Nicolaus Copernicus University
Centre for Astronomy, Department of Radio Astronomy
ul. Gagarina 11, PL 87-100 Toruń, Poland
`http://www.astro.uni.torun.pl/?lang=en`

POLGRAW, VIRGO Collaboration

Top2Bary — Software for Precise Reference of Detectors to Solar System Barycenter

Document Number: **VIR-735A-11**
Issue: 1.0
Date: December 19, 2011
Author: Kazimierz M. Borkowski

Abstract: *This document describes a set of FORTRAN procedures, a* `Top2Bary` *module, worked out in the years 2000 to 2011 for referencing of terrestrial astronomical observations in general, and of data collected from gravitational wave detectors in particular, to the Solar System Barycenter. The referencing relies on calculation with very high precision of the position and velocity of the observatory location with respect to the Earth barycenter and position and velocity of this center with respect to Solar System mass center as embodied in the JPL Planetary and Lunar Ephemerides, DE405/LE405. These positions and velocities are expressed in the rectangular equatorial coordinates of the ICRS, which correspond to the reference frame defined by the equator and equinox of the standard epoch of J2000.0. Algorithms employed, program structure and its usage are detailed. Also, presented with particulars are three practical programs which call the module to obtain the location barycentric coordinates and a correction for time delay (*`SSB2Det` *and* `SSB2DetC`*), and the Earth position on its orbit (*`Orb`*).*

VIRGO Collaboration
EGO — Via E. Amaldi – I-56021 S. Stefano a Macerata, Cascina (Pisa)
Secretariat: telephone (39) 050 752 511 – fax (39) 050 752 550 – e-mail `perus@ego-gw.it`

Table of contents

1. Introduction

Programs collected into the `Top2Bary` package or module, were originally developed in FORTRAN by a team consisting of Pia Astone, Piotr Jaranowski, Andrzej Królak, Maciej Pietka and me for analyses of high quality data gathered from resonant bar gravitational wave detector named EXPLORER, which was built and is operated by the Gravitational Wave Detector group at the National Institute for Nuclear Physics – Frascati National Laboratories (Astone *et al.* 2002, 2003 and 2005; see also `www.astro.uni.torun.pl/~kb/AllSky/Explorer/SearchE.htm` and `www.astro.uni.torun.pl/~kb/AllSky/Docs/Top2B2.htm`), and later for similar analyses of data from Nautilus and Virgo detectors (Astone *et al.* 2006, 2008 and 2010).

The `Top2Bary` module when called with given time and location coordinates returns four vectors of position and velocity of the location with respect to the Earth barycenter and of this barycenter with respect to the Solar System Barycenter (SSB). The sum of these two vectors in position and two in velocity represent the location position and velocity referred to the SSB. Thus `Top2Bary` is quite general and as such could be used for a variety of other scientific purposes. This report describes algorithms and procedures incorporated into it.

Over the years, since its first version of 2001, `Top2Bary` has been considerably improved and developed. While that first version has been created in Poland, our Italian part has developed similar but completely independent tool, the `PSS_astro` package (Astone 2003), based on the NOVAS (Naval Observatory Vector Astrometry Subroutines) package (Kaplan 1989). A comparison of outputs from both these tools allowed us to fix their bugs with the result that differences now remain at the level of round-off errors.

Furthermore, making extensive use of the SOFA package (Wallace 2004) we have *ad hoc* composed yet two programs (`T2C2kA` and `T2C2kB`, see Appendix), based on a new International Astronomical Union (IAU) nutation-precession theory and reference systems officially implemented in astronomy and space geodesy since 2003 (Capitaine *et al.* 2002, McCarthy & Petit 2003, Borkowski 2003, Wallace 2004), with the purpose of checking our geocentric outputs against those obtainable from the best and most recent theories and algorithms available. We have found that our rectangular celestial coordinates are in agreement with the new system below 3 cm level in each coordinate and 6 cm in absolute position displacement (Fig. 1). These offsets correspond to 1 to 2 mas (milliarcseconds) in spherical coordinates. This good agreement could have been achieved only after supplementing our packages with such fine corrections as for the celestial pole offsets and the equation of equinoxes. The mentioned 6 cm satisfactorily compare with the accuracy inherent in interpolated positions of the JPL ephemerides, which is said to be not worse than 1 m (for the Moon) to 25 m (for inner planets) or 1 mas. We note also that the new simplified IAU theory of nutation, NU2000B, seems to be of the same order of accuracy (see Fig. 3).

The present version, 3.0, of the `Top2Bary` module and practical examples are those of December 2011, and from the user point of view differ from the version 2.1 (of February 2006, described on the web page `/www.astro.uni.torun.pl/~kb/AllSky/Docs/Top2B2.htm`) in the following aspects:

- `Top2Bary` can handle UTC or TT on input. The `sitePV` subroutine subtracts 1 s from UT1 – UTC if its new argument `leap` equals 1. This allows for dealing with the leap second itself.
- The `Top2Bary` module now can use other JPL ephemerides, notably DE200.
- The `state` subroutine was modified to accept JPL binary data with and without headers, and to work correctly when compiled with `-dbl` and `-quad` precisions (8-byte REALs and 16-byte DP) offered by the Lahey-Fujitsu Fortran LF95 v.7.2 compiler.
- The `SSB2Det`, `SSB2DetC` and `Orb` programs can accept MJD argument (their first argument on the command line) based on the ET/TDT/TT and GPS time instead of UTC. This allows

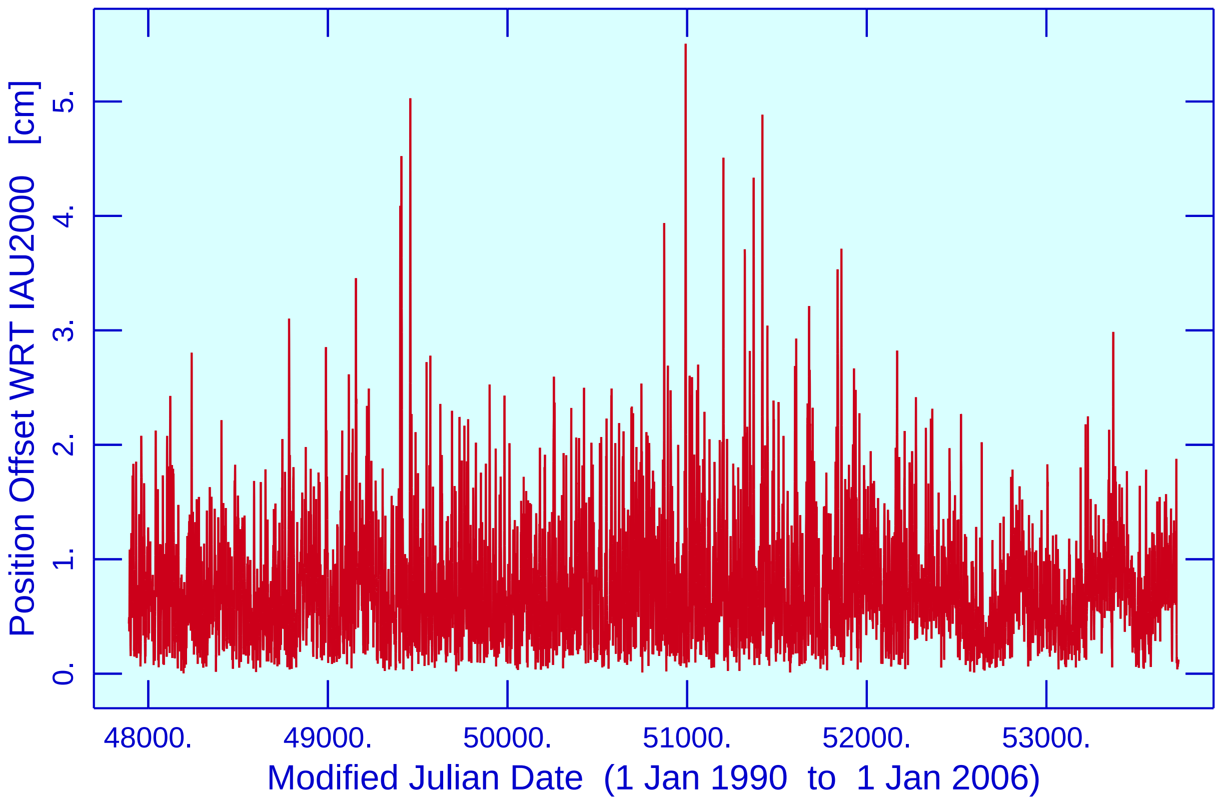

Fig. 1. Absolute difference between the Explorer location referred to the geocentric ICRF computed with two packages: the `Top2Bary` and `T2C2kA`, the latter incorporating full precision software based on the 1997–2000 IAU astronomical reference systems, time scales, and Earth rotation models. The data plotted were sampled every 14 UTC hours over 16 years (10019 points starting at MJD = 47892.0). The differences in x, y and z coordinate over 1990.0 – 2006.0 averaged to 0.07, −0.14 and 0.00 cm, respectively, with overall rms of only 0.96 cm and the global absolute maximum of 5.50 cm at MJD = 50991.8333 (to get these estimates a denser set of 70130 samples spaced 2 hours has been generated and analysed).

for working in and across leap seconds.
• The `SSB2Det` program generates one more output file (`phir.dat`) containing the local apparent sidereal time of each sample.
• The `polmot` subroutine was modified to accept also the new format of data in `eopc04.yy` files.

Enabling the quadruple precision allows for time argument to be specified with precision well below 1 ns, but it requires that this version should be compiled with the compiler's quadruple precision enabled. The higher precision gives slightly different results than we had earlier, but the differences are all within limits of accuracy of these computations.

2. Overview of transformations involved

Remarks on reference systems and frames

These remarks are based on recent recomendations of IAU (Kaplan 2005).

