# Meeting of March 17th, 2023

## Two subroutines

Additional comments in blue

C/A is the point for which $x = x0$ and $y = y0$ (parameters 31, and 32 in the *.csv*)

### 1 • Subroutine - orthoproj(loo,lao,lon,lat,x,y,z,fg)

```
!-------------------------------------!
! orthonormal projection
! -loo,lao coordinates at origin
! -lon,lat coordinates point
! -x,y,z coordinates in the Bessel plan
! -fg flag before the plan (1)
!          behind the plan (0)
!-------------------------------------!
```

#### List of variables and parameters

- inputs: real variables - entries
  - lon, lat
  - loo, lao (entries 81 and 82 in the *.csv*) – loo, lao are the coordinates at the centre of the map, for which $x = 0$ & $y = 0$).
- outputs
  - real variables: $x, y, z$
  - integer variable: fg
- other variables
  - real, scalars: la, lo, $la_0$, $lo_0$, cc, rho, th, thp, bb
  - real, vectors, dimension (3): xyz, U, V
  - integer: j

#### The subroutine

lo=lon
la=lat
$lo_0$=loo
$la_0$=lao

call iau_GD2GCE ( 1E0_wp, apf, lo, la, h, XYZ, J )

Transforms geodetic coordinates to geocentric for a reference ellipsoid of specified form, *e. g.* WGS84.
1E0_wp: Earth's equatorial radius
apf: Earth flattening
lo: longitude (radians, East +ve)
la: latitude (geodetic, radians)
h: height above ellipsoid (which you get from your routines), initially set at 0.0d0
**XYZ** is the output geocentric vector

if (j.ne.0) write(*,*) 'alert ',lo,la

J is a flag parameter that was added to alert in case there is an unphysical or arithmetic error

The projection is made through these 2 consecutive rotations
calling rotation function (input vector **xyz** from above,(the axis number about which the rotation is done, the angle of rotation (in radians), output three dimensional vector after rotation

call rotation(xyz,3,lo0,U)

call rotation(U,2,-la0,V)

The corresponding coordinates in the Bessel referential write (x, y, z).
If $z < 0$, the the pont is behind the Bessel plan (invisible)!

```
x=V(2)
y=V(3)
cc=V(1)
z=V(1)
```

Checking if the point is within the ellipsoid

```
th=pi*0.5d0-la0
thp=atan(tan(th)*(1E0_wp-apf)**2)
bb=pi*0.5d0-la0-thp
bb=-bb
```

final test, output flag (fg)

```
if (cc.lt.(y*tan(bb))) fg=0
if (cc.ge.(y*tan(bb))) fg=1
```

end subroutine

## 2 • SUBROUTINE - ORTHOPROJINV(LOO,LAO,X,Y,LON,LAT,FG)

(This is the inverse projection)

```
!————————————————————————————————!
! orthonormal projection
! -loo,lao coordinates at origin
! -lon,lat point coordinates
! -x,y coordinates on the Bessel plan
!————————————————————————————————!
```

### List of variables and parameters

- inputs: real variables - entries
  - loo, lao (entries 81 and 82 in the *.csv*)
  - $x$, $y$
- outputs
  - real variables: lon, lat
  - integer variable: fg
- other variables
  - real, scalars: la, lo, la0, lo0, cc, rho, th, thp, bb, hh, xx, yy, zz, r2, b2
  - real, vectors, dimension (3): xyz, U, V
  - integer: i, j

### The subroutine

```
lo0=loo
la0=lao
```

check if the point is on Earth's surface

```
b2=1E0_wp+(apf**2-2*apf)*cos(la0)**2 r2=sqrt(x**2+y**2/b2)
```

if the point is not on Earth's surface

```
if (r2.gt.1E0_wp) then
! write(*,*) 'not on Earth'
lon=0d0
lat=100d0*enrad
fg=0 (you don't care about this case, as it is outside of Earth)
else

fg=1 cc=sqrt(1E0_wp-x**2-y**2)
V(1)=cc V(2)=x V(3)=y
```

The projection is made through these 2 consecutive rotations.

```
call rotation(V, 2, la0, U)
call rotation(U, 3, -lo0, XYZ)
call iau_GC2GDE ( 1E0_wp, apf, xyz, lon, lat, hh, j )
```

(Transform geocentric coordinates to geodetic for a reference ellipsoid of specified form.)
1E0_wp: Earth's equatorial radius
apf: Earth flattening
**XYZ** is the input geocentric vector
lo: longitude (radians, East +ve)
la: latitude (geodetic, radians)
h: height above ellipsoid

```
i=0
do while ((abs(hh).ge.1e-8).and.(i.le.1000))
i=i+1
if (i==1000) write(*,*) 'i1000 orthoprojinv'
call orthoproj(loo, lao, lon, lat, xx, yy, zz, fg)
V(1)=zz
V(2)=x
V(3)=y
call rotation(V,2,la0,U)
call rotation(U,3,-lo0,XYZ)
call iau_GC2GDE (1E0_wp, apf, xyz, lon, lat, hh, j)
enddo
endif
```

## Constants

Earth's equatorial radius: 6378.137 km

Earth's flattening: 0.003352810665.

In our case, we set the equatorial radius to 1, and express $x$ and $y$ in these units.

# Additional info

Basically you write

$$x = x0 + xp * (t - t0)$$

$$y = y0 + yp * (t - t0)$$

The parameters are x, y, xp, yp, are 31, 32, 33, 34, respectively in the *.csv*.

**Reminder:** Access to the csv database, the associated header is given here.