SSE

SOFA-based solar system ephemeris tools

Patrick Wallace

RAL Space, UK

Software version 1
Document revision 1.0

2022 March 13

Contents

1 Introduction

This document describes EPH, a library of C functions that implements the analytical solar-system ephemerides VSOP2010[1] (planets, see the file `VSOP2010.pdf`) and ELP/MPP02 (Moon, see the file `ELP_MPP02.pdf`). It is compatible with the IAU Standards of Fundamental Astronomy software,[2] from which its astrometric and vector/matrix tools are drawn.

The ephemerides are with respect to ICRS axes as implemented in JPL DE405.

EPH is appropriate for applications that require high accuracy without the need to download and interpolate files containing numerically integrated ephemerides.

The choice of VSOP2010 over the more recent VSOP2013 was to optimize compatibility with JPL DE405, VSOP2013 having been fitted instead to INPOP10a (IMCCE, Observatoire de Paris). It should be noted that although the agreement between the different ephemerides is generally very good (for geocentric directions, milliarcsecond level) this falls off dramatically (2-3 orders of magnitude) for the outer planets.

2 Transforming ephemeris files into binary

Both VSOP2010 and ELP/MPP02 are provided as ASCII files containing trigonometric series for the various elements. Even on modern processors these take a considerable time to input and decode, and EPH includes freestanding applications `plan_bin.c` and `moon_bin.c` that read the ASCII files and turn them into binary forms that are ready to be used by the EPH functions.

Both `plan_bin.c` and `moon_bin.c` contain hardwired file names and so simply have to be executed without command-line arguments to generate the binary files.

The input ASCII files are:

```
elp_main.s1
elp_main.s2
elp_main.s3
elp_pert.s1
elp_pert.s2
elp_pert.s3
VSOP2010p1.dat
VSOP2010p2.dat
VSOP2010p3.dat
VSOP2010p4.dat
VSOP2010p5.dat
VSOP2010p6.dat
VSOP2010p7.dat
VSOP2010p8.dat
VSOP2010p9.dat
```

[1] Pluto is omitted.
[2] See `https://www.iausofa.org`.

... and the output binary files are:

```
ELP_MPP02_JPL.ctx
ELP_MPP02_LLR.ctx
VSOP2010_1.ctx
VSOP2010_2.ctx
VSOP2010_3.ctx
VSOP2010_4.ctx
VSOP2010_5.ctx
VSOP2010_6.ctx
VSOP2010_7.ctx
VSOP2010_8.ctx
VSOP2010_9.ctx
```

3 The EPH functions

The EPH library comprises only ten functions. Three of these are used internally and are of no great concern to the EPH user:

- `ephBopr` opens a binary ephemeris file for reading.
- `ephMooni` populates a context directly from the ELP/MPP02 lunar ephemeris ASCII files.
- `ephPlani` populates a context directly from one of the VSOP2010 planetary ephemeris ASCII files.

Of the remaining seven functions, two are used for populating the context structures that contain the binary tables needed for generating ephemerides:

- `ephMoonc` populates a context by reading a lunar ephemeris binary file.
- `ephPlanc` populates a context by reading a planetary ephemeris binary file.

Next come three low-level functions that generate position+velocity state vectors for the different bodies:

- `ephEarth`
- `ephMoon`
- `ephPlanet`

The final two functions are the highest level of those offered. For a given moment in time they calculate the ICRS astrometric $[\,\alpha, \delta\,]$ and the topocentric apparent $[\,\alpha, \delta\,]$ of the specified body as seen by an observer either on the Earth's surface or at the geocenter:

- `ephRdplan`
- `ephRdplanq`

They do the same job in two different ways, with `ephRdplan` a little simpler to call and `ephRdplanq` significantly faster in the common case where a series of predictions for a single body are to be made.

Full details of all ten functions are given later, in Section 5. For most readers, the best introduction is to study the two example applications described in the next section.