Reference data for positional astronomy, such as the data in barycentric planetary ephemerides, are now specified within the International Celestial Reference System (ICRS). The ICRS is a coordinate system whose origin is at the solar system barycenter and whose axis directions are effectively defined by the adopted coordinates of 212 extragalactic radio sources which are assumed to have no observable intrinsic angular motions. Thus, the ICRS is a space-fixed system (more precisely, a kinematically non-rotating system) without an associated epoch. However, the ICRS closely matches the conventional dynamical system defined by the Earths mean equator and equinox of J2000.0; the alignment difference is at the 0.02 arcsecond level, negligible for many applications. The list of radio source positions that define ICRS for practical purposes is called the International Celestial Reference Frame (ICRF).

The position and velocity 3-vectors taken from the JPL DE405/LE405 ephemeris are in equatorial rectangular coordinates referred to the solar system barycenter. The reference frame for the DE405 is the ICRF; the alignment onto this frame, and therefore onto the ICRS, has an estimated **accuracy of a few milliarcseconds**, at least for the inner-planet data.

The DE405 was developed using T_{eph}, a barycentric coordinate time (Standish 1998). T_{eph} is rigorously equivalent to Barycentric Coordinate Time (TCB) in a mathematical sense, differing only in rate: the rate of T_{eph} matches the average rate of TT (Terrestrial Time, or TDT), while the rate of TCB is defined by the SI system. The IAU time scale, Barycentric Dynamical Time (TDB), often (but erroneously) considered to be the same as T_{eph}, is a quantity that cannot be physically realized, due to its flawed definition. So, in fact, the use of the name TDB actually refers to quantities based on or created with T_{eph} (because of this, the IAU Working Group on Nomenclature for Fundamental Astronomy has recommended changing the definition of TDB to be consistent with that of T_{eph}). Astronomical constants obtained from ephemerides based on T_{eph} (or TDB) are not in the SI system of units and must therefore be scaled for use with TCB or other SI-based time scales.

J2000.0 is the epoch 2000 January 1, 12^h TT (JD 2451545.0 TT) at the geocenter (J2000.0 system is shorthand for the celestial reference system defined by the mean dynamical equator and equinox of J2000.0). The coordinate system defined by the equator and equinox of J2000.0, can be thought of as either barycentric or geocentric.

It is also worth noting that the recent IAU resolutions do not describe the proper reference system of the observer — the local, or topocentric, system in which most measurements are actually taken. The resolutions as adopted apply specifically to Einstein's theory of gravity, i.e., the general theory of relativity.

Time scales

It is assumed that the time, associated with observational data we are dealing with, is the Coordinated Universal Time (UTC) as disseminated by international time services. The UTC scale since 1972 is essentially uniform, except for occasional 1 second steps (leap seconds) introduced internationally to compensate for the variable Earth rotation. By 1 January 2009 there were 34 leap seconds. Earlier, 1961 to 1971, the adjustments were continuous. UTC devoid of these adjustments is called the International Atomic Time, TAI. The TAI − UTC differences are available in tabular form at `hpiers.obspm.fr/eop-pc/earthor/utc/TAI-UTC_tab.html`. Our `tai_ut` function reads similar file and returns the difference calculated for given UTC Julian Date. So, this difference added to UTC converts it to TAI which is uniform. TAI in turn differs from the Terrestrial (Dynamical) Time, TDT or TT (normally used to describe celestial phenomena by astronomers), only by a constant term:

$$\mathrm{TDT} = \mathrm{TAI} + 32.184\,\mathrm{s}.$$

In principle, the time argument of the JPL positions and velocities of celestial objects is T_{eph} or the barycentric coordinate time. This time scale in practice can be equated with the

Barycentric Dynamical Time, TDB (in spite of already noted inadequacy in definition). The TDB differs from the TDT only by small periodic terms, which to sufficient accuracy are usually simplified to only two largest terms:

$$\mathrm{TDB} - \mathrm{TDT} = 0.001658 \sin(g) + 0.000014 \sin(2g) \text{ seconds},$$

where $g = 357.53 + 0.9856003$ (JD $-$ 2451545.0) degrees, and JD is the Julian Date equal to MJD + 2400000.5, MJD being the Modified Julian Date. So essentially, for many practical purposes, the two scales do not have to be distinguished and could be assumed equal, TDB = TDT (this possibility can be made effective in our procedures by setting the `iTDB` option, in the `useTop2B.cfg` configuration file, to 0 or 2).

One can thus relate given UTC with barycentric positions of all the major celestial bodies of the Solar System as obtainable from the JPL ephemeris. These relations are incorporated into the `Top2Bpv` subroutine, which also calls the `tai_ut` function that reads a file with the UTC adjustments.

However, to relate the position of a point on the Earth to the geocentric ICRF (i.e. the same frame as the barycentric ICRF except for the origin of axes which is now at the barycenter of the Earth) one has to use yet another time scale — the rotational time scale UT1, which is nonuniform and is determined from astronomical observations. The difference UT1 − UTC, the value of which is presently maintained by international services within ± 0.90 s, is taken from the IERS tabulations of daily values available as `eopc04.yy` files, where `yy` stands for a two digit year number (e.g. `99` for 1999, and `11` for 2011), at the IERS Internet site `hpiers.obspm.fr/iers/eop/eopc04`. The UTC to UT1 conversion takes place in the `sitePV` subroutine, which calls a polar motion routine, `polmot`. The latter returns the UT1 − UTC difference interpolated (between nearest midnight values read from proper `eopc04` file) to the UTC given, along with other Earth-axis motion parameters (terrestrial and celestial pole offsets) similarly interpolated. The UT1 time can be readily converted to the Earth rotational angle or the sidereal time. This last conversion is made in the `sid` function, which is a function of UT1 and the location geographical longitude corrected for the polar motion.

Location coordinates and velocities with respect to Earth barycenter

To be able to express coordinates of a point on the Earth surface in the Earth centered ICRF it is necessary to know orientation of the Earth in space. The primary effects that should be taken into account are: diurnal (variable) rotation, precession and nutation of the Earth rotational axis and polar motion. The precession is accounted for by applying standard astronomical theories. We use the new IAU theory (Lieske 1979). The nutation could be also computed basing on a theory, but the DE405 has it in numerical form, so we just read the nutation angles, $\Delta\psi$ and $\Delta\varepsilon$. These nutation angles are not the same as defined in the newest IAU nutation theory, so when highest precision is required the celestial pole offsets, $\mathrm{d}\psi$ and $\mathrm{d}\varepsilon$, must also be added to these angles (in the past the magnitude of these offsets remained below 0.1 arcsecond). For past years, since 1962, these two offsets are included in the already mentioned `eopc04.yy` files. The remaining two effects, the Earth variable rotation and polar motion, are unpredictable for a longer future, so also observational data must be used. The data necessary for reduction are taken from the `eopc04.yy` files as well.

The polar motion can be taken into account by modifying the conventional geographical coordinates of a point on the Earth (see Eqs. 5.1 in Astone *et al.*, 2002) or rectangular coordinates corresponding to these conventional geographical coordinates:

$$\begin{aligned} x_o &= r_o \cos \lambda_\circ, \\ y_o &= r_o \sin \lambda_\circ, \quad \text{and} \\ z_o &= b \sin \psi + h \sin \phi_\circ, \end{aligned} \tag{1}$$

where $\phi_\circ$ is the conventional geographical latitude, $\lambda_\circ$ — the conventional longitude, h — the height above the Earth ellipsoid, $\psi = \arctan(b \tan \phi_\circ / a)$ — the reduced latitude, $r_o = a \cos \psi + h \cos \phi_\circ$ is the equatorial component of the radius vector, and $a = 6378.140\,\mathrm{km}$ and $b = a(1-1/f)$ are the semiaxes of the ellipsoid (here the flattening f is taken equal to 298.257 or 1/0.00335281, which are the IAU or NOVAS value, respectively). We have adopted the latter possibility (i.e., modification of the location x, y and z coordinates) using the following relations:

$$\begin{aligned} x &= x_o - P_x z_o \\ y &= y_o + P_y z_o \\ z &= z_o + P_x x_o - P_y y_o \end{aligned} \tag{2}$$

where P_x and P_y are the IERS coordinates of the pole, with respect to the Conventional International Origin (to which refer the 'conventional' geographical coordinates), converted to radians.

These (x, y, z) coordinates are expressed in the terrestrial reference frame with the x axis directed toward the Greenwich meridian. To relate them to the celestial frame, the ICRF, the rotational angle of the Earth must be taken into account. This is done through conversion of UTC to UT1 (as described in the preceding section). The UT1 serves to calculate the apparent (or true) local sidereal time θ (returned by the `sid` function), which includes the nutational component (nutation in longitude corrected for the corresponding IERS celestial pole offset, also taken from the IERS `eopc04` files). The apparent local sidereal time is advanced with respect to the mean Greenwich time by the true location longitude (calculated as $\lambda = \arctan(y/x)$ with proper choice of one of the four quadrants) and the mentioned nutational component. This component is equal to the so called equation of equinoxes $(\Delta\psi + d\psi)\cos\varepsilon$ plus a very small correction to this equation (which amounts to less than $0.003''$ and depends on the mean longitude of the ascending node of the Moon). Our configuration file allows the user to neglect this small correction or even use the mean sidereal time instead of apparent one. So computed local sidereal time is finally used to find rectangular coordinates of the location in the IERS celestial reference frame:

$$\begin{aligned} X &= r_e \cos \theta, \\ Y &= r_e \sin \theta, \\ Z &= z, \end{aligned} \tag{3}$$

where $r_e = \sqrt{x^2 + y^2}$ is the equatorial component.

At this point the location velocity due to Earth rotation can be computed. The location motion relative to the Earth barycenter is represented by a vector of constant length (in principle $v_\circ = 2\pi r_e$ per sidereal day, or $v_\circ = \Omega r_e$) and directed always towards the East in the topocentric reference frame. This diurnal velocity vector has the following Cartesian components:

$$\begin{aligned} V_x &= v_\circ \cos(\theta + \pi/2) = -v_\circ \sin \theta, \\ V_y &= v_\circ \sin(\theta + \pi/2) = +v_\circ \cos \theta, \\ V_z &= 0, \end{aligned} \tag{4}$$

where the numerical value of v_o in our program is calculated with the NOVAS constant of the Earth angular rotation speed, $\Omega = 2\pi/(\text{sidereal day in TAI seconds}) = 7.2921151467 \times 10^{-5}$ rad/s. This constant corresponds to `NOmega` (a parameter listed in the configuration file) set to 1, and can be changed to the IERS value of 7.292115×10^{-5} (`NOmega` = 0) or to $2\pi/(24 \times 3600) \times 1.002737909350795 = 7.2921158553 \times 10^{-5}$ (`NOmega` = 2).

