THE

```
        SSSSS           OOOOOO       FFFFFFFFFFFFF      AAAAAAA
     SSSSSSSSSS     OOOOOOOOOOOO     FFFFFFFFFFFF      AAAAAAAA
    SSSSSSSSSSS   OOOOOOOOOOOOOO    FFFFFFFFFFFF      AAAA AAAA
   SSSS      S   OOOOOO    OOOOO    FFFF             AAAA  AAAA
  SSSSS         OOOOO       OOOO   FFFFF            AAAA   AAAA
  SSSSSSSSSS    OOOO       OOOOO   FFFFFFFFFFFF    AAAA    AAAA
    SSSSSSSSS  OOOOO       OOOO   FFFFFFFFFFFF    AAAAAAAAAAAAA
        SSSSS  OOOO       OOOO    FFFF           AAAAAAAAAAAAAA
 S      SSSS   OOOOO    OOOOO    FFFF           AAAAAAAAAAAAAAA
SSSSSSSSSSS    OOOOOOOOOOOOO     FFFF          AAAA       AAAAA
SSSSSSSSS       OOOOOOOOOO      FFFF          AAAA        AAAAA
  SSSS            OOOOO         FFFF         AAAA         AAAAA
```

SOFTWARE

LIBRARIES

International Astronomical Union

Division A: Fundamental Astronomy

Standards Of Fundamental Astronomy Board

Release 14

2018 January 1

CONTENTS

THE IAU-SOFA SOFTWARE LIBRARIES

SOFA stands for "Standards Of Fundamental Astronomy". The SOFA software libraries are a collection of subprograms, in source-code form, which implement official IAU algorithms for fundamental-astronomy computations. The subprograms at present comprise 185 "astronomy" routines supported by 55 "vector/matrix" routines, available in both Fortran77 and C implementations.

THE SOFA INITIATIVE

SOFA is an IAU Service which operates as a Standing Working Group under Division A (Fundamental Astronomy).

The IAU set up the SOFA initiative at the 1994 General Assembly, to promulgate an authoritative set of fundamental-astronomy constants and algorithms. At the subsequent General Assembly, in 1997, the appointment of a review board and the selection of a site for the SOFA Center (the outlet for SOFA products) were announced.

The SOFA initiative was originally proposed by the IAU Working Group on Astronomical Standards (WGAS), under the chairmanship of Toshio Fukushima. The proposal was for "...new arrangements to establish and maintain an accessible and authoritative set of constants, algorithms and procedures that implement standard models used in fundamental astronomy". The SOFA Software Libraries implement the "algorithms" part of the SOFA initiative. They were developed under the supervision of an international panel called the SOFA Board. The current membership of this panel is listed in an appendix.

A feature of the original SOFA software proposals was that the products would be self-contained and not depend on other software. This includes basic documentation, which, like the present file, will mostly be plain ASCII text. It should also be noted that there is no assumption that the software will be used on a particular computer and Operating System. Although OS-related facilities may be present (Unix make files for instance, use by the SOFA Center of automatic code management systems, HTML versions of some documentation), the routines themselves will be visible as individual text files and will run on a variety of platforms.

ALGORITHMS

The SOFA Board’s initial goal has been to create a set of callable subprograms. Whether "subroutines" or "functions", they are all referred to simply as "routines". They are designed for use by software developers wishing to write complete applications; no runnable, free-standing applications are included in SOFA’s present plans.

The algorithms are drawn from a variety of sources. Because most of the routines so far developed have either been standard "text-book" operations or implement well-documented standard algorithms, it has not been necessary to invite the whole community to submit algorithms, though consultation with authorities has occurred where necessary. It should also be noted that consistency with the conventions published by the International Earth Rotation Service was a stipulation in the original SOFA proposals, further constraining the software designs. This state of affairs will continue to exist for some time, as there is a large backlog of agreed extensions to work on. However, in the future the Board may decide to call for proposals, and is in the meantime willing to look into any suggestions that are received by the SOFA Center.

SCOPE

The routines currently available are listed in the next two chapters of this document.

The "astronomy" library comprises 185 routines (plus one obsolete Fortran routine that now appears under a revised name). The areas addressed include calendars, astrometry, time scales, Earth rotation, ephemerides, precession−nutation, star catalog transformations, gnomonic projection, horizon/equatorial transformations and geodetic/geocentric transformations.

The "vector−matrix" library, comprising 55 routines, contains a collection of simple tools for manipulating the vectors, matrices and angles used by the astronomy routines.

There is no explicit commitment by SOFA to support historical models, though as time goes on a legacy of superseded models will naturally accumulate. There is, for example, no support of B1950/FK4 star coordinates, or pre−1976 precession models, though these capabilities could be added were there significant demand.

Though the SOFA software libraries are rather limited in scope, and are likely to remain so for a considerable time, they do offer distinct advantages to prospective users. In particular, the routines are:

* authoritative: they are IAU−backed and have been constructed with great care;

* practical: they are straightforward to use in spite of being precise and rigorous (to some stated degree);

* accessible and supported: they are downloadable from an easy−to−find place, they are in an integrated and consistent form, they come with adequate internal documentation, and help for users is available.

VERSIONS

Once it has been published, an issue is never revised or updated, and remains accessible indefinitely. Subsequent issues may, however, include corrected versions under the original routine name and filenames. However, where a different model is introduced, it will have a different name.

The issues will be referred to by the date when they were announced. The frequency of re−issue will be decided by the Board, taking into account the importance of the changes and the impact on the user community.

DOCUMENTATION

At present there is little free−standing documentation about individual routines. However, each routine has preamble comments which specify in detail what the routine does and how it is used.

The files sofa_pn_f.pdf and sofa_pn_c.pdf (for Fortran and C users respectively) describe the SOFA tools for precession−nutation and other aspects of Earth attitude, and include example code and, in an appendix, diagrams showing the interrelationships between the routines supporting the latest (IAU 2006/2000A) models. Two other pairs of documents introduce time scale transformations (sofa_ts_f.pdf and sofa_ts_c.pdf) and astrometric transformations (sofa_ast_f.pdf and sofa_ast_c.pdf).

PROGRAMMING LANGUAGES AND STANDARDS

The SOFA routines are available in two programming languages at present: Fortran77 and ANSI C. Related software in other languages is under consideration.

The Fortran code conforms to ANSI X3.9−1978 in all but two minor respects: each has an IMPLICIT NONE declaration, and its name has a prefix of "iau_" and may be longer than 6 characters. A global edit to erase both of these will produce ANSI-compliant code with no change in its function.

Coding style, and restrictions on the range of language features, have been much debated by the Board, and the results comply with the majority view. There is (at present) no document that defines the standards, but the code itself offers a wide range of examples of what is acceptable.

The Fortran routines contain explicit numerical constants (the INCLUDE statement is not part of ANSI Fortran77). These are drawn from the file consts.lis, which is listed in an appendix. Constants for the SOFA/C functions are defined in a header file sofam.h.

The naming convention is such that a SOFA routine referred to generically as "EXAMPL" exists as a Fortran subprogram iau_EXAMPL and a C function iauExampl. The calls for the two versions are very similar, with the same arguments in the same order. In a few cases, the C equivalent of a Fortran SUBROUTINE subprogram uses a return value rather than an argument.

Each language version includes a "testbed" main-program that can be used to verify that the SOFA routines have been correctly compiled on the end user's system. The Fortran and C versions are called t_sofa_f.for and t_sofa_c.c respectively. The testbeds execute every SOFA routine and check that the results are within expected accuracy margins. It is not possible to guarantee that all platforms will meet the rather stringent criteria that have been used, and an occasional warning message may be encountered on some systems.

COPYRIGHT ISSUES

Copyright for all of the SOFA software and documentation is owned by the IAU SOFA Board. The Software is made available free of charge for all classes of user, including commercial. However, there are strict rules designed to avoid unauthorized variants coming into circulation. It is permissible to distribute derived works and other modifications, but they must be clearly marked to avoid confusion with the SOFA originals.

Further details are included in the block of comments which concludes every routine. The text is also set out in an appendix to the present document.

ACCURACY

The SOFA policy is to organize the calculations so that the machine accuracy is fully exploited. The gap between the precision of the underlying model or theory and the computational resolution has to be kept as large as possible, hopefully leaving several orders of magnitude of headroom.

The SOFA routines in some cases involve design compromises between rigor and ease of use (and also speed, though nowadays this is seldom a major concern).

ACKNOWLEDGEMENTS

The Board is indebted to a number of contributors, who are acknowledged in the preamble comments of the routines concerned.

The Board's effort is provided by the members' individual institutes.

Resources for operating the SOFA Center are provided by Her Majesty's Nautical Almanac Office, operated by the United Kingdom Hydrographic Office.

SOFA Astronomy Library

PREFACE

The routines described here comprise the SOFA astronomy library. Their general appearance and coding style conforms to conventions agreed by the SOFA Board, and their functions, names and algorithms have been ratified by the Board. Procedures for soliciting and agreeing additions to the library are still evolving.

PROGRAMMING LANGUAGES

The SOFA routines are available in two programming languages at present: Fortran 77 and ANSI C.

Except for a single obsolete Fortran routine, which has no C equivalent, there is a one−to−one relationship between the two language versions. The naming convention is such that a SOFA routine referred to generically as "EXAMPL" exists as a Fortran subprogram iau_EXAMPL and a C function iauExampl. The calls for the two versions are very similar, with the same arguments in the same order. In a few cases, the C equivalent of a Fortran SUBROUTINE subprogram uses a return value rather than an argument.

GENERAL PRINCIPLES

The principal function of the SOFA Astronomy Library is to provide definitive algorithms. A secondary function is to provide software suitable for convenient direct use by writers of astronomical applications.

The astronomy routines call on the SOFA vector/matrix library routines, which are separately listed.

The routines are designed to exploit the full floating−point accuracy of the machines on which they run, and not to rely on compiler optimizations. Within these constraints, the intention is that the code corresponds to the published formulation (if any).

Dates are always Julian Dates (except in calendar conversion routines) and are expressed as two double precision numbers which sum to the required value.

A distinction is made between routines that implement IAU−approved models and those that use those models to create other results. The former are referred to as "canonical models" in the preamble comments; the latter are described as "support routines".

Using the library requires knowledge of positional astronomy and time−scales. These topics are covered in "Explanatory Supplement to the Astronomical Almanac", 3rd Edition, Sean E. Urban & P. Kenneth Seidelmann (eds.), University Science Books, 2013. Recent developments are documented in the scientific journals, and references to the relevant papers are given in the SOFA code as required. The IERS Conventions are also an essential reference. The routines concerned with Earth attitude (precession−nutation etc.) are described in the SOFA document sofa_pn.pdf. Those concerned with transformations between different time scales are described in sofa_ts_f.pdf (Fortran) and sofa_ts_c.pdf (C). Those concerned with astrometric transformations are described in sofa_ast_f.pdf (Fortran) and sofa_ast_c (C).