4 Example applications using EPH functions

4.1 Occultation

The example application `occultation.c` looks at an occultation by the Moon of Mars on 2020 February 18 as seen from a location in the USA. The time span covers first and second contact. Here are extracts from the report:

```
2020/02/18  12:04:00.000 UTC  clear of Moon
2020/02/18  12:04:00.500 UTC  clear of Moon
2020/02/18  12:04:01.000 UTC  clear of Moon
                       :
2020/02/18  12:04:11.000 UTC  clear of Moon
2020/02/18  12:04:11.500 UTC  clear of Moon
2020/02/18  12:04:12.000 UTC  clear of Moon
2020/02/18  12:04:12.500 UTC  mostly visible
2020/02/18  12:04:13.000 UTC  mostly visible
2020/02/18  12:04:13.500 UTC  mostly visible
                       :
2020/02/18  12:04:18.500 UTC  mostly visible
2020/02/18  12:04:19.000 UTC  mostly visible
2020/02/18  12:04:19.500 UTC  mostly visible
2020/02/18  12:04:20.000 UTC  mostly hidden
2020/02/18  12:04:20.500 UTC  mostly hidden
2020/02/18  12:04:21.000 UTC  mostly hidden
2020/02/18  12:04:21.500 UTC  mostly hidden
                       :
2020/02/18  12:04:26.000 UTC  mostly hidden
2020/02/18  12:04:26.500 UTC  mostly hidden
```

```
2020/02/18  12:04:27.000 UTC  mostly hidden
2020/02/18  12:04:27.500 UTC  completely hidden
2020/02/18  12:04:28.000 UTC  completely hidden
2020/02/18  12:04:28.500 UTC  completely hidden
                       :
```

4.2 Planets

The example application `planets.c` produces a table listing topocentric apparent $[\alpha, \delta]$ and angular diameter for the Sun, Moon and planets for a given date, time and geographical location. The circumstances are specified interactively. A default location is defined early on in the code, while the default time is obtained from the system clock.

Here is an example report, where the location has been allowed to default but the UTC date and time has been specified:

```
Longitude? (D,M,S, east +ve):
Latitude? (D,M,S, north +ve):
Height above sea level? (meters, g=geocentric):
Date? (UTC Y,M,D, Gregorian): 2020 6 19
Time? (UTC H,M,S, q=quit): 8

 2 13 53.0 E  +48 48 18.0, alt 162.0m
 2020/06/19  08:00:00.0 UTC
 2020/06/19  08:01:09.2 TDB

 Sun        05 53 27.80  +23 25 35.7  1887.8    97.51  38.19
 Mercury    07 03 06.75  +20 55 45.4    11.1    86.22  24.99
 Venus      04 18 38.46  +18 33 44.0    51.4   125.25  48.99
 Moon       04 18 08.13  +18 41 00.5  1835.4   125.30  49.16
 Mars       23 44 39.29  -04 45 27.4    10.5   219.56  28.77
 Jupiter    19 49 19.46  -21 19 47.9    46.5   257.63 -17.73
 Saturn     20 12 09.63  -20 09 30.8    18.2   254.42 -13.23
 Uranus     02 28 39.26  +14 12 04.0     3.4   168.26  54.93
 Neptune    23 28 27.32  -04 34 06.4     2.3   223.76  27.17

Time? (UTC H,M,S, q=quit):q
```

Note that a lunar occultation of Venus is in progress.

All but a sixth of the code is concerned with running the interactive session (input decoding, reporting *etc.*), and the only part concerned with using the EPH library is lines 202-272.

5 Function specifications

ephBopr	*open binary ephemeris file for reading*	**ephBopr**

CALL :

```
fp = ephBopr ( path, file, &jstat );
```

ACTION :

Internal function: open a binary ephemeris file for reading.

GIVEN :

path	`const char*`	path for filenames
file	`const char*`	filename of the binary ephemeris file

RETURNED :

jstat	`int`	status: 0 = OK −1 = path + filename too long −2 = file not found

RETURNED (function value) :

	`FILE*`	file pointer for opened file

NOTES :

1. Generating planetary ephemerides involves first reading files to populate arrays of coefficients *etc.* that can later be used for computation of the coordinates for a nominated time. The present function assists in this first phase, by opening one of the ephemeris files (in its binary form).
2. The given path, which typically comes from the command line, may or may not be empty. If it is null, but the named file turns out not to be in the current directory, a second attempt is made using a standard path appropriate for the given platform.
3. The `path` argument is used exactly as given, with no handling of leading or trailing spaces, and must contain any required directory separators such as slash or backslash.
4. If `path` is the null pointer the behavior is as for an empty string.

ephEarth — *heliocentric Earth PV* — ephEarth

CALL :

```
j = ephEarth ( date, &cmoon, &cemb, pv );
```

ACTION :

Heliocentric ICRS position and velocity of the Earth.