All the above conversions are done in the `sitePV` subroutine, whose complete listing is given below.

```
      subroutine sitePV(sCJD,leap,Xlat,Ylong,Zheigh,xyz)

c This subroutine calculates Cartesian coordinates and velocity of an
c observer at given geographical location with respect to the Earth
c barycenter in the reference frame of date |sCJD| (and not J2000.0).
c sCJD - (signed) Julian Date (UTC time scale)
c leap - if assigned 1 causes subtraction of 1 s from IERS UT1-UTC value;
c        this ensures correct outputs (xyz vector) during leap seconds
c Xlat, Ylong, Zheigh - geodetic coordinates (rad) and altitude (m);
c  if sCJD<0, these are the Cartesian coordinates Xlat=X, Ylong=Y, Zheigh=Z (km)
c xyz - 6-vector of Earth-centered rectangular coordinates (km) and velocity (km/s)

c Subroutines called:
c  polmot(CJD,Px,Py,UT1_C,dpsi,deps) - returns polar motion, UT1-UTC and offsets
c  sid(dj1,along)   - calculates local sidereal time (apparent or mean)

      implicit real*8 (a-h,o-z)
      dimension xyz(6)
      data a,frec/6378.140d0,298.257d0/
     & ,pi/3.14159265358979323846264338327950d0/
      common /options/ iUT1_C,iPM,iTDB,NutSid,NutRem,NovF,NOmega,iSunV
      common /nuta/ pnut(4)

      CJD = dabs(sCJD)

c If appropriate, convert geographical to rectangular coordinates
        if(sCJD.gt.0d0) then                ! angular coordinates given
      deg = pi/180d0
      b = a - a/frec
      if(NovF.eq.1) b = a - a*0.00335281d0   ! NOVAS value
      psi = datan(b*dtan(Xlat*deg)/a)
      ro = a*dcos(psi) + Zheigh*1d-3*dcos(Xlat*deg)
      along = Ylong*deg
      zo = b*dsin(psi) + Zheigh*1d-3*dsin(Xlat*deg)
      xo = ro*dcos(along)
      yo = ro*dsin(along)
        else                                ! Cartesian coordinates given
      xo = Xlat
      yo = Ylong
      zo = Zheigh
        endif

c Polar motion data: Px,Py  (from 'EOPC04.yy' file)
      call polmot(CJD,Px,Py,UT1_C,ddpsi,ddeps)
      if(leap.eq.1) UT1_C = UT1_C - 1d0 ! leap second removal from UT1-UTC
```

```
            if(NutSid.eq.2) then
c ddpsi and ddeps correspond to the celestial ephemeris pole (CEP) offsets;
c EOPC04.yy data and IERS subroutine give similar (but different) values
c     ddpsi = CEPIERS96(CJD,1)*1d-3  ! max. diff. over 10 years (1991-2001)
c     ddeps = CEPIERS96(CJD,2)*1d-3  !       is ~73 cm
      pnut(1) = pnut(1) + ddpsi*pi/(3600*180)
      pnut(2) = pnut(2) + ddeps*pi/(3600*180)
            endif

      conv=1d0/206264.8062470963551564733573307790D0
c Cancel out the Earth orientation parameters, if requested
            if(iUT1_C.eq.0) UT1_C = 0
            if(iPM.eq.0)    conv  = 0
      Px = Px*conv
      Py = Py*conv

c Compute observer's vectors with respect to Earth barycenter;
C wobble rotation follows
      x = xo - Px*zo
      y = yo + Py*zo
      zo = zo + Px*xo - Py*yo
      ro = dsqrt(x*x + y*y)
      along = datan2(y,x)        ! longitude corrected for polar motion

      dj1 = CJD + UT1_C/86400d0  ! 'Rotational' Julian Date
      alst = sid(dj1,along)      ! local sidereal time

      xyz(1) = ro*dcos(alst)     ! Site position X component
      xyz(2) = ro*dsin(alst)     ! Site position Y component
      xyz(3) = zo                ! Site position Z component

      vo = ro*7.292115d-5        ! mean Earth rotation rate (IERS 1992/2000)
      if(NOmega.eq.1) vo = ro*7.2921151467d-5 ! NOVAS value [rad/s]
      if(NOmega.eq.2) vo = 2*pi*ro/(24*3600)*1.002737909350795d0 != ro*7.2921158553d-5

      xyz(4) = -vo*dsin(alst)    ! Site velocity Vx component
      xyz(5) = +vo*dcos(alst)    ! Site velocity Vy component
      xyz(6) = 0d0               ! Site velocity Vz component

      end
```

Since our approach is essentially classical these Cartesian coordinates (X, Y, Z) and velocities (V_x, V_y, V_z) are naturally referred to the frame of equator and equinox of date. Therefore they are further nutated and precessed (in this order) back to the standard epoch J2000.0, and this is performed in the `Top2Bpv` subroutine by calling the `RemNut` and `prexyz` procedures, separately for the position vector and velocity vector.

Barycentric coordinates of the Earth (JPL ephemeris)

For computing the coordinates of the Earth barycenter, relative to the Solar System barycenter, use is made of the fundamental Solar System ephemerides from the Jet Propulsion Laboratory (JPL). The latest JPL Planetary and Lunar Ephemerides, "DE405/LE405" or just "DE405," were created in 1997 and are described in detail by Standish (1998). They represent an improvement over their predecessor, DE403, and are available via the Internet (anonymous ftp: `ssd.jpl.nasa.gov/pub/eph/planets/ascii/de405/`) or on CD-rom (from the publisher: Willmann-Bell, Inc.; `www.willbell.com/software/jpl.htm`).

As mentioned, the DE405 ephemeris is based upon the ICRF (an earlier popular ephemeris DE200, which has been the basis of the *Astronomical Almanac* since 1984, is within 0.01 arcseconds of the ICRF frame). It constitutes of a set of Chebyshev polynomials fit with full precision to a numerical integration over 1600 AD to 2200 AD. Over this interval the interpolating accuracy is no worse than 25 meters for any planet and no worse than 1 meter for the Moon.

The JPL package allows a professional user to obtain the rectangular coordinates of the Sun, Moon, and nine major planets anywhere between JED (i.e. Julian Ephemeris Date) 2305424.50 (1599 DEC 09) to JED 2525008.50 (2201 FEB 20). Besides coordinates it includes nutations and librations. Our routines do make use of the JPL nutation in longitude and in obliquity after correction for (addition of) the IERS celestial pole offsets.

The ephemeris gives separately the position and velocity of the Earth–Moon barycenter and the Moon's position and velocity, relative to this barycenter. The Earth position and velocity vectors are thus calculated as a fraction (involving the masses of the two bodies) of the Moon's ones and opposite to the latter. For example, the x-coordinate of Earth barycenter with respect to the SSB is obtained as:

$$x_{\mathrm{Earth}} = x_{\mathrm{EM}} - x_{\mathrm{M}}/82.30056,$$

where EM and M subscripts refer to the Earth–Moon barycenter and the Moon, respectively, and the numerical denominator is equal to the Earth–plus–Moon to Moon ratio of masses taken from the JPL ephemeris.

Since the DE405/LE405 coordinates are given in the J2000.0 reference frame the position and velocity of the Earth barycenter so obtained need not be nutated nor precessed.

Finally, the velocity vector of the Sun motion towards its apex (with the speed of 20 km/s) can be optionally added to the Earth barycenter velocity. The direction of solar apex is assumed at 18^{h} in right ascension and 30° in declination in the frame of equator and equinox of J1900.0. Therefore this direction is first precessed from that epoch to J2000.0. Since catalogue sky positions in astronomy are not corrected for the apex motion this component is actually only optionally included in the presented programs (corresponding `iSunV` option is set to 0 in the code). If desired, the `iSunV` option can be changed by the user just by setting a nonzero value in the configuration file, `useTop2B.cfg`. In this case the velocity vector towards the apex is added to the Earth velocity vector (the position remaining unaffected).

The position and velocity vectors of the Earth are computed by calling the `EarthPV` subroutine, which in turn calls the original JPL `STATE` routine slightly modified for our needs. This subroutine reads and interpolates the JPL planetary ephemeris from the file named `DE405_80.39`.

3. The `Top2Bary` module

The described algorithms were implemented in an easily callable subroutine package or module named `Top2Bary` consisting of about 900 lines of FORTRAN code. The module consists of the following subprograms:

`Top2Bpv` — main subroutine
`init` — reads the `useTop2B.cfg` file and sets some parameters and constants
`sitePV` — computes location geocentric vectors
`EarthPV` — computes Earth barycentric vectors
`STATE` — modified JPL STATE that reads the DE405
`INTERP` — original JPL subprogram