ROUTINES

Calendars

CAL2JD	Gregorian calendar to Julian Day number
EPB	Julian Date to Besselian Epoch
EPB2JD	Besselian Epoch to Julian Date
EPJ	Julian Date to Julian Epoch
EPJ2JD	Julian Epoch to Julian Date
JD2CAL	Julian Date to Gregorian year, month, day, fraction
JDCALF	Julian Date to Gregorian date for formatted output

Astrometry

AB	apply stellar aberration
APCG	prepare for ICRS <-> GCRS, geocentric, special
APCG13	prepare for ICRS <-> GCRS, geocentric
APCI	prepare for ICRS <-> CIRS, terrestrial, special
APCI13	prepare for ICRS <-> CIRS, terrestrial
APCO	prepare for ICRS <-> observed, terrestrial, special
APCO13	prepare for ICRS <-> observed, terrestrial
APCS	prepare for ICRS <-> CIRS, space, special
APCS13	prepare for ICRS <-> CIRS, space
APER	insert ERA into context
APER13	update context for Earth rotation
APIO	prepare for CIRS <-> observed, terrestrial, special
APIO13	prepare for CIRS <-> observed, terrestrial
ATCI13	catalog -> CIRS
ATCIQ	quick ICRS -> CIRS
ATCIQN	quick ICRS -> CIRS, multiple deflections
ATCIQZ	quick astrometric ICRS -> CIRS
ATCO13	ICRS -> observed
ATIC13	CIRS -> ICRS
ATICQ	quick CIRS -> ICRS
ATCIQN	quick CIRS -> ICRS, multiple deflections
ATIO13	CIRS -> observed
ATIOQ	quick CIRS -> observed
ATOC13	observed -> astrometric ICRS
ATOI13	observed -> CIRS
ATOIQ	quick observed -> CIRS
LD	light deflection by a single solar-system body
LDN	light deflection by multiple solar-system bodies
LDSUN	light deflection by the Sun
PMPX	apply proper motion and parallax
PMSAFE	apply proper motion, with zero-parallax precautions
PVTOB	observatory position and velocity
PVSTAR	space motion pv-vector to star catalog data
REFCO	refraction constants
STARPM	apply proper motion
STARPV	star catalog data to space motion pv-vector

Time scales

D2DTF	format 2-part JD for output
DAT	Delta(AT) (=TAI-UTC) for a given UTC date
DTDB	TDB-TT
DTF2D	encode time and date fields into 2-part JD
TAITT	TAI to TT
TAIUT1	TAI to UT1
TAIUTC	TAI to UTC
TCBTDB	TCB to TDB
TCGTT	TCG to TT
TDBTCB	TDB to TCB
TDBTT	TDB to TT
TTTAI	TT to TAI
TTTCG	TT to TCG
TTTDB	TT to TDB
TTUT1	TT to UT1
UT1TAI	UT1 to TAI
UT1TT	UT1 to TT
UT1UTC	UT1 to UTC
UTCTAI	UTC to TAI
UTCUT1	UTC to UT1

Earth rotation angle and sidereal time

EE00	equation of the equinoxes, IAU 2000

EE00A	equation of the equinoxes, IAU 2000A
EE00B	equation of the equinoxes, IAU 2000B
EE06A	equation of the equinoxes, IAU 2006/2000A
EECT00	equation of the equinoxes complementary terms, IAU 2000
EQEQ94	equation of the equinoxes, IAU 1994
ERA00	Earth rotation angle, IAU 2000
GMST00	Greenwich mean sidereal time, IAU 2000
GMST06	Greenwich mean sidereal time, IAU 2006
GMST82	Greenwich mean sidereal time, IAU 1982
GST00A	Greenwich apparent sidereal time, IAU 2000A
GST00B	Greenwich apparent sidereal time, IAU 2000B
GST06	Greenwich apparent ST, IAU 2006, given NPB matrix
GST06A	Greenwich apparent sidereal time, IAU 2006/2000A
GST94	Greenwich apparent sidereal time, IAU 1994

Ephemerides (limited precision)

EPV00	Earth position and velocity
PLAN94	major-planet position and velocity

Precession, nutation, polar motion

BI00	frame bias components, IAU 2000
BP00	frame bias and precession matrices, IAU 2000
BP06	frame bias and precession matrices, IAU 2006
BPN2XY	extract CIP X,Y coordinates from NPB matrix
C2I00A	celestial-to-intermediate matrix, IAU 2000A
C2I00B	celestial-to-intermediate matrix, IAU 2000B
C2I06A	celestial-to-intermediate matrix, IAU 2006/2000A
C2IBPN	celestial-to-intermediate matrix, given NPB matrix, IAU 2000
C2IXY	celestial-to-intermediate matrix, given X,Y, IAU 2000
C2IXYS	celestial-to-intermediate matrix, given X,Y and s
C2T00A	celestial-to-terrestrial matrix, IAU 2000A
C2T00B	celestial-to-terrestrial matrix, IAU 2000B
C2T06A	celestial-to-terrestrial matrix, IAU 2006/2000A
C2TCIO	form CIO-based celestial-to-terrestrial matrix
C2TEQX	form equinox-based celestial-to-terrestrial matrix
C2TPE	celestial-to-terrestrial matrix given nutation, IAU 2000
C2TXY	celestial-to-terrestrial matrix given CIP, IAU 2000
EO06A	equation of the origins, IAU 2006/2000A
EORS	equation of the origins, given NPB matrix and s
FW2M	Fukushima-Williams angles to r-matrix
FW2XY	Fukushima-Williams angles to X,Y
LTP	long-term precession matrix
LTPB	long-term precession matrix, including ICRS frame bias
LTPECL	long-term precession of the ecliptic
LTPEQU	long-term precession of the equator
NUM00A	nutation matrix, IAU 2000A
NUM00B	nutation matrix, IAU 2000B
NUM06A	nutation matrix, IAU 2006/2000A
NUMAT	form nutation matrix
NUT00A	nutation, IAU 2000A
NUT00B	nutation, IAU 2000B
NUT06A	nutation, IAU 2006/2000A
NUT80	nutation, IAU 1980
NUTM80	nutation matrix, IAU 1980
OBL06	mean obliquity, IAU 2006
OBL80	mean obliquity, IAU 1980
PB06	zeta,z,theta precession angles, IAU 2006, including bias
PFW06	bias-precession Fukushima-Williams angles, IAU 2006
PMAT00	precession matrix (including frame bias), IAU 2000
PMAT06	PB matrix, IAU 2006
PMAT76	precession matrix, IAU 1976
PN00	bias/precession/nutation results, IAU 2000
PN00A	bias/precession/nutation, IAU 2000A
PN00B	bias/precession/nutation, IAU 2000B
PN06	bias/precession/nutation results, IAU 2006
PN06A	bias/precession/nutation results, IAU 2006/2000A
PNM00A	classical NPB matrix, IAU 2000A
PNM00B	classical NPB matrix, IAU 2000B
PNM06A	classical NPB matrix, IAU 2006/2000A
PNM80	precession/nutation matrix, IAU 1976/1980
P06E	precession angles, IAU 2006, equinox based

POM00	polar motion matrix
PR00	IAU 2000 precession adjustments
PREC76	accumulated precession angles, IAU 1976
S00	the CIO locator s, given X,Y, IAU 2000A
S00A	the CIO locator s, IAU 2000A
S00B	the CIO locator s, IAU 2000B
S06	the CIO locator s, given X,Y, IAU 2006
S06A	the CIO locator s, IAU 2006/2000A
SP00	the TIO locator s’, IERS 2003
XY06	CIP, IAU 2006/2000A, from series
XYS00A	CIP and s, IAU 2000A
XYS00B	CIP and s, IAU 2000B
XYS06A	CIP and s, IAU 2006/2000A

Fundamental arguments for nutation etc.

FAD03	mean elongation of the Moon from the Sun
FAE03	mean longitude of Earth
FAF03	mean argument of the latitude of the Moon
FAJU03	mean longitude of Jupiter
FAL03	mean anomaly of the Moon
FALP03	mean anomaly of the Sun
FAMA03	mean longitude of Mars
FAME03	mean longitude of Mercury
FANE03	mean longitude of Neptune
FAOM03	mean longitude of the Moon’s ascending node
FAPA03	general accumulated precession in longitude
FASA03	mean longitude of Saturn
FAUR03	mean longitude of Uranus
FAVE03	mean longitude of Venus

Star catalog conversions

FK52H	transform FK5 star data into the Hipparcos system
FK5HIP	FK5 to Hipparcos rotation and spin
FK5HZ	FK5 to Hipparcos assuming zero Hipparcos proper motion
H2FK5	transform Hipparcos star data into the FK5 system
HFK5Z	Hipparcos to FK5 assuming zero Hipparcos proper motion

Ecliptic coordinates

ECEQ06	ecliptic to ICRS, IAU 2006
ECM06	rotation matrix, ICRS to ecliptic, IAU 2006
EQEC06	ICRS to ecliptic, IAU 2006
LTECEQ	ecliptic to ICRS, long term
LTECM	rotation matrix, ICRS to ecliptic, long−term
LTEQEC	ICRS to ecliptic, long term

Galactic coordinates

G2ICRS	transform IAU 1958 galactic coordinates to ICRS
ICRS2G	transform ICRS coordinates to IAU 1958 Galactic

Geodetic/geocentric

EFORM	a,f for a nominated Earth reference ellipsoid
GC2GD	geocentric to geodetic for a nominated ellipsoid
GC2GDE	geocentric to geodetic given ellipsoid a,f
GD2GC	geodetic to geocentric for a nominated ellipsoid
GD2GCE	geodetic to geocentric given ellipsoid a,f

Gnomonic projection

TPORS	solve for tangent point, spherical
TPORV	solve for tangent point, vector
TPSTS	deproject tangent plane to celestial, spherical
TPSTV	deproject tangent plane to celestial, vector
TPXES	project celestial to tangent plane, spherical
TPXEV	project celestial to tangent plane, vector