GIVEN :

date	`double`	date, TDB Modified Julian Date (Note 1)
cmoon	`ephMOONctx*`	context for Moon (Note 2)
cemb	`ephPLANctx*`	context for EMB (Note 2)

RETURNED :

pv	`double[2][3]`	Earth position & velocity (Note 3)

RETURNED (function value) :

`int`	status: 0 = OK else = ephemeris problem

NOTES :

1. The date is TDB as an MJD (= JD−2400000.5). TT can be used instead of TDB in most applications.
2. The context arguments point to tables of constants that are used by the lunar and planetary ephemeris functions `ephMoon` and `ephPlanet`. Before the present function is called, these two context tables must be populated, by calling the function `ephMoonc` in the case of `cmoon` (the Moon) and `ephPlanc` in the case of `cemb` (the Earth-Moon barycenter).
3. The result `pv` is the heliocentric position and velocity of the geocenter in au and au/s with respect to ICRS axes.
4. Compared with JPL DE405 over the 21st century, the position errors are 1.0 km RMS (1.5 km worst case), and the speed errors are 0.2 mm/s RMS (0.3 mm/s worst case).

ephMoon — *Moon position and velocity* — ephMoon

CALL :

```
j = ephMoon ( &c, date, pv );
```

ACTION :

Geocentric ICRS position and velocity of the Moon.

GIVEN :

c	`ephMOONctx*`	ephemeris context structure
date	`double`	TDB as a Modified Julian Date (JD−2400000.5)

RETURNED :

pv	`double[2][3]`	Moon $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$, (au, au/s)

RETURNED (function value) :

	`int`	status: 0 = OK -1 = uninitialized context

NOTES :

1. Before the present function can be called, the `ephMoonc` or `ephMooni` function must be executed.
2. TT can be used instead of TDB in most applications, but not UTC.
3. Generating the ephemeris involves first reading files to obtain arrays of coefficients *etc.*, followed by computation of the coordinates for a nominated time. The first phase, initializing a context data structure, is carried out by calling one of the functions `ephMooni` or `ephMoonc`. Once this context is available the present function can be called to compute the ephemeris.
4. The context is a data structure of type `ephMOONctx` supplied by the caller. It contains:
   - initialization status:
     0 : not yet initialized
     1 : `icor` value (1 for LLR or 2 for DE405)
   - ELP/MPP02 constants
   - ELP/MPP02 series

   Because the caller supplies the context structure, it is possible to have more than one in operation at once, though this will not as a rule be useful. The principal benefit of separating the context from the logic is to make reentrancy possible.

ephMoonc — *read Moon binary file* — ephMoonc

CALL :

```
j = ephMoonc ( path, icor, &c );
```

ACTION :

Read from a binary file the context needed for computation of Moon ephemerides.

GIVEN :

path	`char*`	path for filenames
icor	`int`	choice of corrections to the constants: `icor = 1` : LLR `icor = 2` : DE405

RETURNED :

c	`ephMOONctx*`	context structure

RETURNED (function value) :

	`int`	status: 0 = OK −1 = illegal `icor` −2 = file path+name too big else = file related errors

NOTES :

1. Generating the ephemeris involves first reading files to obtain arrays of coefficients *etc.*, followed by computation of the coordinates for a nominated time. The present function performs the first phase, initializing the context data structure by reading it directly from a binary file. Once this context is available a different function, `ephMoon`, can be called to compute the ephemeris data.
2. The context is a data structure of type `ephMOONctx` supplied by the caller. It contains:
   - initialization status:
     - 0 : not yet initialized
     - 1 : `icor` value (1 for LLR or 2 for DE405)
   - ELP/MPP02 constants
   - ELP/MPP02 series

   Because the caller supplies the context structure, it is possible to have more than one in operation at once, though this will not as a rule be useful. The principal benefit of separating the context from the logic is to make reentrancy possible.

3. The code explicitly declares the names of the files that contain the ELP/MPP02 contexts (for the LLR and DE405 options respectively), and these can be edited if necessary, for example to include the directory name. The latter is in any case supplied through the argument path, which can be the null string if the code has been edited to include the path or the files are in the current directory.

4. The theoretical values of some constants have to be corrected. There are two sets of corrections, and the choice is indicated by the argument `icor`:

   If `icor = 1`, the constants are fitted to LLR observations provided from 1970 to 2001; it is the default value;

   If `icor = 2`, the constants are fitted to the DE405 ephemeris over one century (1950-1960); the lunar angles $w1$, $w2$, $w3$ receive also additive corrections to the secular coefficients.

   Binary files for these two options exist and the appropriate one read when the present function is called.