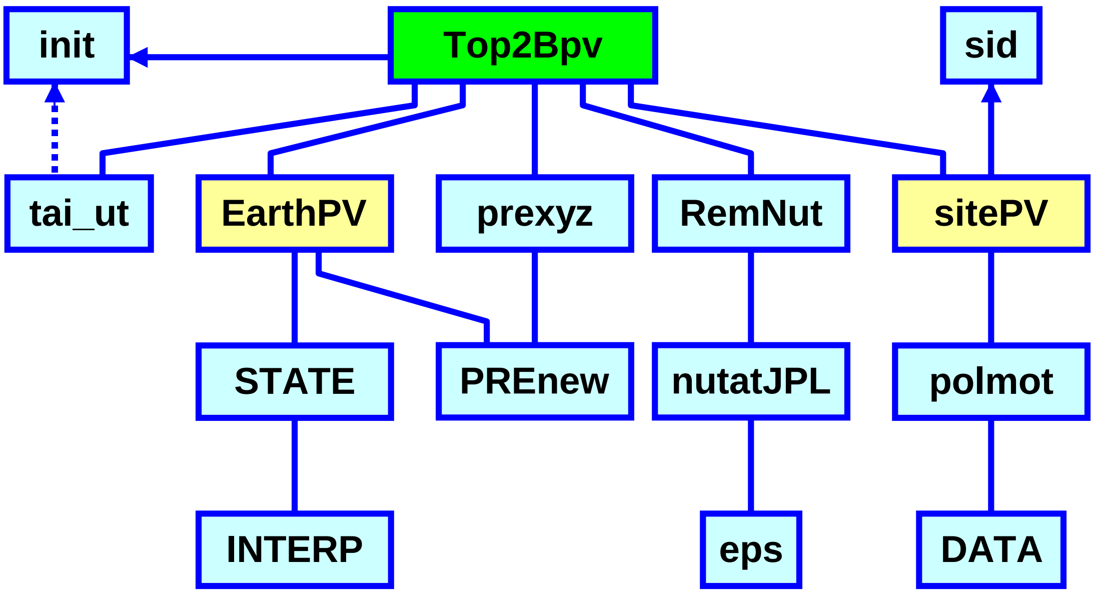

Fig. 2. Block diagram of the `Top2Bary` module

`prexyz` — precesses rectangular coordinates using `PREnew`
`PREnew` — computes general precession (using the new IAU theory[1])
`RemNut` — eliminates nutation from rectangular equatorial vector
`nutatJPL` — returns JPL nutation angles and the mean obliquity
`eps` — calculates the mean obliquity
`sid` — computes local sidereal time (mean or apparent)
`tai_ut` — calculates the difference of TAI − UTC
`polmot` — interpolates IERS UT1 − UTC and offsets of poles
`DATA` — finds Gregorian or Julian calendar date from the JD number

The overall structure of the module is shown in Fig. 2. The module expects the presence of the following data files:
`DE405_80.39` — JPL DE405/LE405 ephemeris file spanning 1980 to 2039, inclusive (binary)
`tai-utc.dat` — TAI − UTC table (ASCII)
`eopc04.yy` — IERS files, one per `yy`-year, for desired years (ASCII)
`useTop2B.cfg` — optional configuration file (ASCII)
Except for the last file which is looked for in the current directory, all the other files must be placed in the `/Top2Bary/EphData` subdirectory. All the ASCII files MUST be formatted in Windows style (i.e. the lines must be terminated with CR/LF) and not UNIX style with LFs only. This is important for a user who updates or modifies existing files or downloads new `eopc04.yy` files from the IERS site, where they are stored in the UNIX format.

The user is expected to append whole the `Top2Bary` code to his program and call the `Top2Bpv` subroutine only, although, of course, he is free to call any subprogram function or subroutine of the package as well. The source code of this main procedure is reproduced verbatim below. A calling program must avoid opening of devices (for reading or printing) with the logical numbers

[1]The once adequate word 'new' is misleading in view of recent revolutionary changes of concepts since by this name really referred is here that old IAU 1976 theory (Lieske 1979).

11, 12, 15 and 16 which are used within the Top2Bary module by procedures tai_ut, STATE, polmot and init, respectively.

If not called earlier, this subroutine calls the init one where certain parameters and constants are defined (either these coded there explicitely or read from the useTop2B.cfg file described in the following section). The ephemeris files required by the module, i.e. DE405_80.39 (or other one as set by the user), tai-utc.dat and yearly eopc04.yy, must be kept in the directory \Top2Bary\EphData on the current disk.

```
      subroutine Top2Bpv(sCJD,clat,clong,height,pve,pvo)

c Procedure to calculate position (km) and velocity (km/s) of an observatory
c located at given geographical position (conventional coordinates in degrees
c and height above an Earth ellipsoid in meters, or corresponding Cartesian
c coordinates) at given Julian Date. It requires a JPL ephemeris file plus
c IERS time and Earth rotation data files.

c        Argument description:
c sCJD  - signed Julian Date representing UTC or TT time
c         If sCJD is negative it signifies rectangular coordinates x,y,z below
c clat  - observatory conventional geographical latitude (deg), or x (km)
c clong - observatory conventional geographical longitude (deg), or y (km)
c height- observatory height above the Earth ellipsoid (m), or z (km)
c pvo   - 6 element array (3 elements for position vector in km, and 3 for
c         velocity, km/s) of the observatory relative to Earth barycenter
c pve   - 6 element array (3 elements for position vector in km, and 3 for
c         velocity, km/s) of the Earth barycenter relative to Solar System Barycenter
c         Subroutines and functions called:
c init - sets a few parameters and constants
c tai_ut(CJD) - returns the TAI - UTC difference
c sitePV(CJD,glat,glong,height_km,oxyz) - returns site vectors of date (CJD epoch)
c earthPV(TDB_JD,pve) - returns JPL Earth barycenter vectors (ICRF)
c RemNut(TDB_JD,pvo) - nutates a 3-vector from CJD back to J2000.0
c prexyz(TDB_JD,2451545d0,pvo) - precesses a 3-vector back to J2000.0

      implicit real*8 (a-h,o-z)
      double precision pvo(6),pve(6)
      common /options/ iUT1_C,iPM,iTDB,NutSid,NutRem,NovF,NOmega,iSunV
      data pi/3.14159265358979323846264338327950d0/

        if(np.eq.0) call init
      leap=0

      EJD = dabs(sCJD)
            if(iTDB.lt.2) then                  ! |sCJD| is indeed JD(UTC)
      CJD = EJD
      EJD = CJD + (tai_ut(CJD) + 32.184d0)/86400d0
            else                                ! |sCJD| is JD(TT), here EJD
      TAI_UTC = tai_ut(-EJD)
      if(TAI_UTC.lt.0d0) leap = 1
      CJD = EJD - (dabs(TAI_UTC) + 32.184d0)/86400d0
            endif

      TDB=EJD
      if(mod(iTDB,2).eq.0) go to 6 ! No conversion to barycentric time

c TDB = TT + 0.001658 sin(g) + 0.000014 sin(2g)  seconds, where
c g = 357.53 + 0.9856003 (JD - 2451545.0) degrees and JD is the Julian Date.
      g = (357.53d0 + 0.9856003d0*(CJD - 2451545.d0))*pi/180
      TDB_EJD = (0.001658*dsin(g) + 0.000014*dsin(2*g))/86400d0
      TDB = EJD + TDB_EJD

6     call earthPV(TDB,pve)        ! Get Earth SSB position & velocity (J2000.0 frame)
c Get observer's Cartesian coordinates and velocity (frame of date, UTC)

      call sitePV(dsign(CJD,sCJD),leap,clat,clong,height,pvo)

c Rotate these vectors to the J2000.0 frame (nutate and precess)
            if(NutRem.ne.0) then
      call RemNut(TDB,pvo)              ! remove nutation from position
      call RemNut(TDB,pvo(4))           ! remove nutation from velocity
            endif
      call prexyz(TDB,2451545d0,pvo)    ! precess position to J2000.0
      call prexyz(TDB,2451545d0,pvo(4)) ! precess velocity to J2000.0
      end
```

4. Configuration file

If in the directory of a program that uses the `Top2Bary` module there is a file named `useTop2B.cfg`, the `init` subroutine opens it and reads some parameters therefrom. The file that originally comes with the module has the following selfexplanatory content:

```
   This is configuration file for the Top2Bary module. The first four lines
and lines starting with a space ' ' or '!' contain comments only. Then  the
module looks for 8 lines which start with 1-digit parameter, which is read.

1 1   iPM          0/1 without/with polar motion
1 1   iUT1_C       0/1 without/with UTC to UT1 conversion
1 1   iTDB         0/1 without/with TDT to TDB conversion, JD is JD(UTC)
                   2/3 as above but assuming JD(TT/TDT) as input argument
2 2   NutSid       0/1/2 mean/apparent sid. time/+IERS corrections (CEP,EqEq)
1 1   NutRem       0/1 without/with nutation of calculated vectors
1 1   NovF         0/1 without/with NOVAS Earth flattening factor
1 1   NOmega       0/1/2 IERS/NOVAS/calculated mean Earth rotation speed
0 0   iSunV        0/1 without/with solar motion towards apex

 The second column above shows default values assumed by the module when it
 cannot find this file. Then finally follow a 12-character ephemeris name:
DE405_80.39        binary JPL ephemeris file (default is DE405_80.39)

The rest of this file is not read by Top2Bary ...
```

To experiment with different parameters just modify the first column of digits above using the options shown further to the right of a parameter name. For example, setting the first parameter (`iPM`) to 0 results in neglecting the polar motion data while computing the detector coordinates; `iTDB` = 3 causes the module to assume TT time on input (TT has no leap second jumps!).

There are a few binary ephemeris files redied in the past for use with Top2Bary, but the following two now replace them all (although those older can be used as well):
• DE405_80.39 — DE405/LE405 for dates 1 Dec 1979 (0^h ET) – 7 Jan 2040 (0^h ET)
• DE200_90.20 — DE200/LE200 for dates 28 Dec 1989 (0^h ET) – 2 Jan 2021 (0^h ET)

These two direct access binaries were compiled with the LF95 and thus they are without the header (the previously used LF90 compiler appended it as the first record in each such file). The user may place any of the names of available binaries in the useTop2B.cfg file at the beginning of the line immediately following the 8 lines containing numerical parameters. If he decides on using predefined settings, he may remove this file altogether or, equivalently, rename it to e.g. DON'TuseTop2B.cfg.