Horizon/equatorial

AE2HD	(azimuth, altitude) to (hour angle, declination)

```
     HD2AE     (hour angle, declination) to (azimuth, altitude)
     HD2PA     parallactic angle

  Obsolete

     C2TCEO    former name of C2TCIO


CALLS: FORTRAN VERSION

   CALL iau_AB     ( PNAT, V, S, BM1, PPR )
   CALL iau_AE2HD  ( AZ, EL, PHI, HA, DEC )
   CALL iau_APCG   ( DATE1, DATE2, EB, EH, ASTROM )
   CALL iau_APCG13 ( DATE1, DATE2, ASTROM )
   CALL iau_APCI   ( DATE1, DATE2, EB, EH, X, Y, S, ASTROM )
   CALL iau_APCI13 ( DATE1, DATE2, ASTROM, EO )
   CALL iau_APCO   ( DATE1, DATE2, EB, EH, X, Y, S,
  :                  THETA, ELONG, PHI, HM, XP, YP, SP,
  :                  REFA, REFB, ASTROM )
   CALL iau_APCO13 ( UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
  :                  PHPA, TC, RH, WL, ASTROM, EO, J )
   CALL iau_APCS   ( DATE1, DATE2, PV, EB, EH, ASTROM )
   CALL iau_APCS13 ( DATE1, DATE2, PV, ASTROM )
   CALL iau_APER   ( THETA, ASTROM )
   CALL iau_APER13 ( UT11, UT12, ASTROM )
   CALL iau_APIO   ( SP, THETA, ELONG, PHI, HM, XP, YP,
  :                  REFA, REFB, ASTROM )
   CALL iau_APIO13 ( UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
  :                  PHPA, TC, RH, WL, ASTROM, J )
   CALL iau_ATCI13 ( RC, DC, PR, PD, PX, RV, DATE1, DATE2, RI, DI, EO )
   CALL iau_ATCIQ  ( RC, DC, PR, PD, PX, RV, ASTROM, RI, DI )
   CALL iau_ATCIQN ( RC, DC, PR, PD, PX, RV, ASTROM, N, B, RI, DI )
   CALL iau_ATCIQZ ( RC, DC, ASTROM, RI, DI )
   CALL iau_ATCO13 ( RC, DC, PR, PD, PX, RV, UTC1, UTC2, DUT1, ELONG,
  :                  PHI, HM, XP, YP, PHPA, TC, RH, WL,
  :                  AOB, ZOB, HOB, DOB, ROB, EO, J )
   CALL iau_ATIC13 ( RI, DI, DATE1, DATE2, RC, DC, EO )
   CALL iau_ATICQ  ( RI, DI, ASTROM, RC, DC )
   CALL iau_ATCIQN ( RI, DI, ASTROM, N, B, RC, DC )
   CALL iau_ATIO13 ( RI, DI, UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
                     PHPA, TC, RH, WL, AOB, ZOB, HOB, DOB, ROB, J )
   CALL iau_ATIOQ  ( RI, DI, ASTROM, AOB, ZOB, HOB, DOB, ROB )
   CALL iau_ATOC13 ( TYPE, OB1, OB2, UTC1, UTC2, DUT1,
  :                  ELONG, PHI, HM, XP, YP, PHPA, TC, RH, WL,
  :                  RC, DC, J )
   CALL iau_ATOI13 ( TYPE, OB1, OB2, UTC1, UTC2, DUT1,
  :                  ELONG, PHI, HM, XP, YP, PHPA, TC, RH, WL,
  :                  RI, DI, J  )
   CALL iau_ATOIQ  ( TYPE, OB1, OB2, ASTROM, RI, DI )
   CALL iau_BI00   ( DPSIBI, DEPSBI, DRA )
   CALL iau_BP00   ( DATE1, DATE2, RB, RP, RBP )
   CALL iau_BP06   ( DATE1, DATE2, RB, RP, RBP )
   CALL iau_BPN2XY ( RBPN, X, Y )
   CALL iau_C2I00A ( DATE1, DATE2, RC2I )
   CALL iau_C2I00B ( DATE1, DATE2, RC2I )
   CALL iau_C2I06A ( DATE1, DATE2, RC2I )
   CALL iau_C2IBPN ( DATE1, DATE2, RBPN, RC2I )
   CALL iau_C2IXY  ( DATE1, DATE2, X, Y, RC2I )
   CALL iau_C2IXYS ( X, Y, S, RC2I )
   CALL iau_C2T00A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
   CALL iau_C2T00B ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
   CALL iau_C2T06A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
   CALL iau_C2TCEO ( RC2I, ERA, RPOM, RC2T )
   CALL iau_C2TCIO ( RC2I, ERA, RPOM, RC2T )
   CALL iau_C2TEQX ( RBPN, GST, RPOM, RC2T )
   CALL iau_C2TPE  ( TTA, TTB, UTA, UTB, DPSI, DEPS, XP, YP, RC2T )
   CALL iau_C2TXY  ( TTA, TTB, UTA, UTB, X, Y, XP, YP, RC2T )
   CALL iau_CAL2JD ( IY, IM, ID, DJM0, DJM, J )
   CALL iau_D2DTF  ( SCALE, NDP, D1, D2, IY, IM, ID, IHMSF, J )
   CALL iau_DAT    ( IY, IM, ID, FD, DELTAT, J )
   D =  iau_DTDB   ( DATE1, DATE2, UT, ELONG, U, V )
   CALL iau_DTF2D  ( SCALE, IY, IM, ID, IHR, IMN, SEC, D1, D2, J )
   CALL iau_ECEQ06 ( DATE1, DATE2, DL, DB, DR, DD )
```

```
 CALL iau_ECM06  ( DATE1, DATE2, RM );
 D =  iau_EE00   ( DATE1, DATE2, EPSA, DPSI )
 D =  iau_EE00A  ( DATE1, DATE2 )
 D =  iau_EE00B  ( DATE1, DATE2 )
 D =  iau_EE06A  ( DATE1, DATE2 )
 D =  iau_EECT00 ( DATE1, DATE2 )
 CALL iau_EFORM  ( N, A, F, J )
 D =  iau_EO06A  ( DATE1, DATE2 )
 D =  iau_EORS   ( RNPB, S )
 D =  iau_EPB    ( DJ1, DJ2 )
 CALL iau_EPB2JD ( EPB, DJM0, DJM )
 D =  iau_EPJ    ( DJ1, DJ2 )
 CALL iau_EPJ2JD ( EPJ, DJM0, DJM )
 CALL iau_EPV00  ( DJ1, DJ2, PVH, PVB, J )
 CALL iau_EQEC06 ( DATE1, DATE2, DR, DD, DL, DB )
 D =  iau_EQEQ94 ( DATE1, DATE2 )
 D =  iau_ERA00  ( DJ1, DJ2 )
 D =  iau_FAD03  ( T )
 D =  iau_FAE03  ( T )
 D =  iau_FAF03  ( T )
 D =  iau_FAJU03 ( T )
 D =  iau_FAL03  ( T )
 D =  iau_FALP03 ( T )
 D =  iau_FAMA03 ( T )
 D =  iau_FAME03 ( T )
 D =  iau_FANE03 ( T )
 D =  iau_FAOM03 ( T )
 D =  iau_FAPA03 ( T )
 D =  iau_FASA03 ( T )
 D =  iau_FAUR03 ( T )
 D =  iau_FAVE03 ( T )
 CALL iau_FK52H  ( R5, D5, DR5, DD5, PX5, RV5,
:                  RH, DH, DRH, DDH, PXH, RVH )
 CALL iau_FK5HIP ( R5H, S5H )
 CALL iau_FK5HZ  ( R5, D5, DATE1, DATE2, RH, DH )
 CALL iau_FW2M   ( GAMB, PHIB, PSI, EPS, R )
 CALL iau_FW2XY  ( GAMB, PHIB, PSI, EPS, X, Y )
 CALL iau_G2ICRS ( DL, DB, DR, DD )
 CALL iau_GC2GD  ( N, XYZ, ELONG, PHI, HEIGHT, J )
 CALL iau_GC2GDE ( A, F, XYZ, ELONG, PHI, HEIGHT, J )
 CALL iau_GD2GC  ( N, ELONG, PHI, HEIGHT, XYZ, J )
 CALL iau_GD2GCE ( A, F, ELONG, PHI, HEIGHT, XYZ, J )
 D =  iau_GMST00 ( UTA, UTB, TTA, TTB )
 D =  iau_GMST06 ( UTA, UTB, TTA, TTB )
 D =  iau_GMST82 ( UTA, UTB )
 D =  iau_GST00A ( UTA, UTB, TTA, TTB )
 D =  iau_GST00B ( UTA, UTB )
 D =  iau_GST06  ( UTA, UTB, TTA, TTB, RNPB )
 D =  iau_GST06A ( UTA, UTB, TTA, TTB )
 D =  iau_GST94  ( UTA, UTB )
 CALL iau_H2FK5  ( RH, DH, DRH, DDH, PXH, RVH,
:                  R5, D5, DR5, DD5, PX5, RV5 )
 CALL iau_HD2AE  ( HA, DEC, PHI, AZ, EL )
 D =  iau_HD2PA  ( HA, DEC, PHI )
 CALL iau_HFK5Z  ( RH, DH, DATE1, DATE2, R5, D5, DR5, DD5 )
 CALL iau_ICRS2G ( DR, DD, DL, DB  )
 CALL iau_JD2CAL ( DJ1, DJ2, IY, IM, ID, FD, J )
 CALL iau_JDCALF ( NDP, DJ1, DJ2, IYMDF, J )
 CALL iau_LD     ( BM, P, Q, E, EM, DLIM, P1 )
 CALL iau_LDN    ( N, B, OB, SC, SN )
 CALL iau_LDSUN  ( P, E, EM, P1 )
 CALL iau_LTECEQ ( EPJ, DL, DB, DR, DD )
 CALL iau_LTECM  ( EPJ, RM] )
 CALL iau_LTEQEC ( EPJ, DR, DD, DL, DB )
 CALL iau_LTP    ( EPJ, RP )
 CALL iau_LTPB   ( EPJ, RPB )
 CALL iau_LTPECL ( EPJ, VEC )
 CALL iau_LTPEQU ( EPJ, VEQ )
 CALL iau_NUM00A ( DATE1, DATE2, RMATN )
 CALL iau_NUM00B ( DATE1, DATE2, RMATN )
 CALL iau_NUM06A ( DATE1, DATE2, RMATN )
 CALL iau_NUMAT  ( EPSA, DPSI, DEPS, RMATN )
 CALL iau_NUT00A ( DATE1, DATE2, DPSI, DEPS )
```

```
 CALL iau_NUT00B ( DATE1, DATE2, DPSI, DEPS )
 CALL iau_NUT06A ( DATE1, DATE2, DPSI, DEPS )
 CALL iau_NUT80  ( DATE1, DATE2, DPSI, DEPS )
 CALL iau_NUTM80 ( DATE1, DATE2, RMATN )
 D =  iau_OBL06  ( DATE1, DATE2 )
 D =  iau_OBL80  ( DATE1, DATE2 )
 CALL iau_PB06   ( DATE1, DATE2, BZETA, BZ, BTHETA )
 CALL iau_PFW06  ( DATE1, DATE2, GAMB, PHIB, PSIB, EPSA )
 CALL iau_PLAN94 ( DATE1, DATE2, NP, PV, J )
 CALL iau_PMAT00 ( DATE1, DATE2, RBP )
 CALL iau_PMAT06 ( DATE1, DATE2, RBP )
 CALL iau_PMAT76 ( DATE1, DATE2, RMATP )
 CALL iau_PMPX   ( RC, DC, PR, PD, PX, RV, PMT, POB, PCO )
 CALL iau_PMSAFE ( RA1, DEC1, PMR1, PMD1, PX1, RV1,
:                  EP1A, EP1B, EP2A, EP2B,
:                  RA2, DEC2, PMR2, PMD2, PX2, RV2, J )
 CALL iau_PN00   ( DATE1, DATE2, DPSI, DEPS,
:                  EPSA, RB, RP, RBP, RN, RBPN )
 CALL iau_PN00A  ( DATE1, DATE2,
:                  DPSI, DEPS, EPSA, RB, RP, RBP, RN, RBPN )
 CALL iau_PN00B  ( DATE1, DATE2,
:                  DPSI, DEPS, EPSA, RB, RP, RBP, RN, RBPN )
 CALL iau_PN06   ( DATE1, DATE2, DPSI, DEPS,
:                  EPSA, RB, RP, RBP, RN, RBPN )
 CALL iau_PN06A  ( DATE1, DATE2,
:                  DPSI, DEPS, RB, RP, RBP, RN, RBPN )
 CALL iau_PNM00A ( DATE1, DATE2, RBPN )
 CALL iau_PNM00B ( DATE1, DATE2, RBPN )
 CALL iau_PNM06A ( DATE1, DATE2, RNPB )
 CALL iau_PNM80  ( DATE1, DATE2, RMATPN )
 CALL iau_P06E   ( DATE1, DATE2,
:                  EPS0, PSIA, OMA, BPA, BQA, PIA, BPIA,
:                  EPSA, CHIA, ZA, ZETAA, THETAA, PA, GAM, PHI, PSI )
 CALL iau_POM00  ( XP, YP, SP, RPOM )
 CALL iau_PR00   ( DATE1, DATE2, DPSIPR, DEPSPR )
 CALL iau_PREC76 ( DATE01, DATE02, DATE11, DATE12, ZETA, Z, THETA )
 CALL iau_PVSTAR ( PV, RA, DEC, PMR, PMD, PX, RV, J )
 CALL iau_PVTOB  ( ELONG, PHI, HM, XP, YP, SP, THETA, PV )
 CALL iau_REFCO  ( PHPA, TC, RH, WL, REFA, REFB )
 D =  iau_S00    ( DATE1, DATE2, X, Y )
 D =  iau_S00A   ( DATE1, DATE2 )
 D =  iau_S00B   ( DATE1, DATE2 )
 D =  iau_S06    ( DATE1, DATE2, X, Y )
 D =  iau_S06A   ( DATE1, DATE2 )
 D =  iau_SP00   ( DATE1, DATE2 )
 CALL iau_STARPM ( RA1, DEC1, PMR1, PMD1, PX1, RV1,
:                  EP1A, EP1B, EP2A, EP2B,
:                  RA2, DEC2, PMR2, PMD2, PX2, RV2, J )
 CALL iau_STARPV ( RA, DEC, PMR, PMD, PX, RV, PV, J )
 CALL iau_TAITT  ( TAI1, TAI2, TT1, TT2, J )
 CALL iau_TAIUT1 ( TAI1, TAI2, DTA, UT11, UT12, J )
 CALL iau_TAIUTC ( TAI1, TAI2, UTC1, UTC2, J )
 CALL iau_TCBTDB ( TCB1, TCB2, TDB1, TDB2, J )
 CALL iau_TCGTT  ( TCG1, TCG2, TT1, TT2, J )
 CALL iau_TDBTCB ( TDB1, TDB2, TCB1, TCB2, J )
 CALL iau_TDBTT  ( TDB1, TDB2, DTR, TT1, TT2, J )
 CALL iau_TPORS  ( XI, ETA, A, B, A01, B01, A02, B02, N )
 CALL iau_TPORV  ( XI, ETA, V, V01, V02, N )
 CALL iau_TPSTS  ( XI, ETA, A0, B0, A, B )
 CALL iau_TPSTV  ( XI, ETA, V0, V )
 CALL iau_TPXES  ( A, B, A0, B0, XI, ETA, J )
 CALL iau_TPXEV  ( V, V0, XI, ETA, J )
 CALL iau_TTTAI  ( TT1, TT2, TAI1, TAI2, J )
 CALL iau_TTTCG  ( TT1, TT2, TCG1, TCG2, J )
 CALL iau_TTTDB  ( TT1, TT2, DTR, TDB1, TDB2, J )
 CALL iau_TTUT1  ( TT1, TT2, DT, UT11, UT12, J )
 CALL iau_UT1TAI ( UT11, UT12, TAI1, TAI2, J )
 CALL iau_UT1TT  ( UT11, UT12, DT, TT1, TT2, J )
 CALL iau_UT1UTC ( UT11, UT12, DUT, UTC1, UTC2, J )
 CALL iau_UTCTAI ( UTC1, UTC2, DTA, TAI1, TAI2, J )
 CALL iau_UTCUT1 ( UTC1, UTC2, DUT, UT11, UT12, J )
 CALL iau_XY06   ( DATE1, DATE2, X, Y )
 CALL iau_XYS00A ( DATE1, DATE2, X, Y, S )
```

```
   CALL iau_XYS00B ( DATE1, DATE2, X, Y, S )
   CALL iau_XYS06A ( DATE1, DATE2, X, Y, S )
```