REFERENCE :

Jean Chapront & Gerard Francou, *Lunar solution ELP version ELP/MPP02*, Observatoire de Paris, SYRTE Department, UMR 8630/CNRS, October 2002

ephMooni — *read context from lunar ephemeris ASCII file* — ephMooni

CALL :

```
j = ephMooni ( path, icor, &c );
```

ACTION :

Internal function: generate the context needed for computation of lunar ephemerides reading from an ELP/MPP02 theory ASCII file.

GIVEN :

path	`char*`	path for filenames
icor	`int`	choice of corrections to the constants: `icor = 1` : LLR `icor = 2` : DE405

RETURNED :

c	`ephMOONctx*`	context structure

RETURNED (function value) :

	`int`	status: 0 = OK −1 = illegal `icor` −2 = file path+name too big else = file related errors (Note 5)

NOTES :

1. Generating lunar ephemerides involves first reading files to obtain arrays of coefficients *etc.*, followed by computation of the coordinates for a nominated time. The present function performs the first phase, initializing a context data structure by reading and interpreting an ASCII file. Once this context is available a different function, `ephMoon`, can be called to compute the ephemeris.

2. The context data structure, of type `ephMOONctx`, contains:

   - initialization status:
     - 0 : not yet initialized
     - 1 : `icor` value (1 for LLR or 2 for DE405)
   - ELP/MPP02 constants
   - ELP/MPP02 series

   Because the caller supplies the context structure, it is possible to have more than one in operation at once, though this will not as a rule be useful. The principal benefit of separating the context from the logic is to make reentrancy possible.

3. The code explicitly declares the names of the six files that contain the ELP/MPP02 series, and these can be edited if necessary, for example to include the directory name. The latter is in any case supplied through the argument `path`, which can be the null string if the code has been edited to include the path or the files are in the current directory.

4. The theoretical values of some constants have to be corrected. There are two sets of corrections, and the choice is indicated by the argument `icor`:

   `icor = 1`: the constants are fitted to LLR observations provided from 1970 to 2001; it is the default value;

   `icor = 2`: the constants are fitted to the DE405 ephemeris over one century (1950-1960); the lunar angles $w1$, $w2$, $w3$ receive also additive corrections to the secular coefficients.

5. Several status values are file-related. For the three main-problem files, returned status -3 is an `open` error, -4 and -6 are `read` errors, -5 and -7 indicate wrong content, and -8 is a `close` error. For the three perturbations files, -9 is an `open` error, -10 and -12 are `read` errors, -11 and -13 indicate wrong content, and -14 is a `close` error. For more detail see the code itself.

REFERENCE :

Jean Chapront & Gerard Francou, *Lunar solution ELP version ELP/MPP02*, Observatoire de Paris, SYRTE Department, UMR 8630/CNRS, October 2002

---

ephPlanc *read planet binary file* **ephPlanc**

CALL :

```
j = ephPlanc ( ibody, path, &c );
```

ACTION :

Read from a binary file the context needed for computation of a planet's ephemerides.

GIVEN :

ibody	`int`	body index: 1 : Mercury 2 : Venus 3 : Earth-Moon barycenter 4 : Mars 5 : Jupiter 6 : Saturn 7 : Uranus 8 : Neptune
path	`char*`	path for filenames

RETURNED :

c	`ephPLANctx*`	context structure

RETURNED (function value) :

	`int`	status: 0 = OK −1 = illegal `ibody` −2 = file path+name too big else = file related errors

NOTES :

1. Generating the ephemeris involves first reading files (one per body) to obtain arrays of coefficients *etc.*, followed by computation of the coordinates for a nominated time. The present function performs the first phase, initializing the context data structure by reading it directly from a binary file. Once this context is available a different function, `ephPlan`, can be called to compute the ephemeris data.

2. The context is a data structure of type `ephPLANctx` supplied by the caller. It contains:

   - initialization status (0 = not yet initialized)
   - VSOP2010 constants
   - VSOP2010 series

In an application, ephemerides for successive bodies can be computed using a single context structure that is reinitialized repeatedly. Should the application need to compute ephemerides for more than one body at once, multiple context structures must be declared.

3. The code explicitly declares the names of the nine files that contain the contexts for each body, and these can be edited if necessary, for example to include the directory name. The latter can in any case be supplied through the argument `path`, which can be set to an empty string if the code has been edited to include the path or because the files are in the current directory.

REFERENCE :

J.-L. Simon, G. Francou, A. Fienga & H. Manche, *New analytical planetary theories VSOP2010 and TOP2013*, Astronomy & Astrophysics **557**, A49 (2013).

ephPlanet — *heliocentric PV of a planet* — ephPlanet

CALL :

```
j = ephPlanet ( ibody, &c, date, pv );
```

ACTION :

Heliocentric ICRS position and velocity of a planet.