5. Practical examples

SSB2Det program

The program presented here has been used to generate coordinates and velocities required for searches of Explorer data described in Astone *et al.* (2003, 2005 and 2006), and later also for searches of data from the Nautilius and Virgo detectors (e.g., Astone *et al.* 2008 and 2010). For its compilation earlier we have used Lahey Compiler LF90, v. 4.50, and since 2011 Lahey/Fujitsu Fortran 95 Compiler, Release 7.2. During compilation the main module, which is the subject matter of this document, is appended to this small program, as specified in the last line.

```
      Program SSB2Det

c         USAGE:  ssb2det MJD,clat,clong,height,step,N
c SPECIAL USAGE:  ssb2det -MJD, x, y, z, step, N
c                 ssb2det GPS, clat,clong,height,step,N
c                 ssb2det -GPS, x, y, z, step, N
c                 ssb2det JD, clat,clong,height,step,N
c                 ssb2det -JD,  x, y, z, step, N

c where the command line arguments are:
c    MJD - (signed) Modified Julian Date (UTC or TT) of the first sample; if MJD
c       is signed negative, it signifies rectangular coordinates x,y,z below;
c       if iTDB > 1 (in useTop2B.cfg), MJD is assumed to be TT based
c    GPS - (signed) GPS Time in seconds of the first sample;
c       if GPS is negative it signifies rectangular coordinates x,y,z below
c       the GPS Time is recognized by its absolute value greater than 25E5;
c    JD  - (signed) Julian Day Number; valid dates are from 20 Jan 4560 BC (JD55902.5)
c       to 31 Aug 2132 AD (JD2500000.); negative for rectangular coordinates
c    clat   - site conventional geodetic latitude (deg), or x (km)
c    clong  - site geographical longitude, East positive (deg), or y (km)
c    height - site height above the Earth ellipsoid (m), or z (km)
c    step   - sampling interval (s)
c    N      - number of samples

c The program returns four vectors for each sample in four separate files
c (in units of km for positions and km/s for velocities):
c  1. Position vector of the location relative to the Earth barycenter (rDet.dat)
c  2. Position vector of the Earth barycenter relative to the SSB (rSSB.dat)
c  3. Velocity vector of the location relative to the Earth barycenter (vDet.dat)
c  4. Velocity vector of the Earth barycenter relative to the SSB (rSSB.dat)
```

Test data from SSB2Det

```
              rDet.dat                                        rSSB.dat
-2370.863732 -3021.495060  5075.246432        -129159430.774  -70544670.345  -30593308.915
 -537.331253 -3800.569168  5076.958017        -129052968.953  -70714470.051  -30666939.416
 1439.726269 -3555.536363  5078.769833        -128946244.114  -70884127.157  -30740508.024
 3027.590154 -2352.420796  5080.193681        -128839256.472  -71053641.303  -30814014.579
 3798.410052  -515.402279  5080.845908        -128732006.243  -71223012.127  -30887458.920
 3544.488340  1460.534050  5080.550785        -128624493.644  -71392239.268  -30960840.889
 2334.244301  3042.971611  5079.387847        -128516718.892  -71561322.364  -31034160.325
  493.778531  3805.522175  5077.670462        -128408682.206  -71730261.054  -31107417.066
-1480.994958  3542.716287  5075.861387        -128300383.807  -71899054.976  -31180610.955
-3057.972902  2325.367069  5074.448078        -128191823.914  -72067703.768  -31253741.829
-3812.238159   481.489603  5073.811345        -128083002.752  -72236207.067  -31326809.528
-3540.553744 -1492.082823  5074.122743        -127973920.542  -72404564.512  -31399813.893
-2316.125097 -3063.570579  5075.298358        -127864577.511  -72572775.740  -31472754.762
 -468.874851 -3809.535863  5077.021416        -127754973.883  -72740840.389  -31545631.975
 1503.454914 -3528.978134  5078.827643        -127645109.886  -72908758.095  -31618445.372
 3069.419406 -2297.493756  5080.230362        -127534985.748  -73076528.496  -31691194.791
 3807.069070  -446.906540  5080.851623        -127424601.699  -73244151.229  -31763880.073
 3517.644007  1524.142276  5080.524042        -127313957.970  -73411625.931  -31836501.057
 2279.129882  3084.553040  5079.335892        -127203054.793  -73578952.239  -31909057.582
  425.244668  3813.872636  5077.607317        -127091892.401  -73746129.791  -31981549.487
-1544.481756  3515.585689  5075.804069        -126980471.030  -73913158.223  -32053976.613
-3099.306065  2270.065686  5074.412012        -126868790.915  -74080037.172  -32126338.799
-3820.280425   412.918318  5073.806216        -126756852.295  -74246766.275  -32198635.883
-3513.138096 -1555.447567  5074.149896        -126644655.408  -74413345.169  -32270867.707
-2260.638663 -3104.655261  5075.350443        -126532200.494  -74579773.492  -32343034.109
```

```
           vDet.dat                           vSSB.dat                 epsilon.dat      phir.dat
 0.220342 -0.172545  0.000208   14.766243 -23.590229 -10.229470   0.409146859   4.045933961823
 0.277153 -0.038842  0.000256   14.802784 -23.570451 -10.220886   0.409146865   4.571031890198
 0.259285  0.105327  0.000235   14.839294 -23.550623 -10.212280   0.409146870   5.096129817295
 0.171553  0.221116  0.000151   14.875775 -23.530745 -10.203651   0.409146875   5.621227743153
 0.037595  0.277325  0.000026   14.912226 -23.510817 -10.195001   0.409146879   6.146325667816
-0.106492  0.258809 -0.000106   14.948647 -23.490838 -10.186328   0.409146882   0.388238284149
-0.221885  0.170557 -0.000209   14.985037 -23.470809 -10.177633   0.409146884   0.913336206563
-0.277491  0.036348 -0.000256   15.021397 -23.450730 -10.168915   0.409146886   1.438434127930
-0.258327 -0.107655 -0.000234   15.057726 -23.430601 -10.160175   0.409146887   1.963532048306
-0.169557 -0.222650 -0.000149   15.094025 -23.410421 -10.151413   0.409146887   2.488629967752
-0.035099 -0.277652 -0.000024   15.130292 -23.390192 -10.142629   0.409146886   3.013727886331
 0.108816 -0.257841  0.000108   15.166529 -23.369911 -10.133823   0.409146885   3.538825804109
 0.223411 -0.168554  0.000211   15.202734 -23.349581 -10.124994   0.409146882   4.063923720999
 0.277807 -0.033850  0.000257   15.238907 -23.329200 -10.116143   0.409146878   4.589021636343
 0.257349  0.109974  0.000233   15.275050 -23.308769 -10.107269   0.409146874   5.114119551103
 0.167547  0.224166  0.000147   15.311160 -23.288287 -10.098373   0.409146868   5.639217465359
 0.032600  0.277957  0.000021   15.347238 -23.267755 -10.089455   0.409146862   6.164315379191
-0.111131  0.256851 -0.000110   15.383284 -23.247172 -10.080515   0.409146854   0.406227985503
-0.224918  0.166538 -0.000212   15.419298 -23.226539 -10.071552   0.409146846   0.931325898741
-0.278100  0.031350 -0.000257   15.455279 -23.205856 -10.062567   0.409146836   1.456423811814
-0.256349 -0.112285 -0.000232   15.491228 -23.185122 -10.053560   0.409146825   1.981521724812
-0.165524 -0.225664 -0.000145   15.527143 -23.164338 -10.044531   0.409146814   2.506619637828
-0.030099 -0.278239 -0.000019   15.563026 -23.143504 -10.035479   0.409146801   3.031717550956
 0.113436 -0.255841  0.000112   15.598875 -23.122619 -10.026405   0.409146787   3.556815464291
 0.226406 -0.164508  0.000213   15.634691 -23.101684 -10.017309   0.409146772   4.081913377926
```

```
c ascribed to disk devices as the units 1 to 4, respectively. True obliquity
c and sidereal time (in radians) are saved in files epsilon.dat and phir.dat.

      implicit real*8 (a-h,o-z)
      real*8 pvo(6),pve(6)            ! arrays for site (pvo) and Earth (pve) vectors
      character cline*127             ! max 127 characters for command line arguments
      common /options/ iUT1_C,iPM,iTDB,NutSid,NutRem,NovF,NOmega,iSunV
      common /apparent/ djUT1,ALST    ! DJ(UT1) and apparent local sidereal time

      call getcl(cline)               ! get command line to cline
            if(cline.eq.' ') then
      print*,' No command line arguments given.  Running a test ...'
      cline=' 48002.0123456789, 53.1, 18.55, 127, 7200.9001, 25'
            endif

      read(cline,*) sMJD,clat,clong,height,step,N      ! read command line arguments

c If input time is Julian Day Number, convert it to MJD. The conditions assumed
c below mean that MJD can be used at input only for dates until 31 August 2132. Valid
c JD envelopes dates from 20 Jan 4560 BC till 31 August 2132 AD.
      if(dabs(sMJD).le.2500000d0.and.dabs(sMJD).gt.55902.5d0)
     & sMjD = dsign(dabs(sMJD) - 2400000.5d0,sMJD)

      call init

         if(dabs(sMJD).gt.2500000d0) then     ! Condition for GPS time being given
c instead of MJD. Since the value 2500000 s corresponds to GPS time in Feb 3, 1980,
c this option can be used for later dates only.
      gMJD0 = 44244                           ! origin of GPS time (Jan 6, 1980)
      gMJD = gMJD0 + dabs(sMJD)/86400         ! MJD(GPS)
      TTJD0 = 2400000.5d0 + gMJD  + (19 + 32.184d0)/86400d0         ! JD(TT)
c 19 s above is the TAI - UT diffrence at gMJD0 (origin of GPS time)

      if(iTDB.lt.2) iTDB=iTDB+2               ! forcing TT input time for Top2Bpv
      print*,' The first argument is assumed to be GPS time in seconds!'
      TTorUT = TTJD0
         else
      TTorUT = 2400000.5d0 + dabs(sMJD)       ! TT or UTC Julian Date of first sample
            if(iTDB.lt.2) then                ! |sMJD| is MJD(UTC)
      TTJD0 = TTorUT + (tai_ut(TTorUT) + 32.184d0)/86400d0         ! JD(TT)
            else                              ! |sMJD| is MJD(TT)
      TTJD0 = TTorUT                          ! JD(TT) <- Julian Date of first sample
            endif
         endif

      open(1,file='rDet.dat')
      open(2,file='rSSB.dat')
      open(3,file='vDet.dat')
      open(4,file='vSSB.dat')
      open(5,file='epsilon.dat')
      open(7,file='phir.dat')

            do 1 i = 0, N-1
      Toff = i*step/86400d0
      sCJD=dsign(TTorUT + Toff,sMJD)
      call Top2Bpv(sCJD,clat,clong,height,pve,pvo)
      write(1,'(3f13.6)') (pvo(j),j=1,3)
      write(2,'(3f16.3)') (pve(j),j=1,3)
      write(3,'(3f10.6)') (pvo(j),j=4,6)
      write(4,'(3f11.6)') (pve(j),j=4,6)
      call nutatJPL(TTJD0 + Toff,dpsi,deps,eps)
      write(5,'(1x,f11.9)') eps + deps
      write(7,'(f15.12)') ALST             ! ALST comes from common block /apparent/
1     continue
      end

      include 'Top2Bary3.0.for'            ! append the module
```