CALLS: C VERSION

```
       iauAb     ( pnat, v, s, bm1, ppr );
       iauAe2hd  ( az, el, phi, &ha, &dec );
       iauApcg   ( date1, date2, eb, eh, &astrom );
       iauApcg13 ( date1, date2, &astrom );
       iauApci   ( date1, date2, eb, eh, x, y, s, &astrom );
       iauApci13 ( date1, date2, &astrom, &eo );
       iauApco   ( date1, date2, eb, eh, x, y, s,
                   theta, elong, phi, hm, xp, yp, sp,
                   refa, refb, &astrom );
   i = iauApco13 ( utc1, utc2, dut1, elong, phi, hm, xp, yp,
                   phpa, tc, rh, wl, &astrom, &eo );
       iauApcs   ( date1, date2, pv, eb, eh, &astrom );
       iauApcs13 ( date1, date2, pv, &astrom );
       iauAper   ( theta, &astrom );
       iauAper13 ( ut11, ut12, &astrom );
       iauApio   ( sp, theta, elong, phi, hm, xp, yp, refa, refb,
                   &astrom );
   i = iauApio13 ( utc1, utc2, dut1, elong, phi, hm, xp, yp,
                   phpa, tc, rh, wl, &astrom );
       iauAtci13 ( rc, dc, pr, pd, px, rv, date1, date2,
                   &ri, &di, &eo );
       iauAtciq  ( rc, dc, pr, pd, px, rv, &astrom, &ri, &di );
       iauAtciqn ( rc, dc, pr, pd, px, rv, astrom, n, b, &ri, &di );
       iauAtciqz ( rc, dc, &astrom, &ri, &di );
   i = iauAtco13 ( rc, dc, pr, pd, px, rv, utc1, utc2, dut1,
                   elong phi, hm, xp, yp, phpa, tc, rh, wl,
                   aob, zob, hob, dob, rob, eo );
       iauAtic13 ( ri, di, date1, date2, &rc, &dc, &eo );
       iauAticq  ( ri, di, &astrom, &rc, &dc );
       iauAtciqn ( ri, di, astrom, n, b, &rc, &dc );
   i = iauAtio13 ( ri, di, utc1, utc2, dut1, elong, phi, hm, xp, yp,
                   phpa, tc, rh, wl, aob, zob, hob, dob, rob );
       iauAtioq  ( ri, di, &astrom, &aob, &zob, &hob, &dob, &rob );
   i = iauAtoc13 ( type, ob1, ob2, utc1, utc2, dut1,
                   elong, phi, hm, xp, yp, phpa, tc, rh, wl,
                   &rc, &dc );
   i = iauAtoi13 ( type, ob1, ob2, utc1, utc2, dut1, elong, phi, hm,
                   xp, yp, phpa, tc, rh, wl, &ri, &di  );
       iauAtoiq  ( type, ob1, ob2, &astrom, &ri, &di );
       iauBi00   ( &dpsibi, &depsbi, &dra );
       iauBp00   ( date1, date2, rb, rp, rbp );
       iauBp06   ( date1, date2, rb, rp, rbp );
       iauBpn2xy ( rbpn, &x, &y );
       iauC2i00a ( date1, date2, rc2i );
       iauC2i00b ( date1, date2, rc2i );
       iauC2i06a ( date1, date2, rc2i );
       iauC2ibpn ( date1, date2, rbpn, rc2i );
       iauC2ixy  ( date1, date2, x, y, rc2i );
       iauC2ixys ( x, y, s, rc2i );
       iauC2t00a ( tta, ttb, uta, utb, xp, yp, rc2t );
       iauC2t00b ( tta, ttb, uta, utb, xp, yp, rc2t );
       iauC2t06a ( tta, ttb, uta, utb, xp, yp, rc2t );
       iauC2tcio ( rc2i, era, rpom, rc2t );
       iauC2teqx ( rbpn, gst, rpom, rc2t );
       iauC2tpe  ( tta, ttb, uta, utb, dpsi, deps, xp, yp, rc2t );
       iauC2txy  ( tta, ttb, uta, utb, x, y, xp, yp, rc2t );
   i = iauCal2jd ( iy, im, id, &djm0, &djm );
   i = iauD2dtf  ( scale, ndp, d1, d2, &iy, &im, &id, ihmsf );
   i = iauDat    ( iy, im, id, fd, &deltat );
   d = iauDtdb   ( date1, date2, ut, elong, u, v );
   i = iauDtf2d  ( scale, iy, im, id, ihr, imn, sec, &d1, &d2 );
       iauEceq06 ( date1, date2, dl, db, &dr, &dd );
       iauEcm06  ( date1, date2, rm );
   d = iauEe00   ( date1, date2, epsa, dpsi );
   d = iauEe00a  ( date1, date2 );
   d = iauEe00b  ( date1, date2 );
   d = iauEe06   ( date1, date2 );
```

```
d = iauEect00 ( date1, date2 );
i = iauEform  ( n, &a, &f );
d = iauEo06   ( date1, date2 );
d = iauEors   ( rnpb, s );
d = iauEpb    ( dj1, dj2 );
    iauEpb2jd ( epb, &djm0, &djm );
d = iauEpj    ( dj1, dj2 );
    iauEpj2jd ( epj, &djm0, &djm );
i = iauEpv00  ( dj1, dj2, pvh, pvb );
    iauEqec06 ( date1, date2, dr, dd, &dl, &db );
d = iauEqeq94 ( date1, date2 );
d = iauEra00  ( dj1, dj2 );
d = iauFad03  ( t );
d = iauFae03  ( t );
d = iauFaf03  ( t );
d = iauFaju03 ( t );
d = iauFal03  ( t );
d = iauFalp03 ( t );
d = iauFama03 ( t );
d = iauFame03 ( t );
d = iauFane03 ( t );
d = iauFaom03 ( t );
d = iauFapa03 ( t );
d = iauFasa03 ( t );
d = iauFaur03 ( t );
d = iauFave03 ( t );
    iauFk52h  ( r5, d5, dr5, dd5, px5, rv5,
                &rh, &dh, &drh, &ddh, &pxh, &rvh );
    iauFk5hip ( r5h, s5h );
    iauFk5hz  ( r5, d5, date1, date2, &rh, &dh );
    iauFw2m   ( gamb, phib, psi, eps, r );
    iauFw2xy  ( gamb, phib, psi, eps, &x, &y );
    iauG2icrs ( dl, db, &dr, &dd );
i = iauGc2gd  ( n, xyz, &elong, &phi, &height );
i = iauGc2gde ( a, f, xyz, &elong, &phi, &height );
i = iauGd2gc  ( n, elong, phi, height, xyz );
i = iauGd2gce ( a, f, elong, phi, height, xyz );
d = iauGmst00 ( uta, utb, tta, ttb );
d = iauGmst06 ( uta, utb, tta, ttb );
d = iauGmst82 ( uta, utb );
d = iauGst00a ( uta, utb, tta, ttb );
d = iauGst00b ( uta, utb );
d = iauGst06  ( uta, utb, tta, ttb, rnpb );
d = iauGst06a ( uta, utb, tta, ttb );
d = iauGst94  ( uta, utb );
    iauH2fk5  ( rh, dh, drh, ddh, pxh, rvh,
                &r5, &d5, &dr5, &dd5, &px5, &rv5 );
    iauHd2ae  ( ha, dec, phi, &az, &el );
d = iauHd2pa  ( ha, dec, phi );
    iauHfk5z  ( rh, dh, date1, date2,
                &r5, &d5, &dr5, &dd5 );
    iauIcrs2g ( dr, dd, &dl, &db );
i = iauJd2cal ( dj1, dj2, &iy, &im, &id, &fd );
i = iauJdcalf ( ndp, dj1, dj2, iymdf );
    iauLd     ( bm, p, q, e, em, dlim, p1 );
    iauLdn    ( n, b, ob, sc, sn );
    iauLdsun  ( p, e, em, p1 );
    iauLteceq ( epj,  dl, db, &dr, &dd );
    iauLtecm  ( epj,  rm );
    iauLteqec ( epj,  dr,  dd, &dl, &db );
    iauLtp    ( epj,  rp );
    iauLtpb   ( epj,  rpb );
    iauLtpecl ( epj,  vec );
    iauLtpequ ( epj,  veq );
    iauNum00a ( date1, date2, rmatn );
    iauNum00b ( date1, date2, rmatn );
    iauNum06a ( date1, date2, rmatn );
    iauNumat  ( epsa, dpsi, deps, rmatn );
    iauNut00a ( date1, date2, &dpsi, &deps );
    iauNut00b ( date1, date2, &dpsi, &deps );
    iauNut06a ( date1, date2, &dpsi, &deps );
    iauNut80  ( date1, date2, &dpsi, &deps );
    iauNutm80 ( date1, date2, rmatn );
```

```
d = iauObl06  ( date1, date2 );
d = iauObl80  ( date1, date2 );
    iauPb06   ( date1, date2, &bzeta, &bz, &btheta );
    iauPfw06  ( date1, date2, &gamb, &phib, &psib, &epsa );
i = iauPlan94 ( date1, date2, np, pv );
    iauPmat00 ( date1, date2, rbp );
    iauPmat06 ( date1, date2, rbp );
    iauPmat76 ( date1, date2, rmatp );
    iauPmpx   ( rc, dc, pr, pd, px, rv, pmt, pob, pco );
i = iauPmsafe ( ra1, dec1, pmr1, pmd1, px1, rv1,
                ep1a, ep1b, ep2a, ep2b,
                &ra2, &dec2, &pmr2, &pmd2, &px2, &rv2);
    iauPn00   ( date1, date2, dpsi, deps,
                &epsa, rb, rp, rbp, rn, rbpn );
    iauPn00a  ( date1, date2,
                &dpsi, &deps, &epsa, rb, rp, rbp, rn, rbpn );
    iauPn00b  ( date1, date2,
                &dpsi, &deps, &epsa, rb, rp, rbp, rn, rbpn );
    iauPn06   ( date1, date2, dpsi, deps,
                &epsa, rb, rp, rbp, rn, rbpn );
    iauPn06a  ( date1, date2,
                &dpsi, &deps, &epsa, rb, rp, rbp, rn, rbpn );
    iauPnm00a ( date1, date2, rbpn );
    iauPnm00b ( date1, date2, rbpn );
    iauPnm06a ( date1, date2, rnpb );
    iauPnm80  ( date1, date2, rmatpn );
    iauP06e   ( date1, date2,
                &eps0, &psia, &oma, &bpa, &bqa, &pia, &bpia,
                &epsa, &chia, &za, &zetaa, &thetaa, &pa,
                &gam, &phi, &psi );
    iauPom00  ( xp, yp, sp, rpom );
    iauPr00   ( date1, date2, &dpsipr, &depspr );
    iauPrec76 ( date01, date02, date11, date12, &zeta, &z, &theta );
i = iauPvstar ( pv, &ra, &dec, &pmr, &pmd, &px, &rv );
    iauPvtob  ( elong, phi, hm, xp, yp, sp, theta, pv );
    iauRefco  ( phpa, tc, rh, wl, refa, refb );
d = iauS00    ( date1, date2, x, y );
d = iauS00a   ( date1, date2 );
d = iauS00b   ( date1, date2 );
d = iauS06    ( date1, date2, x, y );
d = iauS06a   ( date1, date2 );
d = iauSp00   ( date1, date2 );
i = iauStarpm ( ra1, dec1, pmr1, pmd1, px1, rv1,
                ep1a, ep1b, ep2a, ep2b,
                &ra2, &dec2, &pmr2, &pmd2, &px2, &rv2 );
i = iauStarpv ( ra, dec, pmr, pmd, px, rv, pv );
i = iauTaitt  ( tai1, tai2, &tt1, &tt2 );
i = iauTaiut1 ( tai1, tai2, dta, &ut11, &ut12 );
i = iauTaiutc ( tai1, tai2, &utc1, &utc2 );
i = iauTcbtdb ( tcb1, tcb2, &tdb1, &tdb2 );
i = iauTcgtt  ( tcg1, tcg2, &tt1, &tt2 );
i = iauTdbtcb ( tdb1, tdb2, &tcb1, &tcb2 );
i = iauTdbtt  ( tdb1, tdb2, dtr, &tt1, &tt2 );
i = iauTpors  ( xi, eta, a, b, &a01, &b01, &a02, &b02 );
i = iauTporv  ( xi, eta, v, v01, v02 );
    iauTpsts  ( xi, eta, a0, b0, &a, &b );
    iauTpstv  ( xi, eta, v0, v );
i = iauTpxes  ( a, b, a0, b0, &xi, &eta );
i = iauTpxev  ( v, v0, &xi, &eta );
i = iauTttai  ( tt1, tt2, &tai1, &tai2 );
i = iauTttcg  ( tt1, tt2, &tcg1, &tcg2 );
i = iauTttdb  ( tt1, tt2, dtr, &tdb1, &tdb2 );
i = iauTtut1  ( tt1, tt2, dt, &ut11, &ut12 );
i = iauUt1tai ( ut11, ut12, &tai1, &tai2 );
i = iauUt1tt  ( ut11, ut12, dt, &tt1, &tt2 );
i = iauUt1utc ( ut11, ut12, dut, &utc1, &utc2 );
i = iauUtctai ( utc1, utc2, dta, &tai1, &tai2 );
i = iauUtcut1 ( utc1, utc2, dut, &ut11, &ut12 );
    iauXy06   ( date1, date2, &x, &y );
    iauXys00a ( date1, date2, &x, &y, &s );
    iauXys00b ( date1, date2, &x, &y, &s );
    iauXys06a ( date1, date2, &x, &y, &s );
```