GIVEN :

ibody	`int`	body index: 1 : Mercury 2 : Venus 3 : Earth-Moon barycenter 4 : Mars 5 : Jupiter 6 : Saturn 7 : Uranus 8 : Neptune
c	`ephPLANctx*`	context for chosen body (Notes 1,2)
date	`double`	TDB as an MJD (Note 3)

RETURNED :

pv	`double[2][3]`	heliocentric $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$ (ICRS, au, au/s)

RETURNED (function value) :

	`int`	status: $+1$ = warning: date outside 1-4000 CE 0 = OK -1 = uninitialized context -2 = context is for wrong body

NOTES :

1. Before calling the present function (as many times as necessary), a (single) call to `ephPlanc` (or alternatively `ephPlani`) must be made to populate the context `c`. If the application involves more than one body at once, in order to avoid costly re-reading of the ephemeris files, each body must have its own context variable.
2. The choice of body was made when the context `c` was populated (Note 1). The `ibody` argument serves only to allow a check that the context is for the correct body.
3. The date is a Modified Julian Date, JD−2400000.5. The time scale is TDB, but for most applications Terrestrial Time TT can be used without significant loss of accuracy (but not of course UTC).
4. The reference frame is heliocentric ICRS, roughly J2000.0 mean equinox and equator.

5. Applications requiring apparent place must allow for planetary aberration (by supplying a date that is one light-time before the present) and apply bias-precession-nutation to the results.

6. The context data structure `c`, of type `ephPLANctx`, contains:

   - initialization status (0 = not yet initialized)
   - VSOP2010 constants
   - VSOP2010 series

   In an application, ephemerides for successive bodies can be computed using a single context structure that is reinitialized repeatedly. Should the application need to compute ephemerides for more than one body at once, multiple context structures will need to be declared.

7. For the given time (in the TDB time scale) the function calculates the following elliptic variables:

   - semi-major axis a (au)
   - mean longitude λ (radians)
   - $k = e \cos \varpi$ (radians)
   - $h = e \sin \varpi$ (radians)
   - $q = \sin(i/2) \cos \Omega$ (radians)
   - $p = \sin(i/2) \sin \Omega$ (radians)

   where e is the eccentricity, ϖ is the longitude of perihelion, i is the inclination and Ω is the longitude of the ascending node. These six quantities are then used to predict ecliptic position and velocity vectors, which are then rotated onto ICRS axes (returned as array `pv`).

8. Comparisons with DE405 over the interval 2000-2100 gave the following results:

	position		*velocity*	
	RMS	worst	RMS	worst
Mercury	0.057	0.178	0.043	0.157
Venus	0.096	0.310	0.030	0.094
Earth	0.177	0.408	0.035	0.078
Mars	2.181	5.245	0.234	0.617
Jupiter	5.069	8.364	0.085	0.132
Saturn	9.854	20.354	0.067	0.133
Uranus	20.996	49.481	0.033	0.077
Neptune	33.799	66.281	0.042	0.079
	km	*km*	*mm/s*	*mm/s*

   The comparisons with DE430 were as follows:

	position		*velocity*	
	RMS	worst	RMS	worst
Mercury	2.025	3.259	1.767	3.619
Venus	0.791	1.184	0.256	0.379
Earth	0.964	1.322	0.192	0.267
Mars	3.405	5.028	0.365	0.609
Jupiter	306.242	515.699	5.150	8.951
Saturn	661.195	989.748	3.793	5.423
Uranus	4778.612	7684.369	11.047	17.113
Neptune	24117.350	40137.632	26.750	42.459
	km	*km*	*mm/s*	*mm/s*

REFERENCES :

1. X. Moisson, P. Bretagnon, Celestial Mechanics & Dynamical Astronomy (2001) **80**, 205.
2. J.-L. Simon, G. Francou, A. Fienga & H. Manche, *New analytical planetary theories VSOP2010 and TOP2013*, Astronomy & Astrophysics **557**, A49 (2013).

ephPlani *read context from planetary ephemeris ASCII file* ephPlani

CALL :

```
j = ephPlani ( path, ibody, &c );
```

ACTION :

Internal function: generate the context needed for computation of planetary ephemerides reading from a VSOP2010 theory ASCII file.