The program executable `SSB2Det.exe` can be called by the user (or another application) with arguments explained in comments of the program (see source code printout above). When run without the command line arguments, it assumes values specified in the code itself so that the generated output files can be used to check proper functioning of the module. For convenience of the user their content is printed here in full on a separate page. These test data can be reproduced (with occasional differences at the last digit due to round-off errors of input data) by executing this program in one of the following manners:

```
ssb2det  48002.0123456789,    53.1,18.55, 127, 7200.9001,25  — MJD(UTC) as the start time
ssb2det -48002.0123456789,    3638.473270,1220.947798,5077.337129,7200.9001,25
ssb2det  2448002.5123456789,  53.1,18.55, 127, 7200.9001,25  — JD(UTC) as the start time
ssb2det -2448002.5123456789,  3638.473270,1220.947798,5077.337129,7200.9001,25
ssb2det  324692272.66665696,  53.1,18.55, 127, 7200.9001,25  — GPS Time as the start time
ssb2det -324692272.66665696,  3638.473270,1220.947798,5077.337129,7200.9001,25
ssb2det  48002.01300753075185,53.1,18.55, 127, 7200.9001,25  — MJD(TT) as the start time
ssb2det -48002.01300753075185,3638.473270,1220.947798,5077.337129,7200.9001,25
ssb2det  2448002.51300753075185,53.1,18.55, 127,7200.9001,25  — JD(TT) as the start time
ssb2det -2448002.51300753075185,3638.473270,1220.947798,5077.337129,7200.9001,25
```

where the last four cases (with the Terrestrial Time, TT, at the input) must be run with `iTDB` set to 3 inside `useTop2B.cfg`.

Each of the output test files has 25 rows and 3 columns (corresponding to x, y and z components of a vector), except `epsilon.dat` and `phir.dat` which are one-column files.

SSB2DetC program

This is just a variety of the previous program. For specified coordinates on the sky it generates one output file, `CorSec.dat`, wherein there is one column table that consists of sampled delay (in seconds) of signal wave-front arriving at the detector with respect to the SSB. Here is the header of the program:

```
Program SSB2DetC

c This program returns a time correction (in seconds) for each sample.
c The correction is equal to the time needed by the wave front of a signal
c coming from a source at given position (RA2000,D2000) to travel the distance
```

```
c from the detector to the Solar System barycenter. Results are written to
c a file named CorSec.dat. This is version of Dec. 11, 2011 and it uses
c the package Top2Bary3.0, which at its input allows MJD to represent TT
c instead of UTC (useful to work across leap seconds) and rectangular
c coordinates. MJD(TT) must be associated with iTDB set to 2 or 3 in
c the useTop2B.cfg configuration file. GPS time must be given in seconds
c and must be greater than 2500000.000000000 s (4 Feb, 1980 onwards), but
c need not be associated with iTDB > 1 (this program forces proper setting).

c USAGE:          ssb2detC MJD,clat,clong,height,step,N,RA2000,D2000
c SPECIAL USAGE:  ssb2detC -MJD, x, y, z, step, N,RA2000,D2000
c                 ssb2detC GPS, clat,clong,height,step,N,RA2000,D2000
c                 ssb2detC -GPS, x, y, z, step, N,RA2000,D2000
c                 ssb2detC JD, clat,clong,height,step,N,RA2000,D2000
c                 ssb2detC -JD, x, y, z, step, N,RA2000,D2000

c where the command line arguments are:
c    MJD   - (signed) Modified Julian Date (UTC or TT) of the first sample;
c       if MJD is signed negative, it signifies rectangular coordinates x,y,z below;
c       if iTDB > 1 (in useTop2B.cfg), MJD is assumed to be TT based
c    GPS   - (signed) GPS Time in seconds of the first sample;
c       if GPS is negative, it signifies rectangular coordinates x,y,z below
c       The GPS Time is recognized by its absolute value greater than 25E5
c    JD    - (signed) Julian Day Number from the range 55902.5 < JD < 2500000, i.e.
c       from 20 Jan 4560 BC to 31 August 2132 AD; if iTDB > 1, JD is JD(TT);
c       if JD is negative, it signifies rectangular coordinates x,y,z below
c    clat  - site conventional geodetic latitude (deg), or x (km)
c    clong - site geographical longitude, East positive (deg), or y (km)
c    height- site height above the Earth ellipsoid (m), or z (km)
c    step - sampling interval (s)
c    N - number of samples
c    RA2000 - Right Ascension of a point on the sky in degrees (J2000.0)
c    D2000 - Declination on the sky in degrees (J2000.0)

c    Subroutines called:
c init - reads the useTop2B.cfg file and sets a few parameters and constants
c skydir2(ra,dec,uvec) - returns unit vector in the requested direction (ra,dec)
c Top2Bpv(UTCJD,clat,clong,height,pve,pvo) -  Earth and observatory position/velocity
```

Here is a test run:

```
ssb2detc 48002.0123456789, 53.1,18.55, 127, 7200.9001,5, 90.,30
```

and contents of the resulting `CorSec.dat` file:

```
  -254.81031330
  -255.42587231
  -256.03795762
  -256.64675943
  -257.25321357
```

The same is obtained by running `ssb2detc` without command line arguments.

Orb program

Here we present another application, that we have used just for check-up of less accurate data describing the Earth orbit in space. The executable `Orb.exe` generates rectangular coordinates of the Earth center of mass with respect to the SSB (JPL DE405 frame, thus ICRF) according to user specifications in the command line or in response to a program prompt. Results of computations are written to a file (with the name composed of 'Orb' and the Modified Julian Date value of the first sample rounded to three decimal places) in a 4-column table. The table colums contain the time offset (relative to the first sample) in TAI/UTC days and the three Cartesian coordinates of the Earth expressed in astronomical units (AU). The source program listing below includes a numerical example of input arguments and the resulting output.

```
      Program Orb                      ! computes coordinates of Earth wrt SSB

c USAGE:             Orb MJD,step,N
c SPECIAL USAGE:     Orb GPS,step,N
c                    Orb  JD,step,N

c where the command line arguments are as follows:
c       MJD - Modified Julian Date (UTC or TT based) of the first sample
c             (if based on TT, do set iTDB to 3 in useTop2B.cfg file)
c       GPS - GPS Time in seconds. It is recognized by its value greater
c             than 25E5 (thus can be used for dates after Feb 3, 1980 only)
c       JD  - Julian Day Number from the range 55902.5 < JD < 2500000, i.e. from
c             20 Jan 4560 BC to 31 August 2132 AD; if iTDB > 1, JD is JD(TT)
c       step - sampling interval [UTC/TAI seconds]
c       N - number of samples

c The program returns rectangular coordinates of the Earth referred to
c SSB (frame: ICRF) in a file 'orb00000.000' where zeroes stand for MJD;
c the first column of the file gives time offset (in days) of each sample.
c The input MJD may represent terrestrial time TT (istead of normal UTC)
c which allows taking samples across leap seconds. MJD(TT) must be associated
c with iTDB set to 2 or 3 in useTop2B.cfg file. The following DOS command:
c "orb 53735.999988 .25 14" with MJD(UTC) as start time corresponds to
c "orb 53736.00073087 .25 14" with the TT option on (iTDB set to 2 or 3).
c GPS time on input also allows for computing within leap seconds, since
c it is automatically converted to TT in this (Orb) program.

      implicit real*8 (a-h,o-z)
      real*8 pve(6),pvo(6),MJD
      character cline*80,outfil*12
      data AU/0.1495978706910d+09/     ! from JPL header.405
     &, clat,clong,height/0d0,0d0,0d0/
      common /options/ iUT1_C,iPM,iTDB,NutSid,NutRem,NovF,NOmega,iSunV

      call getcl(cline)

            if(cline.eq.' ') then
      print'(a,$)',' No command line arguments given. Type-in MJD,'//
     & 'step,N: ' !$
      read(*,*) MJD,step,N
      write(cline,*) MJD,step,N
            endif
      read(cline,*) MJD,step,N

c If input time is Julian Day Number, convert it to MJD. The conditions assumed
c below mean that MJD can be used at input only for dates until 31 August 2132.
c Valid JD envelopes dates from 20 Jan 4560 BC till 31 August 2132 AD.
      if(MJD.le.2500000d0.and.MJD.gt.55902.5d0) MjD = MJD - 2400000.5d0

         if(MJD.gt.2500000d0) then  ! Condition for GPS time being
c given instead of MJD. Since the value 2500000 s corresponds to GPS
c time in Feb 3, 1980, this option can be used for later dates only.
      MJD0 = 44244                  ! origin of GPS time (Jan 6, 1980)
      MJD  = MJD0 + MJD/86400       ! MJD(GPS)
      MJD  = MJD  + (19 + 32.184d0)/86400d0                 ! MJD(TT)
c 19 s above is the TAI - UT diffrence at MJD0 (origin of GPS time)
      call init
            if(iTDB.lt.2) then      ! This MJD is now based in TT so
      iTDB = iTDB + 2               ! forcing TT as input time for Top2Bpv
      print*,' The first argument is assumed to be GPS time in seconds!'
            endif
         endif
      UTCJD0 = 2400000.5d0 + MJD                            ! JD(UTC/TT)

      write(outfil,'(a,f9.3)') 'Orb',MJD
      open(1,file=outfil)

            do 1 j = 0, N-1
      T = j*step/86400d0
      call Top2Bpv(UTCJD0 + T,clat,clong,height,pve,pvo)
1     write(1,'(f14.9,3f19.15)') T,(pve(k)/AU,k=1,3)         ! position
c     & ,(pve(k)/AU*1d5*86400d0,k=4,6)     ! velocity in 1E-5 AU/d (for AA test)
      print'(a,a,f12.5,a,i6)',' Output in the file "'//outfil,
     & '"     step [s] =',step,'     N =',N

      end

      include  'Top2Bary3.0.for'


c        Test example
c Input (typed on prompt or in command line): 48580.790850744, 86400, 3
c Contents of the output file "Orb48580.791"

c                When computed with double precision:
c   0.000000000  0.524712484382844  0.771327942103784  0.334341141730002
c   1.000000000  0.509759571077188  0.779499088435775  0.337884390296571
c   2.000000000  0.494651799591344  0.787432883266874  0.341324852639191

c     When computed with quadruple precision (differences are < 0.3 m):
c   0.000000000  0.524712484381481  0.771327942104544  0.334341141730331
c   1.000000000  0.509759571079175  0.779499088434710  0.337884390296110
c   2.000000000  0.494651799589619  0.787432883267761  0.341324852639576
```