SOFA Vector/Matrix Library

PREFACE

The routines described here comprise the SOFA vector/matrix library. Their general appearance and coding style conforms to conventions agreed by the SOFA Board, and their functions, names and algorithms have been ratified by the Board. Procedures for soliciting and agreeing additions to the library are still evolving.

PROGRAMMING LANGUAGES

The SOFA routines are available in two programming languages at present: Fortran 77 and ANSI C.

There is a one-to-one relationship between the two language versions. The naming convention is such that a SOFA routine referred to generically as "EXAMPL" exists as a Fortran subprogram iau_EXAMPL and a C function iauExampl. The calls for the two versions are very similar, with the same arguments in the same order. In a few cases, the C equivalent of a Fortran SUBROUTINE subprogram uses a return value rather than an argument.

GENERAL PRINCIPLES

The library consists mostly of routines which operate on ordinary Cartesian vectors (x,y,z) and 3x3 rotation matrices. However, there is also support for vectors which represent velocity as well as position and vectors which represent rotation instead of position. The vectors which represent both position and velocity may be considered still to have dimensions (3), but to comprise elements each of which is two numbers, representing the value itself and the time derivative. Thus:

* "Position" or "p" vectors (or just plain 3-vectors) have dimension (3) in Fortran and [3] in C.

* "Position/velocity" or "pv" vectors have dimensions (3,2) in Fortran and [2][3] in C.

* "Rotation" or "r" matrices have dimensions (3,3) in Fortran and [3][3] in C. When used for rotation, they are "orthogonal"; the inverse of such a matrix is equal to the transpose. Most of the routines in this library do not assume that r-matrices are necessarily orthogonal and in fact work on any 3x3 matrix.

* "Rotation" or "r" vectors have dimensions (3) in Fortran and [3] in C. Such vectors are a combination of the Euler axis and angle and are convertible to and from r-matrices. The direction is the axis of rotation and the magnitude is the angle of rotation, in radians. Because the amount of rotation can be scaled up and down simply by multiplying the vector by a scalar, r-vectors are useful for representing spins about an axis which is fixed.

* The above rules mean that in terms of memory address, the three velocity components of a pv-vector follow the three position components. Application code is permitted to exploit this and all other knowledge of the internal layouts: that x, y and z appear in that order and are in a right-handed Cartesian coordinate system etc. For example, the cp function (copy a p-vector) can be used to copy the velocity component of a pv-vector (indeed, this is how the CPV routine is coded).

* The routines provided do not completely fill the range of operations that link all the various vector and matrix options, but are confined to functions that are required by other parts of the SOFA software or which are likely to prove useful.

In addition to the vector/matrix routines, the library contains some routines related to spherical angles, including conversions to and from sexagesimal format.

Using the library requires knowledge of vector/matrix methods, spherical trigonometry, and methods of attitude representation. These topics are covered in many textbooks, including "Spacecraft Attitude Determination and Control", James R. Wertz (ed.), Astrophysics and Space Science Library, Vol. 73, D. Reidel Publishing Company, 1986.

