P:

$$\boldsymbol{g} = \operatorname{grad} W(\mathrm{P}) \approx -g\,\boldsymbol{n} = -g \begin{pmatrix} \cos\Phi\cos\Lambda \\ \cos\Phi\sin\Lambda \\ \sin\Phi \end{pmatrix}$$



where $\boldsymbol{n}$ denotes the outer normal of the equipotential surface through P expressed by astronomical coordinates.

The exact transformation requires information about the direction of $\boldsymbol{g}$ at P in the terrestrial system.

The change of the orientation

$\boldsymbol{x}_{\mathbf{g}} = M_1 R_2\left(\frac{\pi}{2} - \Phi\right) R_3(\Lambda)\boldsymbol{x}_{\mathbf{e}}$ is often approximated with geodetic coordinates ($\Phi \approx B$, $\Lambda \approx L$).

Where M_1 is a reflection matrix. And the (Cartesian to spherical) coordinate conversion is:

$E = \operatorname{atan}\left(\boldsymbol{x}_{\mathbf{g}}(3)/\sqrt{x_g(1)^2 + x_g(2)^2}\right)$, $A = \operatorname{atan2}\left(\boldsymbol{x}_{\mathbf{g}}(2)/\boldsymbol{x}_{\mathbf{g}}(1)\right)$, where atan2 denotes the 4-quadrant ready inverse tangent.

GCRS to local horizon system transformation at once

The vector in the local horizon system can be obtained through matrix multiplication with the vector in ITRS, which is in turn a function of the GCRS vector.

$$\boldsymbol{x}_g = M_1 \, R_2\left(\frac{\pi}{2} - \Phi\right) R_3(\Lambda) \boldsymbol{x}_e \;\; ; \qquad \boldsymbol{x}_e = W \; R \; N \; P \; \boldsymbol{x}_{GCRS}$$

or in one step:

$$\boldsymbol{x}_g = M_1 \, R_2\left(\frac{\pi}{2} - \Phi\right) R_3(\Lambda) \; W \; R \; N \; P \; \boldsymbol{x}_{GCRS}$$

where M_i denotes a mirror matrix and $R_i(\gamma)$ a rotation matrix around the i-th axis with the angle γ. The angles Λ and Φ are the astronomical longitude and latitude of the topocenter. W, R, N, P are the polar motion, Earth rotation, nutation and precession matrices.

Local horizon system

The Cartesian vector in the local horizon system can be converted to Azimuth and Elevation angles through Cartesian to spherical coordinate conversion or with

$$A = \mathrm{atan2}(x_{locS}(2)/x_{locS}(1))$$

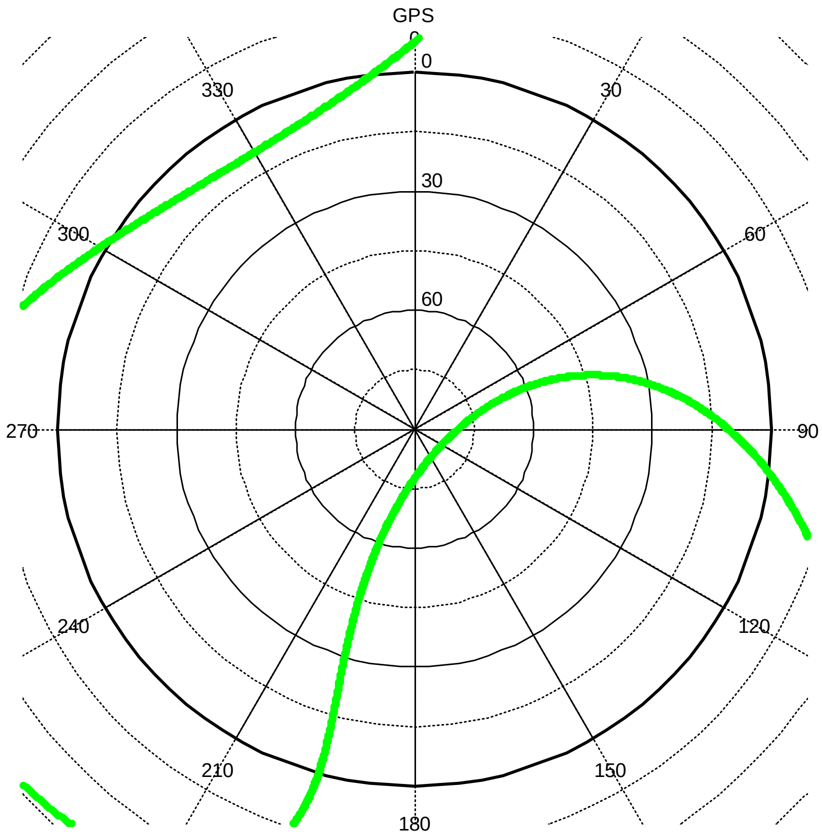
and

$$E = \mathrm{atan}\left(x_{locS}(3)/\sqrt{x_{locS}(1)^2 + x_{locS}(2)^2}\right)$$

where the inverse tangent (atan2) needs to be 4-quadrant ready.