6. Dealing with leap seconds (TT, GPS) and impact of quad precision

The Modified Julian Date based on UTC does not allow to set uniquely the time of a leap second since it would be numerically indistinguishable from the time of the second immediately following it. To have a way to generate data also within the leap second one can revert to using the MJD based on TT (TDT) time instead of UTC as an input argument. This possibility has been implemented first in `Top2Bary` v. 2.2 (of 2006). It requred to considerably modify the `tai_ut` subprogram in order that it is able to accept also JD(TT) as its input argument and to detect moments when JD(TT) falls within the range of a leap second. For the user to work with MJD(TT) he just advances the value of `iTDB` option in `useTop2B.cfg` file by 2 (to make it 2 or 3, depending on the previous value). Another way of having samples across a leap second provides optional GPS time as the start time at the input. This option also uses TT time but internally, so that the user need not to set `iTDB` parameter to 2 or 3 (but he may do so; if he doesn't, it is automatically reset by this program).

The usual UTC based MJD can be converted into the TT based MJD or into GPS Time with the formulae:

$$\mathbf{MJD(TT) = MJD(UTC) + (TAI - UTC + 32.184)/86400},$$

$$\mathbf{GPS\ Time = [MJD(UTC) - 44244]\,86400 + (TAI - UTC) - 19},$$

where the TAI − UTC difference (since 1972 it is an integer number of TAI seconds) can be taken from our `\Top2Bary\EphData\tai-utc.dat` file (remembering that within the leap second itself the difference is the same as prior to it). E.g., the most recently introduced leap second at the end of 2008 begins at MJD = 54832.0 and lasts till another numerically the same MJD but 1 UTC/TAI second later. Since TAI − UTC in 2008 equalled to 33 s, these two moments, when expressed in TT, equal to 54832 + 65.184/86400 and 54832 + 66.184/86400 at the beginning and end, respectively, of the leap second. The same span of one second expressed in GPS time would extend from 914803214.0 to 914803215.0.

The option to work with the TT also allows for direct comparison of certain computations with data published in astronomical yearbooks (which frequently contain ephemerides, whose argument is the ephemeris time ET, which is equivalent or close to TDT, or TT, or T_{eph}). As an example, having set `iTDB = 3`, we have used the `Orb` program (with the line following the label 1 there uncommented to have velocities as well) supplying the following input data: `53004, 864000, 3`, which means requesting starting date of January 0, 2004 (i.e. December 31, 2003), at 0.0 hours of TT, step of 10 round days, and generating 3 samples. The results are included in the table below, where there are three lines for each of the three dates (samples). The X,Y,Z coordinates in the first line were computed with the code compiled as it is or (in practice equivalently) with double precision option enabled (DP; thus all reals declared in the code to be 4-byte long are automatically changed into 8-byte reals) and in the second line − with quadruple precision enabled (QP, all 8-byte reals changed into 16-byte reals and 4-byte reals into 8-byte reals). The third line contains corresponding data taken from the *Astronomical Almanac for the year 2004 (AA)*.

Earth position (X, Y and Z in AU) with respect to SSB at 0^h TT (T_{eph} in *AA*)

0.000000000	-0.147440491730541	0.888923036138659	0.385320984472414	DP
	-0.147440491731235	0.888923036138560	0.385320984472371	QP
Jan 0, 2004	-0.147440492	0.888923036	0.385320984	*AA*
10.000000000	-0.316918997408261	0.850456482944105	0.368637656795466	DP
	-0.316918997410452	0.850456482943417	0.368637656795168	QP
Jan 10, 2004	-0.316918997	0.850456483	0.368637657	*AA*
20.000000000	-0.476535025262845	0.785575371003840	0.340512466873750	DP
	-0.476535025265341	0.785575371002575	0.340512466873202	QP
Jan 20, 2004	-0.476535025	0.785575371	0.340512467	*AA*

As expected, our positional data agree exactly with those published in the *AA*. Likewise, the corresponding velocities (in 10^{-5} AU/d) turned out to be numerically identical with those of the *AA*, wherein they are printed in the same units and format. Here are the results (they are the same for both the precisions, DP and QP):

Day	X_{dot}	Y_{dot}	Z_{dot}
0	−1727.5341	−248.0625	−107.6159
10	−1653.6985	−519.1662	−225.0970
20	−1530.1763	−775.5049	−336.1788

Appendix: T2C2kA module implementing the new IAU2000 system

Modules T2C2k serve to transform position and velocity vectors from terrestrial to celestial geocentric frame of reference according to most recent developments in the field.

```
      subroutine T2C2k(CJD,clat,clong,height,pvo) ! the 'A' variety

c Main subroutine for transformation of site coordinates from the terrestrial
c (ITRF) to celestial (ICRF) reference frame according to the new IAU algorithms.
c It returns position (km) and velocity (km/s) of an observatory located
c at given geographical position (conventional coordinates in degrees and
c height above the Earth ellipsoid in m) at given Julian Date (UTC).
c Performs similar job as does Top2Bpv described in the main text except that
c no Earth barycenter vectors wrt SSB are computed.

c         Argument description:
c CJD   - Julian Date representing the Coordinated Universal Time
c clat  - observatory conventional geographical latitude (deg)
c clong - observatory conventional geographical longitude (deg)
c height- observatory height above the Earth ellipsoid (m)
c pvo   - 6-element array (3 elements for position vector in km, and 3 elements
c         for velocity, km/s) of the observatory relative to Earth barycenter
```