OPERATIONS INVOLVING P-VECTORS AND R-MATRICES

Initialize

ZP	zero p-vector
ZR	initialize r-matrix to null
IR	initialize r-matrix to identity

Copy/extend/extract

CP	copy p-vector
CR	copy r-matrix

Build rotations

RX	rotate r-matrix about x
RY	rotate r-matrix about y
RZ	rotate r-matrix about z

Spherical/Cartesian conversions

S2C	spherical to unit vector
C2S	unit vector to spherical
S2P	spherical to p-vector
P2S	p-vector to spherical

Operations on vectors

PPP	p-vector plus p-vector
PMP	p-vector minus p-vector
PPSP	p-vector plus scaled p-vector
PDP	inner (=scalar=dot) product of two p-vectors
PXP	outer (=vector=cross) product of two p-vectors
PM	modulus of p-vector
PN	normalize p-vector returning modulus
SXP	multiply p-vector by scalar

Operations on matrices

RXR	r-matrix multiply
TR	transpose r-matrix

Matrix-vector products

RXP	product of r-matrix and p-vector
TRXP	product of transpose of r-matrix and p-vector

Separation and position-angle

SEPP	angular separation from p-vectors
SEPS	angular separation from spherical coordinates
PAP	position-angle from p-vectors
PAS	position-angle from spherical coordinates

Rotation vectors

RV2M	r-vector to r-matrix
RM2V	r-matrix to r-vector

OPERATIONS INVOLVING PV-VECTORS

Initialize

ZPV	zero pv−vector

Copy/extend/extract

CPV	copy pv−vector
P2PV	append zero velocity to p−vector
PV2P	discard velocity component of pv−vector

Spherical/Cartesian conversions

S2PV	spherical to pv−vector
PV2S	pv−vector to spherical

Operations on vectors

PVPPV	pv−vector plus pv−vector
PVMPV	pv−vector minus pv−vector
PVDPV	inner (=scalar=dot) product of two pv−vectors
PVXPV	outer (=vector=cross) product of two pv−vectors
PVM	modulus of pv−vector
SXPV	multiply pv−vector by scalar
S2XPV	multiply pv−vector by two scalars
PVU	update pv−vector
PVUP	update pv−vector discarding velocity

Matrix−vector products

RXPV	product of r−matrix and pv−vector
TRXPV	product of transpose of r−matrix and pv−vector

OPERATIONS ON ANGLES

ANP	normalize radians to range 0 to 2pi
ANPM	normalize radians to range −pi to +pi
A2TF	decompose radians into hours, minutes, seconds
A2AF	decompose radians into degrees, arcminutes, arcseconds
AF2A	degrees, arcminutes, arcseconds to radians
D2TF	decompose days into hours, minutes, seconds
TF2A	hours, minutes, seconds to radians
TF2D	hours, minutes, seconds to days

CALLS: FORTRAN VERSION

```
CALL iau_A2AF  ( NDP, ANGLE, SIGN, IDMSF )
CALL iau_A2TF  ( NDP, ANGLE, SIGN, IHMSF )
CALL iau_AF2A  ( S, IDEG, IAMIN, ASEC, RAD, J )
D =  iau_ANP   ( A )
D =  iau_ANPM  ( A )
CALL iau_C2S   ( P, THETA, PHI )
CALL iau_CP    ( P, C )
CALL iau_CPV   ( PV, C )
CALL iau_CR    ( R, C )
CALL iau_D2TF  ( NDP, DAYS, SIGN, IHMSF )
CALL iau_IR    ( R )
CALL iau_P2PV  ( P, PV )
CALL iau_P2S   ( P, THETA, PHI, R )
CALL iau_PAP   ( A, B, THETA )
CALL iau_PAS   ( AL, AP, BL, BP, THETA )
CALL iau_PDP   ( A, B, ADB )
CALL iau_PM    ( P, R )
CALL iau_PMP   ( A, B, AMB )
CALL iau_PN    ( P, R, U )
CALL iau_PPP   ( A, B, APB )
CALL iau_PPSP  ( A, S, B, APSB )
CALL iau_PV2P  ( PV, P )
CALL iau_PV2S  ( PV, THETA, PHI, R, TD, PD, RD )
CALL iau_PVDPV ( A, B, ADB )
CALL iau_PVM   ( PV, R, S )
CALL iau_PVMPV ( A, B, AMB )
```

```
   CALL iau_PVPPV ( A, B, APB )
   CALL iau_PVU   ( DT, PV, UPV )
   CALL iau_PVUP  ( DT, PV, P )
   CALL iau_PVXPV ( A, B, AXB )
   CALL iau_PXP   ( A, B, AXB )
   CALL iau_RM2V  ( R, P )
   CALL iau_RV2M  ( P, R )
   CALL iau_RX    ( PHI, R )
   CALL iau_RXP   ( R, P, RP )
   CALL iau_RXPV  ( R, PV, RPV )
   CALL iau_RXR   ( A, B, ATB )
   CALL iau_RY    ( THETA, R )
   CALL iau_RZ    ( PSI, R )
   CALL iau_S2C   ( THETA, PHI, C )
   CALL iau_S2P   ( THETA, PHI, R, P )
   CALL iau_S2PV  ( THETA, PHI, R, TD, PD, RD, PV )
   CALL iau_S2XPV ( S1, S2, PV )
   CALL iau_SEPP  ( A, B, S )
   CALL iau_SEPS  ( AL, AP, BL, BP, S )
   CALL iau_SXP   ( S, P, SP )
   CALL iau_SXPV  ( S, PV, SPV )
   CALL iau_TF2A  ( S, IHOUR, IMIN, SEC, RAD, J )
   CALL iau_TF2D  ( S, IHOUR, IMIN, SEC, DAYS, J )
   CALL iau_TR    ( R, RT )
   CALL iau_TRXP  ( R, P, TRP )
   CALL iau_TRXPV ( R, PV, TRPV )
   CALL iau_ZP    ( P )
   CALL iau_ZPV   ( PV )
   CALL iau_ZR    ( R )
```