GIVEN :

path	`char*`	path for filenames
ibody	`int`	body index:
		`ibody = 1` : Mercury
		`ibody = 2` : Venus
		`ibody = 3` : Earth-Moon barycenter
		`ibody = 4` : Mars
		`ibody = 5` : Jupiter
		`ibody = 6` : Saturn
		`ibody = 7` : Uranus
		`ibody = 8` : Neptune

RETURNED :

c	`ephPLANctx*`	context structure

RETURNED (function value) :

`int`	status:	0 = OK
		-1 = illegal `ibody`
		-2 = file path+name too big
		-3 = file open error
		-4 = file content error
		else = file read errors

NOTES :

1. Generating planetary ephemerides involves first reading files (one per body) to obtain arrays of coefficients *etc.*, followed by computation of the coordinates for a nominated time. The present function performs the first phase, initializing a context data structure for the chosen body by reading and interpreting an ASCII file. Once this context is available a different function, `ephPlan`, can be called to compute the ephemeris data.

2. The context data structure, of type `ephPLANctx`, contains:

   - initialization status (false = not yet initialized)

- VSOP2010 constants
- VSOP2010 series

In an application, ephemerides for successive bodies can be computed using a single context structure that is reinitialized repeatedly. Should the application need to compute ephemerides for more than one body at once, multiple context structures must be declared.

3. The code explicitly declares the names of the nine files that contain the VSOP2010 series, and these can be edited if necessary, for example to include the directory name. The latter can in any case be supplied through the argument `path`, which can be set to an empty string if the code has been edited to include the path or because the files are in the current directory.

REFERENCE :

J.-L. Simon, G. Francou, A. Fienga & H. Manche, *New analytical planetary theories VSOP2010 and TOP2013*, Astronomy & Astrophysics **557**, A49 (2013).

ephRdplan — $[\alpha, \delta]$ *and diameter of a planet* — ephRdplan

CALL :

```
j = ephRdplan ( &cmoon, &cemb, &cplan, ut1, tdb, np, elong, phi, hm,
                              &rast, &dast, &rapp, &dapp, &eo, &diam );
```

ACTION :

Topocentric apparent $[\alpha, \delta]$ of a planet (also Sun and Moon), and its angular diameter.

GIVEN :

cmoon	`ephMOONctx*`	context for Moon (Note 2)
cemb	`ephPLANctx*`	context for Earth-Moon barycenter (Note 2)
cplan	`ephPLANctx*`	context for planet (Notes 2-4)
ut1	`double`	UT1 (MJD, Note 5)
tdb	`double`	TDB (MJD, Note 5)
np	`int`	body:
		1 = Mercury
		2 = Venus
		3 = Moon
		4 = Mars
		5 = Jupiter
		6 = Saturn
		7 = Uranus
		8 = Neptune
		else = Sun
elong	`double`	observer east longitude (radians)
phi	`double`	observer geodetic latitude (radians)
hm	`double`	observer height above sea level (meters)

RETURNED :

rast	`double*`	right ascension, ICRS astrometric (radians)
dast	`double*`	declination, ICRS astrometric (radians)
rapp	`double*`	right ascension, topocentric apparent (radians)
dapp	`double*`	declination, topocentric apparent (radians)
eo	`double*`	equation of the origins (radians, Note 7)
diam	`double*`	angular diameter, equatorial (radians)

RETURNED (function value) :

`int`	status: +1 = warning: date outside 1-4000 CE
	0 = OK
	−1 = uninitialized context

NOTES :

1. The ephemerides are calculated using the ELP/MPP02 + VSOP2010 theories. For the 21st century the results agree with JPL DE405 to the following geocentric accuracies:

	RMS	worst
Sun	0.243	0.559
Mercury	0.265	0.858
Venus	0.374	3.352
Moon	4.998	9.651
Mars	2.428	16.957
Jupiter	1.355	2.789
Saturn	1.435	3.294
Uranus	1.426	3.543
Neptune	1.433	2.916
	mas	*mas*

Comparisons with JPL DE430 are as follows:

	RMS	worst
Sun	1.329	1.841
Mercury	2.810	6.883
Venus	1.648	4.553
Moon	11.048	16.561
Mars	3.882	19.072
Jupiter	82.013	177.075
Saturn	92.820	164.302
Uranus	337.295	555.496
Neptune	1065.73	1883.56
	mas	*mas*

2. The context arguments point to tables of constants that are used by the lunar and planetary ephemeris functions `ephMoon` and `ephPlanet`. Before the function is called, these three context tables must be populated, by calling the function `ephMoonc` in the case of `cmoon` (the Moon) and `ephPlanc` in the cases of `cemb` and `cplan` (the Earth-Moon barycenter and the chosen planet respectively).