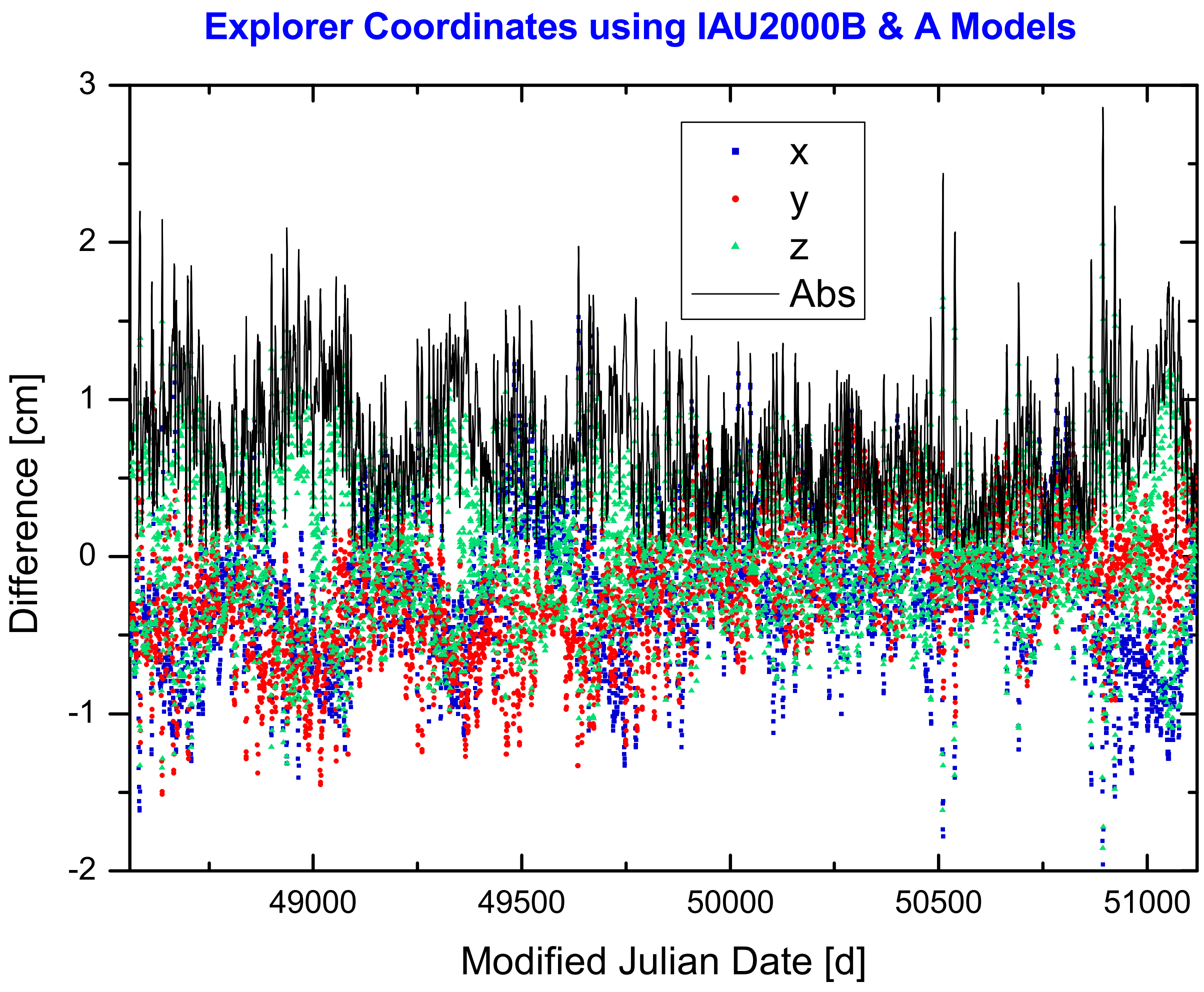

Fig. 3. Differences between Cartesian coordinates of the Explorer referred to the geocentric ICRF as computed with two varieties of the T2C2k corresponding to the IAU model 'B' (less accurate nutation algorithm, classical equator and equinox approach) and 'A' (full precision, presented in this Appendix). The dots represent differences in x, y and z coordinate and the distance between the two positions is drawn with the continuous line. The data plotted were sampled every 12 UTC hours over 7 years starting with MJD = 48561.0 (1 Nov 1991).

```
      implicit real*8 (a-h,o-z)
      double precision pvo(6),xyz(6),RPOM(3,3),RBPN(3,3),RT2C(3,3)
      common /options/ iUT1_C,iPM,iTDB,NutSid,NutRem,NovF,NOmega,iSunV

      call DAT(CJD, tai_utc, J)
      EJD = CJD +( tai_utc + 32.184d0)/86400d0

c Get interpolated Earth orientation and rotation data from EOPC04.yy file
      call polmot(CJD,Px,Py,UT1_C,dpsi,deps)

c Convert or possibly neglect the Earth orientation parameters
```

```
      conv = 1d0/206264.8062470964D0
            if(iPM.eq.0)    conv  = 0
            if(iUT1_C.eq.0) UT1_C = 0
      Px = Px*conv
      Py = Py*conv

c (step 1) Get the s' quantity
      date1 = dint(EJD)
      date2 = EJD - date1
      SP = sp2000(DATE1,DATE2)
*     DATE1,DATE2   d      TT date (JD = DATE1+DATE2)
*     SP2000        d      the quantity s' in radians

c (step 2) Form the polar motion (wobble) matrix (given pole coord.)
      call  POM2000(Px, Py, SP, RPOM)
*     Px,Py         d      coordinates of the pole (radians)
*     SP            d      the quantity s' (radians)
*     RPOM        d(3,3)   polar-motion matrix

c (step 3) Earth Rotation Angle (UT1 required)
      UT1 = CJD + UT1_C/86400d0
      DJ1 = dint(UT1)
      DJ2 = UT1 - DJ1
      ERA = ERA2000(DJ1, DJ2)
*     DJ1,DJ2       d      UT1 date (JD = DJ1 + DJ2)
      call XYS2000A(DATE1, DATE2, X, Y, S)
*     DATE1,DATE2   d      TT as a 2-part Julian Date
*     X,Y           d      Celestial Intermediate Pole
*     S             d      the quantity s

c (step 4) Bias-precession-nutation matrix for intermediate system
c to GCRS transformation
      call BPN2000(X, Y, S, RBPN)
*     X,Y           d      CIP coordinates
*     S             d      the quantity s (radians)
*     RBPN        d(3,3)   intermediate-to-celestial matrix ("Q")

c (step 5) Compose final terrestrial to celestial system rotation matrix
c for the CEO-based transformation
      call T2C2000(RPOM, ERA, RBPN, RT2C)
*     RPOM        d(3,3)   polar motion matrix (W)
*     ERA           d      Earth Rotation Angle (radians, giving matrix R)
*     RBPN        d(3,3)   intermediate-to-celestial matrix (Q)
*     RT2C        d(3,3)   returned terrestrial-to-celestial matrix

c (step 6) Get observer's xyz coordinates and velocities
      deg   = 180/3.141592653589793d0
      along = clong/deg
      alat  = clat/deg
      call OBSxyz(alat,along,height/1000d0,xyz)
c This routine returns (in xyz) conventional rectangular coordinates and velocity
c of site similarly as does sitePV of the Top2Bary but without accounting for
c polar motion and sidereal time (i.e. as if they were all equal to 0).
```

```
c (step 7) Rotate the xyz vectors according to the RT2C matrix (Sofa routine)
      call iau_RXP(RT2C, xyz, pvo)        ! position vector
      call iau_RXP(RT2C, xyz(4), pvo(4))  ! velocity vector

      return
      end

c Other required subroutines appended to or INCLUDEd in the T2C2kA:
c       subroutine OBSxyz(Xlat,Ylong,Zheigh,obs)       - see above (modified sitePV)
c       subroutine init                                - taken from Top2Bary
c       subroutine data(JD,L,M,N,J1G0)                 - taken from Top2Bary
c       subroutine polmot(dj,px,py,pUTC,pdpsi,pdeps)   - taken from Top2Bary
c All the following subprograms come from the SOFA package (www.iau-sofa.rl.ac.uk):
c       DOUBLE PRECISION FUNCTION ERA2000 ( DJ1, DJ2 )     - Earth Rotation Angle
c       DOUBLE PRECISION FUNCTION SP2000 ( DATE1, DATE2 )  - s' quantity
c       DOUBLE PRECISION FUNCTION iau_ANPM ( A ) - normalizes A to -pi <= A < +pi
c       SUBROUTINE DAT ( CJD, DELTAT, J )  - this is iau_DAT modified by KMB
c                                            (to have CJD as input instead of date +)
      include  '\Top2Bary\SOFA\SofaMatr.for' ! matrix calculus grouped in one file
      include  '\Top2Bary\SOFA\POM2000.f'    ! Polar Motion matrix
      include  '\Top2Bary\SOFA\BPN2000.f'    ! Bias+Precession+Nutation matrix
      include  '\Top2Bary\SOFA\XYS2000A.f'   ! Celestial Intermediate Pole IAU2000
      include  '\Top2Bary\SOFA\T2C2000.f'    ! Terrestrial to Celestial final matrix
```

As seen from this partial listing of `T2C2kA` code, it is composed of ready FORTRAN subprograms publicly available in the SOFA package (`www.iau-sofa.rl.ac.uk`) and of a few subroutines adapted or adopted from the `Top2Bary` module. Another variety of this program, `T2C2kB` has similar structure but is built using the classical equator and equinox paradigm (however in this case it is made entirely equivalent to the new paradigm, i.e. using the new nutation-precession theory that does not require the subtle corrections) and employing the shorter nutation procedure (`NU2000B.f`, which is much faster but also considerably less accurate than the one built into `XYS2000A.f` file).

Acknowledgments: My involvement in this project was supported in part by the KBN (Polish State Committee for Scientific Research) Grant No. 1 P03B 029 27 and (since 2010) by MNiSW (Polish Ministery of Science and Higher Education) Grants. Nos. N N203 387237 and 623/N-VIRGO/09/2010/0.

REFERENCES

Astone P., 2003, PSS_astro User guide
(grwavsf.roma1.infn.it/pss/docs/PSS_astro_UG.pdf & .../PSS_astro_PG.pdf).

Astone P., Borkowski K.M., Jaranowski P., Królak A., 2002, Data analysis of gravitational-wave signals from spinning neutron stars. IV. An all-sky search, *Phys. Rev. D*, **65**, 042003
(preprint www.astro.uni.torun.pl/~kb/Papers/PhysicalReview/AllSkySearch.htm).

Astone P. *et al.*, 2003, All-sky upper limit for gravitational radiation from spinning neutron stars, *Class. Quantum Grav.,* **20**, No 17 (7 September 2003), S665–S676
(iopscience.iop.org/0264-9381/20/17/310/pdf/0264-9381_20_17_310.pdf).

Astone P. *et al.*, 2005, An all-sky search of EXPLORER data, *Class. Quantum Grav.,* **22**, No 18 (21 September 2005), S1243–S1254
(iopscience.iop.org/0264-9381/22/18/S38/pdf/0264-9381_22_18_S38.pdf).

Astone P. *et al.*, 2006,, All-sky search of EXPLORER data: search for coincidences, *Class. Quantum Grav.,* **23**, No 19 (7 October 2006), S687S692
(iopscience.iop.org/0264-9381/23/19/S06/pdf/0264-9381_23_19_S06.pdf).

Astone P. *et al.*, 2008, All-sky search of NAUTILUS data, *Class. Quantum Grav.,* **25**, No 18 (21 September 2008), 184012
(iopscience.iop.org/0264-9381/25/18/184012/pdf/0264-9381_25_18_184012.pdf).

Astone P., Borkowski K.M., Jaranowski P., Królak A., Pietka M., 2010, Data analysis of gravitational-wave signals from spinning neutron stars. V. A narrow-band all-sky search, *Phys. Rev. D*, **82**, 022005 (preprint: lanl.arxiv.org/abs/1003.0844).

Borkowski K.M., 2003, Referring a Detector to the Solar System Barycenter, *Mathematics of Gravitation II*, Warsaw, September 1 – 9, 2003
(www.astro.uni.torun.pl/~kb/AllSky/Talks/Borkowski2003/KB-Refer.htm).

Capitaine N. *et al.* (eds.), 2002, Proceedings of the IERS Workshop on the Implementation of the New IAU Resolutions, Paris, 18 – 19 Apr. 2002, *IERS Technical Note* No. 29
(www.iers.org/nn_11216/IERS/EN/Publications/TechnicalNotes/tn29.html).

Kaplan G.H., 2005, The IAU Resolutions on Astronomical Reference Systems, Time Scales, and Earth Rotation Models, *USNO Circ.* No. 179 (Oct. 20, 2005)
(aa.usno.navy.mil/publications/docs/Circular_179.pdf).

Kaplan G.H. *et al.*, 1989, Mean and Apparent Place Computations in the New IAU System. III. Apparent, Topocentric, and Astrometric Places of Planets and Stars, *Astron. J.,* **97**, 1197 – 1210
(aa.usno.navy.mil/software/novas/novas_info.html).

Lieske J.H., 1979, Precession Matrix based on the IAU(1976) System of Astronomical Constants, *Astron. Astrophys.,* **73**, 282 – 284.

McCarthy D. D., Petit G. (eds.), 2003, IERS Conventions (2003), *IERS Tech. Note* 32 (Frankfurt: IERS Central Bureau) (ftp://maia.usno.navy.mil/conv2003/tn32.pdf or www.iers.org/nn_11216/IERS/EN/Publications/TechnicalNotes/tn32.html).

Standish E.M., 1998, JPL Planetary and Lunar Ephemerides, DE405/LE405, IOM 312.F–98–048
(iau-comm4.jpl.nasa.gov/de405iom/de405iom.pdf).

Wallace P.T., 2004, SOFA software support for IAU 2000, American Astronomical Society Meeting 204, #28.02, May 2004 (www.iau-sofa.rl.ac.uk/publications/aas04.pdf).

Report last modified: 2011, Dec. 19