CALLS: C VERSION

```
        iauA2af  ( ndp, angle, &sign, idmsf );
        iauA2tf  ( ndp, angle, &sign, ihmsf );
   i =  iauAf2a  ( s, ideg, iamin, asec, &rad );
   d =  iauAnp   ( a );
   d =  iauAnpm  ( a );
        iauC2s   ( p, &theta, &phi );
        iauCp    ( p, c );
        iauCpv   ( pv, c );
        iauCr    ( r, c );
        iauD2tf  ( ndp, days, &sign, ihmsf );
        iauIr    ( r );
        iauP2pv  ( p, pv );
        iauP2s   ( p, &theta, &phi, &r );
   d =  iauPap   ( a, b );
   d =  iauPas   ( al, ap, bl, bp );
   d =  iauPdp   ( a, b );
   d =  iauPm    ( p );
        iauPmp   ( a, b, amb );
        iauPn    ( p, &r, u );
        iauPpp   ( a, b, apb );
        iauPpsp  ( a, s, b, apsb );
        iauPv2p  ( pv, p );
        iauPv2s  ( pv, &theta, &phi, &r, &td, &pd, &rd );
        iauPvdpv ( a, b, adb );
        iauPvm   ( pv, &r, &s );
        iauPvmpv ( a, b, amb );
        iauPvppv ( a, b, apb );
        iauPvu   ( dt, pv, upv );
        iauPvup  ( dt, pv, p );
        iauPvxpv ( a, b, axb );
        iauPxp   ( a, b, axb );
        iauRm2v  ( r, p );
        iauRv2m  ( p, r );
        iauRx    ( phi, r );
        iauRxp   ( r, p, rp );
        iauRxpv  ( r, pv, rpv );
        iauRxr   ( a, b, atb );
        iauRy    ( theta, r );
        iauRz    ( psi, r );
        iauS2c   ( theta, phi, c );
```

```
    iauS2p   ( theta, phi, r, p );
    iauS2pv  ( theta, phi, r, td, pd, rd, pV );
    iauS2xpv ( s1, s2, pv );
d = iauSepp  ( a, b );
d = iauSeps  ( al, ap, bl, bp );
    iauSxp   ( s, p, sp );
    iauSxpv  ( s, pv, spv );
i = iauTf2a  ( s, ihour, imin, sec, &rad );
i = iauTf2d  ( s, ihour, imin, sec, &days );
    iauTr    ( r, rt );
    iauTrxp  ( r, p, trp );
    iauTrxpv ( r, pv, trpv );
    iauZp    ( p );
    iauZpv   ( pv );
    iauZr    ( r );
```

```
      SUBROUTINE iau_A2AF ( NDP, ANGLE, SIGN, IDMSF )
*+
*  - - - - - - - - -
*   i a u _ A 2 A F
*  - - - - - - - - -
*
*  Decompose radians into degrees, arcminutes, arcseconds, fraction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     NDP       i        resolution (Note 1)
*     ANGLE     d        angle in radians
*
*  Returned:
*     SIGN      c        '+' or '-'
*     IDMSF     i(4)     degrees, arcminutes, arcseconds, fraction
*
*  Called:
*     iau_D2TF     decompose days to hms
*
*  Notes:
*
*  1) NDP is interpreted as follows:
*
*     NDP         resolution
*      :      ...0000 00 00
*     -7         1000 00 00
*     -6          100 00 00
*     -5           10 00 00
*     -4            1 00 00
*     -3            0 10 00
*     -2            0 01 00
*     -1            0 00 10
*      0            0 00 01
*      1            0 00 00.1
*      2            0 00 00.01
*      3            0 00 00.001
*      :            0 00 00.000...
*
*  2) The largest positive useful value for NDP is determined by the
*     size of ANGLE, the format of DOUBLE PRECISION floating-point
*     numbers on the target platform, and the risk of overflowing
*     IDMSF(4).  On a typical platform, for ANGLE up to 2pi, the
*     available floating-point precision might correspond to NDP=12.
*     However, the practical limit is typically NDP=9, set by the
*     capacity of a 32-bit IDMSF(4).
*
*  3) The absolute value of ANGLE may exceed 2pi.  In cases where it
*     does not, it is up to the caller to test for and handle the
*     case where ANGLE is very nearly 2pi and rounds up to 360 degrees,
*     by testing for IDMSF(1)=360 and setting IDMSF(1-4) to zero.
*
*-
```

```
      SUBROUTINE iau_A2TF ( NDP, ANGLE, SIGN, IHMSF )
*+
*  - - - - - - - - -
*   i a u _ A 2 T F
*  - - - - - - - - -
*
*  Decompose radians into hours, minutes, seconds, fraction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     NDP       i        resolution (Note 1)
*     ANGLE     d        angle in radians
*
*  Returned:
*     SIGN      c        '+' or '-'
*     IHMSF     i(4)     hours, minutes, seconds, fraction
*
*  Called:
*     iau_D2TF     decompose days to hms
*
*  Notes:
*
*  1) NDP is interpreted as follows:
*
*     NDP         resolution
*      :      ...0000 00 00
*     -7         1000 00 00
*     -6          100 00 00
*     -5           10 00 00
*     -4            1 00 00
*     -3            0 10 00
*     -2            0 01 00
*     -1            0 00 10
*      0            0 00 01
*      1            0 00 00.1
*      2            0 00 00.01
*      3            0 00 00.001
*      :            0 00 00.000...
*
*  2) The largest useful value for NDP is determined by the size
*     of ANGLE, the format of DOUBLE PRECISION floating-point numbers
*     on the target platform, and the risk of overflowing IHMSF(4).
*     On a typical platform, for ANGLE up to 2pi, the available
*     floating-point precision might correspond to NDP=12.  However,
*     the practical limit is typically NDP=9, set by the capacity of
*     a 32-bit IHMSF(4).
*
*  3) The absolute value of ANGLE may exceed 2pi.  In cases where it
*     does not, it is up to the caller to test for and handle the
*     case where ANGLE is very nearly 2pi and rounds up to 24 hours,
*     by testing for IHMSF(1)=24 and setting IHMSF(1-4) to zero.
*
*-
```

```
      SUBROUTINE iau_AB ( PNAT, V, S, BM1, PPR )
*+
*  - - - - - - -
*   i a u _ A B
*  - - - - - - -
*
*  Apply aberration to transform natural direction into proper
*  direction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*    PNAT      d(3)   natural direction to the source (unit vector)
*    V         d(3)   observer barycentric velocity in units of c
*    S         d      distance between the Sun and the observer (au)
*    BM1       d      sqrt(1-|v|^2): reciprocal of Lorenz factor
*
*  Returned:
*    PPR       d(3)   proper direction to source (unit vector)
*
*  Notes:
*
*  1) The algorithm is based on Expr. (7.40) in the Explanatory
*     Supplement (Urban & Seidelmann 2013), but with the following
*     changes:
*
*     o  Rigorous rather than approximate normalization is applied.
*
*     o  The gravitational potential term from Expr. (7) in
*        Klioner (2003) is added, taking into account only the Sun's
*        contribution.  This has a maximum effect of about
*        0.4 microarcsecond.
*
*  2) In almost all cases, the maximum accuracy will be limited by the
*     supplied velocity.  For example, if the SOFA iau_EPV00 routine is
*     used, errors of up to 5 microarcseconds could occur.
*
*  References:
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013).
*
*     Klioner, Sergei A., "A practical relativistic model for micro-
*     arcsecond astrometry in space", Astr. J. 125, 1580-1597 (2003).
*
*  Called:
*     iau_PDP      scalar product of two p-vectors
*
*-
```

```
      SUBROUTINE iau_AE2HD (AZ, EL, PHI, HA, DEC)
*+
*  - - - - - - - - - -
*   i a u _ A E 2 H D
*  - - - - - - - - - -
*
*  Horizon to equatorial coordinates:  transform azimuth and altitude
*  to hour angle and declination.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     AZ       d     azimuth
*     EL       d     elevation
*     PHI      d     observatory latitude
*
*  Returned:
*     HA       d     hour angle
*     DEC      d     declination
*
*  Notes:
*
*  1)  All the arguments are angles in radians.
*
*  2)  The sign convention for azimuth is north zero, east +pi/2.
*
*  3)  HA is returned in the range +/-pi.  Declination is returned in
*      the range +/-pi/2.
*
*  4)  The latitude PHI is pi/2 minus the angle between the Earth’s
*      rotation axis and the adopted zenith.  In many applications it
*      will be sufficient to use the published geodetic latitude of the
*      site.  In very precise (sub-arcsecond) applications, PHI can be
*      corrected for polar motion.
*
*  5)  The azimuth AZ must be with respect to the rotational north pole,
*      as opposed to the ITRS pole, and an azimuth with respect to north
*      on a map of the Earth’s surface will need to be adjusted for
*      polar motion if sub-arcsecond accuracy is required.
*
*  6)  Should the user wish to work with respect to the astronomical
*      zenith rather than the geodetic zenith, PHI will need to be
*      adjusted for deflection of the vertical (often tens of
*      arcseconds), and the zero point of HA will also be affected.
*
*  7)  The transformation is the same as Ve = Ry(phi-pi/2)*Rz(pi)*Vh,
*      where Ve and Vh are lefthanded unit vectors in the (ha,dec) and
*      (az,el) systems respectively and Rz and Ry are rotations about
*      first the z-axis and then the y-axis.  (n.b. Rz(pi) simply
*      reverses the signs of the x and y components.)  For efficiency,
*      the algorithm is written out rather than calling other utility
*      functions.  For applications that require even greater
*      efficiency, additional savings are possible if constant terms
*      such as functions of latitude are computed once and for all.
*
*  8)  Again for efficiency, no range checking of arguments is carried
*      out.
*
*  Last revision:   2018 January 2
*
*  SOFA release 2018-01-30
*
*  Copyright (C) 2018 IAU SOFA Board.  See notes at end.
*
*-----------------------------------------------------------------------

      IMPLICIT NONE

      DOUBLE PRECISION AZ, EL, PHI, HA, DEC
```

```
      DOUBLE PRECISION SA, CA, SE, CE, SP, CP, X, Y, Z, R


*  Useful trig functions.
      SA = SIN(AZ)
      CA = COS(AZ)
      SE = SIN(EL)
      CE = COS(EL)
      SP = SIN(PHI)
      CP = COS(PHI)

*  Az,Alt unit vector.
      X = - CA*CE*SP + SE*CP
      Y = - SA*CE
      Z = CA*CE*CP + SE*SP

*  To spherical.
      R = SQRT(X*X + Y*Y)
      IF ( R.EQ.0D0 ) THEN
         HA = 0D0
      ELSE
         HA = ATAN2(Y,X)
      END IF
      DEC = ATAN2(Z,R)

*  Finished.

*+----------------------------------------------------------------------
*
*  Copyright (C) 2018
*  Standards Of Fundamental Astronomy Board
*  of the International Astronomical Union.
*
*  =====================
*  SOFA Software License
*  =====================
*
*  NOTICE TO USER:
*
*  BY USING THIS SOFTWARE YOU ACCEPT THE FOLLOWING SIX TERMS AND
*  CONDITIONS WHICH APPLY TO ITS USE.
*
*  1. The Software is owned by the IAU SOFA Board ("SOFA").
*
*  2. Permission is granted to anyone to use the SOFA software for any
*     purpose, including commercial applications, free of charge and
*     without payment of royalties, subject to the conditions and
*     restrictions listed below.
*
*  3. You (the user) may copy and distribute SOFA source code to others,
*     and use and adapt its code and algorithms in your own software,
*     on a world-wide, royalty-free basis.  That portion of your
*     distribution that does not consist of intact and unchanged copies
*     of SOFA source code files is a "derived work" that must comply
*     with the following requirements:
*
*     a) Your work shall be marked or carry a statement that it
*        (i) uses routines and computations derived by you from
*        software provided by SOFA under license to you; and
*        (ii) does not itself constitute software provided by and/or
*        endorsed by SOFA.
*
*     b) The source code of your derived work must contain descriptions
*        of how the derived work is based upon, contains and/or differs
*        from the original SOFA software.
*
*     c) The names of all routines in your derived work shall not
*        include the prefix "iau" or "sofa" or trivial modifications
*        thereof such as changes of case.
*
*     d) The origin of the SOFA components of your derived work must
*        not be misrepresented;  you must not claim that you wrote the
```

```
*        original software, nor file a patent application for SOFA
*        software or algorithms embedded in the SOFA software.
*
*     e) These requirements must be reproduced intact in any source
*        distribution and shall apply to anyone to whom you have
*        granted a further right to modify the source code of your
*        derived work.
*
*     Note that, as originally distributed, the SOFA software is
*     intended to be a definitive implementation of the IAU standards,
*     and consequently third-party modifications are discouraged.  All
*     variations, no matter how minor, must be explicitly marked as
*     such, as explained above.
*
*  4. You shall not cause the SOFA software to be brought into
*     disrepute, either by misuse, or use for inappropriate tasks, or
*     by inappropriate modification.
*
*  5. The SOFA software is provided "as is" and SOFA makes no warranty
*     as to its use or performance.   SOFA does not and cannot warrant
*     the performance or results which the user may obtain by using the
*     SOFA software.  SOFA makes no warranties, express or implied, as
*     to non-infringement of third party rights, merchantability, or
*     fitness for any particular purpose.  In no event will SOFA be
*     liable to the user for any consequential, incidental, or special
*     damages, including any lost profits or lost savings, even if a
*     SOFA representative has been advised of such damages, or for any
*     claim by any third party.
*
*  6. The provision of any version of the SOFA software under the terms
*     and conditions specified herein does not imply that future
*     versions will also be made available under the same terms and
*     conditions.
*
*  In any published work or commercial product which uses the SOFA
*  software directly, acknowledgement (see www.iausofa.org) is
*  appreciated.
*
*  Correspondence concerning SOFA software should be addressed as
*  follows:
*
*      By email:  sofa@ukho.gov.uk
*      By post:   IAU SOFA Center
*                 HM Nautical Almanac Office
*                 UK Hydrographic Office
*                 Admiralty Way, Taunton
*                 Somerset, TA1 2DN
*                 United Kingdom
*
*-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE iau_AF2A ( S, IDEG, IAMIN, ASEC, RAD, J )
*+
*  - - - - - - - - - -
*   i a u _ A F 2 A
*  - - - - - - - - - -
*
*  Convert degrees, arcminutes, arcseconds to radians.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     S          c      sign:  '-' = negative, otherwise positive
*     IDEG       i      degrees
*     IAMIN      i      arcminutes
*     ASEC       d      arcseconds
*
*  Returned:
*     RAD        d      angle in radians
*     J          i      status:  0 = OK
*                                1 = IDEG outside range 0-359
*                                2 = IAMIN outside range 0-59
*                                3 = ASEC outside range 0-59.999...
*
*  Notes:
*
*  1)  If the s argument is a string, only the leftmost character is
*      used and no warning status is provided.
*
*  2)  The result is computed even if any of the range checks fail.
*
*  3)  Negative IDEG, IAMIN and/or ASEC produce a warning status, but
*      the absolute value is used in the conversion.
*
*  4)  If there are multiple errors, the status value reflects only the
*      first, the smallest taking precedence.
*
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_ANP ( A )
*+
*  - - - - - - - -
*   i a u _ A N P
*  - - - - - - - -
*
*  Normalize angle into the range 0 <= A < 2pi.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A          d       angle (radians)