3. It is the caller's responsibility to populate the `cplan` context for the same planet that is identified by the argument `np`.

4. For `np` = 3 (Moon) or `np` = else (Sun), `cplan` is not used and can be left uninitialized. For `np` = 1 or 2 or 4-9, all three contexts must be populated.

5. The date of observation must be supplied in two forms, namely UT1 and TDB, both in the form of Modified Julian Date (JD−2400000.5). Depending on the application, liberties can be taken: usually TT will do instead of TDB and UTC instead of UT1, and in some very low-precision applications UTC might be adequate for both. The particular need for Earth-rotation time as opposed to what used to be called "ephemeris time" is for the accurate computation of lunar geocentric parallax.

6. The observer location coordinates (`elongm`, `phi`, `hm`) allow correction for geocentric parallax. This is a major effect for the Moon, but its effect on planetary positions is small (for some applications negligible, especially for the outer planets). Geocentric positions can be generated by specifying `hm` below -1000.0, for example "`-1e6`".

7. The "current" $[\,\alpha, \delta\,]$s returned are topocentric apparent. For topocentric intermediate add the equation of the origins, `eo`.

8. See also the function `ephRdplanq`, which is more complicated to call but will be faster in cases where more than one target is involved.

ephRdplanq — $[\alpha, \delta]$ *and diameter of a planet* — ephRdplanq

CALL :

```
j = ephRdplanq ( pvgm, pvsb, &cplan, ut1, tdb, np, elong, phi, hm,
                          &rast, &dast, &rapp, &dapp, &eo, &diam );
```

ACTION :

Topocentric apparent $[\alpha, \delta]$ of a planet (also Sun and Moon), and its angular diameter. This "quick" version starts with precomputed position+velocity vectors for geocenter-to-Moon and Sun-to-EMB respectively, so that computationally expensive ephemeris calculations involve the target body alone.

GIVEN :

pvgm	`double[2][3]`	geocenter-to-Moon PV (ICRS, au, au/s, Note 2)
pvsb	`double[2][3]`	Sun-to-EMB PV (ICRS, au, au/s, Note 2)
cplan	`ephPLANctx*`	context for planet ephemeris (Notes 3-5)
ut1	`double`	UT1 (MJD, Note 6)
tdb	`double`	TDB (MJD, Note 6)
np	`int`	body: 1 = Mercury 2 = Venus 3 = Moon 4 = Mars 5 = Jupiter 6 = Saturn 7 = Uranus 8 = Neptune else = Sun
elong	`double`	observer east longitude (radians)
phi	`double`	observer geodetic latitude (radians)
hm	`double`	observer height above sea level (meters)

RETURNED :

rast	`double*`	right ascension, ICRS astrometric (radians)
dast	`double*`	declination, ICRS astrometric (radians)
rapp	`double*`	right ascension, topocentric apparent (radians)
dapp	`double*`	declination, topocentric apparent (radians)
eo	`double*`	equation of the origins (radians, Note 8)
diam	`double*`	angular diameter, equatorial (radians)

RETURNED (function value) :

`int`	status: +1 = warning: date outside 1-4000 CE 0 = OK −1 = uninitialized context

NOTES :

1. The ephemerides are calculated using the ELP/MPP02 + VSOP2010 theories. For the 21st century the results agree with JPL DE405 to the following geocentric accuracies:

	RMS	worst
Sun	0.243	0.559
Mercury	0.265	0.858
Venus	0.374	3.352
Moon	4.998	9.651
Mars	2.428	16.957
Jupiter	1.355	2.789
Saturn	1.435	3.294
Uranus	1.426	3.543
Neptune	1.433	2.916
	mas	*mas*

   Comparisons with JPL DE430 are as follows:

	RMS	worst
Sun	1.329	1.841
Mercury	2.810	6.883
Venus	1.648	4.553
Moon	11.048	16.561
Mars	3.882	19.072
Jupiter	82.013	177.075
Saturn	92.820	164.302
Uranus	337.295	555.496
Neptune	1065.73	1883.56
	mas	*mas*

2. The geocenter-to-Moon and Sun-to-EMB position+velocity vectors `pvgm` and `pvsb` can be computed by calling the functions `ephMoon` and `ephPlanet` respectively.

3. The context argument `cplan` points to tables of constants that are used by the lunar and planetary ephemeris function `ephPlanet`. Before the function is called, this must be populated, by calling the function `ephPlanc`.

4. It is the caller's responsibility to populate the `cplan` context for the same planet that is identified by the argument `np`.