Skyplot

A skyplot is a graphical representation useful for the understanding of visibility in the local horizon system (zenith, north, east).



the center of the spikes represents the local zenith

the solid line labeled “0” represents the local horizon

if the object is below 0 deg elevation it is below the horizon (obscured by the solid Earth) and thus not visible for the local observer

Note: this is an Earth orbiting satellite. Paths of stars look different!

Exercise 4

Given are the coordinates α and δ of four ICRF3 objects (attachment 1) in ICRF3 and the astronomical coordinates of an observatory near Berlin ($\Lambda = 13°24'$, $\phi = 52°36'$) in the body-fixed system. Compute the Azimuth A and elevation E to the celestial objects in the local horizon system of the observatory during one day (2021-06-11, entire day with resolution of one minute). Use the classical transformation $x_g = M_1\, R_2\left(\frac{\pi}{2} - \phi\right) R_3(\Lambda)\, W\, R_3(GMST)\, N\, P\, x_{i0}$, based on the equator of date, where M_i denotes a reflection matrix. $GAST$ shall be approximated with $GMST$. Use linear interpolation for polar motion, if indicated.

Exercise 4

entire day in minute resolution → running variable is time in minutes,

e.g. nested loops with

```
for ut1 = 0:23
for minute = 0:59
…
end
end
```

or with one loop only (or even better with vectors)

Exercise 4

Coordinates of radio sources:

1) Excerpt from ICRF3 catalog:

Source name	α [h min sec.dec.]	δ [deg min sec.dec.]
0454+844	05 08 42.36351222	84 32 04.5441733
1101−536	11 03 52.22168463	- 53 57 00.6966389
1111+149	11 13 58.69508613	14 42 26.9526507
1738+499	17 39 27.39049431	49 55 03.3683385

α is in hh:mm:ss whereas δ is in deg:min:sec. Multiply α with 360/24 = 15 to get α in deg.

Mind the negative sign for the delta of the second radio source. The negative sign applies for the entire coordinate, not just for the degree value: delta(2) = - (53 + 57/60 + …)

Get x, y, z from α, δ through available functions, e.g. assignment#2:

```
[xi0(1),xi0(2),xi0(3)] = plr2xyz(delta(1),alpha(1),1);
```

Attention! Mind the order of delta, alpha, and radius=1.

Exercise 4

The transformation has a time independent and a time dependent part:

$$x_{\mathrm{g}} = M_1\, R_2\left(\tfrac{\pi}{2} - \phi\right) R_3(\Lambda)\, W\, R_3(GMST)\, N\, P\, x_{i0}$$

Calculate everything that is time independent only once, i.e. before the loop(s) in case you work with loops.

Then within the loop calculate the time dependent parts:

P, N, and $GMST$ depend on time, you get them specifying the time in minute resolution. Do not work on the time scales for this assignment#4. Interpret time as realized in UT1 directly.

Realize the EOP in minute resolution.

W (polar motion matrix) must be interpolated linearly.

Get the polar motion values at 2021-06-11 (00:00:00) and one day later (same time).

Then do linear interpolation between the 00:00:00 values, e.g. with MatLab command `interp1`.

Finally apply the complete transformation to get x_{g} for each source at each minute.

Exercise 4

Convert x_g to Azimuth A and elevation E:

$$A = \mathrm{atan}\left(\frac{x_{g(2)}}{x_{g(1)}}\right)$$

$$E = \mathrm{atan}\left(\frac{x_{g(3)}}{\sqrt{x_{g(1)}^2 + x_{g(2)}^2}}\right)$$

For Azimuth, use the quadrant-checking version of atan: `atan2` (assignment#3)

Shift Azimuth from atan2 definition interval [-180; 180] to definition interval [0; 360] by adding a constant of 360 degree to negative Azimuth angles.

Time series plots:
For the time series plots, plot time (hour) on the horizontal axis vs. elevation / azimuth on the vertical axis. Elevation of zero and above indicates visibility of the object.

Skyplots:
Copy the attached MatLab code and save it as a function file in your local directory. Try skyplot with different line styles / markers / colors etc. Plot the 4 objects in one figure. Insert labels!

Assignment #4

Reused functions / code

spherical → Cartesian coordinate conversion
Gregorian calendar date → JD
3d rotation matrix
Cartesian → spherical coordinate conversion
(degree, minute, second) → degree.dec conversion
3d mirror matrix
computation of GMST for a given UT1 time and date
download and extract polar motion values from IERS

New functions / code

compute the precession angles and form precession matrix (exercise#1)
compute the nutation angles and form nutation matrix (exercise#2)
linear interpolation between adjacent polar motion values (exercise#3)
global → local transformation $M_1\ R_2\left(\frac{\pi}{2} - \phi\right)\ R_3(\Lambda)$
Cartesian → Elevation, Azimuth conversion, $A = \mathrm{atan2}\left(\frac{x_{g(2)}}{x_{g(1)}}\right)$, $E = \arctan\left(\frac{x_{g(3)}}{\sqrt{x_{g(1)}^2 + x_{g(2)}^2}}\right)$

comparable to Cartesian → spherical