5. For `np` = 3 (Moon) or `np` = else (Sun), `cplan` is not used and can be left uninitialized. For `np` = 1 or 2 or 4-9, all three contexts must be populated.

6. The date of observation must be supplied in two forms, namely UT1 and TDB, both in the form of Modified Julian Date (JD−2400000.5). Depending on the application, liberties can be taken: usually TT will do instead of TDB and UTC instead of UT1, and in some very low-precision applications UTC might be adequate for both. The particular need for Earth-rotation time as opposed to what used to be called "ephemeris time" is for the accurate computation of lunar geocentric parallax.

7. The observer location coordinates (`elongm`, `phi`, `hm`) allow correction for geocentric parallax. This is a major effect for the Moon, but its effect on planetary positions is small (for some applications negligible, especially for the outer planets). Geocentric positions can be generated by specifying `hm` below -1000.0, for example "`-1e6`".

8. The "current" $[\,\alpha, \delta\,]$s returned are topocentric apparent. For topocentric intermediate add the equation of the origins, `eo`.

9. See also the function `ephRdplan`, which does not need precomputed Moon and EMB vectors but consumes more computing power.

6 Files

The following lists all the EPH files and the purpose of each one.

`bopr.c`	C source code for the `ephBopr` function
`earth.c`	C source code for the `ephEarth` function
`elpmpp02.pdf`	document that came with ELP/MPP02
`elpmpp02_readme.pdf`	document that came with ELP/MPP02
`elp_main.s1`	ELP/MPP02 ASCII file, main series longitude
`elp_main.s2`	ELP/MPP02 ASCII file, main series latitude
`elp_main.s3`	ELP/MPP02 ASCII file, main series distance
`ELP_MPP02_JPL.ctx`	EPH binary file for lunar ephemeris (JPL DE405 variant)
`ELP_MPP02_LLR.ctx`	EPH binary file for lunar ephemeris (LLR variant)
`elp_pert.s1`	ELP/MPP02 ASCII file, perturbations longitude
`elp_pert.s2`	ELP/MPP02 ASCII file, perturbations latitude
`elp_pert.s3`	ELP/MPP02 ASCII file, perturbations distance
`eph.h`	header file for EPH library
`eph.pdf`	user documentation PDF file
`eph.tex`	LaTeX source for user documentation
`makefile`	Unix make file for building and installing the EPH library
`moon.c`	C source code for the `ephMoonc` function
`moon_bin.c`	C source code for the MOON_BIN utility
`moonc.c`	C source code for the `ephMoonc` function
`mooni.c`	C source code for the `ephMooni` function
`occultation.c`	C source code for the OCCULTATION application
`pc.bat`	example MS Windows script for building EPH
`plan_bin.c`	C source code for the PLAN_BIN utility
`planc.c`	C source code for the `ephPlanc` function
`planet.c`	C source code for the `ephPlanet` function
`planets.c`	C source code for the PLANETS application
`plani.c`	C source code for the `ephPlani` function
`rdplan.c`	C source code for the `ephRdplan` function
`rdplanq.c`	C source code for the `ephRdplanq` function
`VSOP2010.pdf`	document that came with VSOP2010
`VSOP2010_1.ctx`	EPH binary file for Mercury ephemeris
`VSOP2010_2.ctx`	EPH binary file for Venus ephemeris
`VSOP2010_3.ctx`	EPH binary file for Earth-Moon barycenter ephemeris
`VSOP2010_4.ctx`	EPH binary file for Mars ephemeris
`VSOP2010_5.ctx`	EPH binary file for Jupiter ephemeris
`VSOP2010_6.ctx`	EPH binary file for Saturn ephemeris
`VSOP2010_7.ctx`	EPH binary file for Uranus ephemeris
`VSOP2010_8.ctx`	EPH binary file for Neptune ephemeris
`VSOP2010_readme.pdf`	document that came with VSOP2010
`VSOP2010p1.dat`	VSOP2010 ASCII file for Mercury
`VSOP2010p2.dat`	VSOP2010 ASCII file for Venus
`VSOP2010p3.dat`	VSOP2010 ASCII file for Earth-Moon barycenter
`VSOP2010p4.dat`	VSOP2010 ASCII file for Mars

`VSOP2010p5.dat`	VSOP2010 ASCII file for Jupiter
`VSOP2010p6.dat`	VSOP2010 ASCII file for Saturn
`VSOP2010p7.dat`	VSOP2010 ASCII file for Uranus
`VSOP2010p8.dat`	VSOP2010 ASCII file for Neptune