*
*  Returned:
*     iau_ANP    d       angle in range 0-2pi
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_ANPM ( A )
*+
*  - - - - - - - - -
*   i a u _ A N P M
*  - - - - - - - - -
*
*  Normalize angle into the range -pi <= A < +pi.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A          d       angle (radians)
*
*  Returned:
*     iau_ANPM   d       angle in range +/-pi
*
*-
```

```
      SUBROUTINE iau_APCG ( DATE1, DATE2, EBPV, EHP, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P C G
*  - - - - - - - - -
*
*  For a geocentric observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and GCRS coordinates.
*  The Earth ephemeris is supplied by the caller.
*
*  The parameters produced by this routine are required in the parallax,
*  light deflection and aberration parts of the astrometric
*  transformation chain.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d       TDB as a 2-part...
*     DATE2    d       ...Julian Date (Note 1)
*     EBPV     d(3,2)  Earth barycentric position/velocity (au, au/day)
*     EHP      d(3)    Earth heliocentric position (au)
*
*  Returned:
*     ASTROM   d(30)   star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) All the vectors are with respect to BCRS axes.
*
```

```
*  3) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  4) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_ZPV      zero pv-vector
*     iau_APCS     astrometry parameters, ICRS-GCRS, space observer
*
*-
```

```
      SUBROUTINE iau_APCG13 ( DATE1, DATE2, ASTROM )
*+
*  - - - - - - - - - - -
*   i a u _ A P C G 1 3
*  - - - - - - - - - - -
*
*  For a geocentric observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and GCRS coordinates.
*  The caller supplies the date, and SOFA models are used to predict
*  the Earth ephemeris.
*
*  The parameters produced by this routine are required in the
*  parallax, light deflection and aberration parts of the astrometric
*  transformation chain.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d      TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 1)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) All the vectors are with respect to BCRS axes.
*
*  3) In cases where the caller wishes to supply his own Earth
```

```
*     ephemeris, the routine iau_APCG can be used instead of the present
*     routine.
*
*  4) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  5) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_EPV00    Earth position and velocity
*     iau_APCG     astrometry parameters, ICRS-GCRS, geocenter
*
*-
```

```
      SUBROUTINE iau_APCI ( DATE1, DATE2, EBPV, EHP, X, Y, S, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P C I
*  - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and geocentric CIRS
*  coordinates.  The Earth ephemeris and CIP/CIO are supplied by the
*  caller.
*
*  The parameters produced by this routine are required in the parallax,
*  light deflection, aberration, and bias-precession-nutation parts of
*  the astrometric transformation chain.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d       TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 1)
*     EBPV     d(3,2)  Earth barycentric position/velocity (au, au/day)
*     EHP      d(3)    Earth heliocentric position (au)
*     X,Y      d       CIP X,Y (components of unit vector)
*     S        d       the CIO locator s (radians)
*
*  Returned:
*     ASTROM   d(30)   star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
```

```
*
*  2) All the vectors are with respect to BCRS axes.
*
*  3) In cases where the caller does not wish to provide the Earth
*     ephemeris and CIP/CIO, the routine iau_APCI13 can be used instead
*     of the present routine.  This computes the required quantities
*     using other SOFA routines.
*
*  4) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  5) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_APCG     astrometry parameters, ICRS-GCRS, geocenter
*     iau_C2IXYS   celestial-to-intermediate matrix, given X,Y and s
*
*-
```

```
      SUBROUTINE iau_APCI13 ( DATE1, DATE2, ASTROM, EO )
*+
*  - - - - - - - - - - -
*   i a u _ A P C I 1 3
*  - - - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and geocentric CIRS
*  coordinates.  The caller supplies the date, and SOFA models are
*  used to predict the Earth ephemeris and CIP/CIO.
*
*  The parameters produced by this routine are required in the parallax,
*  light deflection, aberration, and bias-precession-nutation parts of
*  the astrometric transformation chain.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d      TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 1)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*     EO       d      equation of the origins (ERA-GST)
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) All the vectors are with respect to BCRS axes.
*
```

```
*  3) In cases where the caller wishes to supply his own Earth
*     ephemeris and CIP/CIO, the routine iau_APCI can be used instead of
*     the present routine.
*
*  4) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  5) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_EPV00    Earth position and velocity
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*     iau_APCI     astrometry parameters, ICRS-CIRS
*     iau_EORS     equation of the origins, given NPB matrix and s
*
*-
```

```
      SUBROUTINE iau_APCO ( DATE1, DATE2, EBPV, EHP, X, Y, S,
     :                      THETA, ELONG, PHI, HM, XP, YP, SP,
     :                      REFA, REFB, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P C O
*  - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and observed coordinates.
*  The caller supplies the Earth ephemeris, the Earth rotation
*  information and the refraction constants as well as the site
*  coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d       TDB as a 2-part...
*     DATE2    d       ...Julian Date (Note 1)
*     EBPV     d(3,2)  Earth barycentric pos/vel (au, au/day, Note 2)
*     EHP      d(3)    Earth heliocentric position (au, Note 2)
*     X,Y      d       CIP X,Y (components of unit vector)
*     S        d       the CIO locator s (radians)
*     THETA    d       Earth rotation angle (radians)
*     ELONG    d       longitude (radians, east +ve, Note 3)
*     PHI      d       latitude (geodetic, radians, Note 3)
*     HM       d       height above ellipsoid (m, geodetic, Note 3)
*     XP,YP    d       polar motion coordinates (radians, Note 4)
*     SP       d       the TIO locator s' (radians, Note 4)
*     REFA     d       refraction constant A (radians, Note 5)
*     REFB     d       refraction constant B (radians, Note 5)
*
*  Returned:
*     ASTROM   d(30)   star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
```

```
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) The vectors EB, EH, and all the ASTROM vectors, are with respect
*     to BCRS axes.
*
*  3) The geographical coordinates are with respect to the WGS84
*     reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN
*     CONVENTION:  the longitude required by the present routine is
*     right-handed, i.e. east-positive, in accordance with geographical
*     convention.
*
*  4) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions), measured along the
*     meridians 0 and 90 deg west respectively.  SP is the TIO locator
*     s', in radians, which positions the Terrestrial Intermediate
*     Origin on the equator.  For many applications, XP, YP and
*     (especially) SP can be set to zero.
*
*     Internally, the polar motion is stored in a form rotated onto the
*     local meridian.
*
*  5) The refraction constants REFA and REFB are for use in a
*     dZ = A*tan(Z)+B*tan^3(Z) model, where Z is the observed
*     (i.e. refracted) zenith distance and dZ is the amount of
*     refraction.
*
*  6) It is advisable to take great care with units, as even unlikely
*     values of the input parameters are accepted and processed in
*     accordance with the models used.
*
*  7) In cases where the caller does not wish to provide the Earth
*     Ephemeris, the Earth rotation information and refraction
*     constants, the routine iau_APCO13 can be used instead of the
*     present routine.  This starts from UTC and weather readings etc.
*     and computes suitable values using other SOFA routines.
*
*  8) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  9) The context array ASTROM produced by this routine is used by
*     iau_ATIOQ, iau_ATOIQ, iau_ATCIQ*, and iau_ATICQ*.
*
```

```
*  Called:
*     iau_APER     astrometry parameters: update ERA
*     iau_C2IXYS   celestial-to-intermediate matrix, given X,Y and s
*     iau_PVTOB    position/velocity of terrestrial station
*     iau_TRXPV    product of transpose of r-matrix and pv-vector
*     iau_APCS     astrometry parameters, ICRS-GCRS, space observer
*     iau_CR       copy r-matrix
*
*-
```

```
      SUBROUTINE iau_APCO13 ( UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
     :                        PHPA, TC, RH, WL, ASTROM, EO, J )
*+
*  - - - - - - - - - - -
*   i a u _ A P C O 1 3
*  - - - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between ICRS and observed coordinates.
*  The caller supplies UTC, site coordinates, ambient air conditions and
*  observing wavelength, and SOFA models are used to obtain the Earth
*  ephemeris, CIP/CIO and refraction constants.
*
*  The parameters produced by this routine are required in the parallax,
*  light deflection, aberration, and bias-precession-nutation parts of
*  the ICRS/CIRS transformations.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTC1     d      UTC as a 2-part...
*     UTC2     d      ...quasi Julian Date (Notes 1,2)
*     DUT1     d      UT1-UTC (seconds, Note 3)
*     ELONG    d      longitude (radians, east +ve, Note 4)
*     PHI      d      latitude (geodetic, radians, Note 4)
*     HM       d      height above ellipsoid (m, geodetic, Notes 4,6)
*     XP,YP    d      polar motion coordinates (radians, Note 5)
*     PHPA     d      pressure at the observer (hPa = mB, Note 6)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0-1)
*     WL       d      wavelength (micrometers, Note 7)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*     EO       d      equation of the origins (ERA-GST)
*     J        i      status: +1 = dubious year (Note 2)
*                              0 = OK
*                             -1 = unacceptable date
*
*  Notes:
*
*  1)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2-part quasi Julian Date, as
*      it implements the leap-second-ambiguity convention just
```

```
*       described.
*
*  2)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the
*      future to be trusted.  See iau_DAT for further details.
*
*  3)  UT1-UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1-UTC within +/- 0.9s.  n.b. This
*      practice is under review, and in the future UT1-UTC may grow
*      essentially without limit.
*
*  4)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east-positive
*      (i.e. right-handed), in accordance with geographical convention.
*
*  5)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*      Internally, the polar motion is stored in a form rotated onto
*      the local meridian.
*
*  6)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB), is
*      available, an adequate estimate of hm can be obtained from the
*      expression
*
*            hm = -29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea-level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( -hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to
*      the pressure and that an accurate phpa value is important for
*      precise work.
*
*  7)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  8)  It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  9)  In cases where the caller wishes to supply his own Earth
*      ephemeris, Earth rotation information and refraction constants,
*      the routine iau_APCO can be used instead of the present routine.
*
*  10) This is one of several routines that inserts into the ASTROM
*      array star-independent parameters needed for the chain of
*      astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*      The various routines support different classes of observer and
*      portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
```

```
*
*      Those with names ending in "13" use contemporary SOFA models to
*      compute the various ephemerides.  The others accept ephemerides
*      supplied by the caller.
*
*      The transformation from ICRS to GCRS covers space motion,
*      parallax, light deflection, and aberration.  From GCRS to CIRS
*      comprises frame bias and precession-nutation.  From CIRS to
*      observed takes account of Earth rotation, polar motion, diurnal
*      aberration and parallax (unless subsumed into the ICRS <-> GCRS
*      transformation), and atmospheric refraction.
*
*  11) The context array ASTROM produced by this routine is used by
*      iau_ATIOQ, iau_ATOIQ, iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_UTCTAI   UTC to TAI
*     iau_TAITT    TAI to TT
*     iau_UTCUT1   UTC to UT1
*     iau_EPV00    Earth position and velocity
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_REFCO    refraction constants for given ambient conditions
*     iau_APCO     astrometry parameters, ICRS-observed
*     iau_EORS     equation of the origins, given NPB matrix and s
*
*-
```

```
      SUBROUTINE iau_APCS ( DATE1, DATE2, PV, EBPV, EHP, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P C S
*  - - - - - - - - -
*
*  For an observer whose geocentric position and velocity are known,
*  prepare star-independent astrometry parameters for transformations
*  between ICRS and GCRS.  The Earth ephemeris is supplied by the
*  caller.
*
*  The parameters produced by this routine are required in the space
*  motion, parallax, light deflection and aberration parts of the
*  astrometric transformation chain.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d       TDB as a 2-part...
*     DATE2    d       ...Julian Date (Note 1)
*     PV       d(3,2)  observer's geocentric pos/vel (m, m/s)
*     EBPV     d(3,2)  Earth barycentric position/velocity (au, au/day)
*     EHP      d(3)    Earth heliocentric position (au)
*
*  Returned:
*     ASTROM   d(30)   star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
```

```
*  2) All the vectors are with respect to BCRS axes.
*
*  3) Providing separate arguments for (i) the observer’s geocentric
*     position and velocity and (ii) the Earth ephemeris is done for
*     convenience in the geocentric, terrestrial and Earth orbit cases.
*     For deep space applications it maybe more convenient to specify
*     zero geocentric position and velocity and to supply the
*     observer’s position and velocity information directly instead of
*     with respect to the Earth.  However, note the different units:
*     m and m/s for the geocentric vectors, au and au/day for the
*     heliocentric and barycentric vectors.
*
*  4) In cases where the caller does not wish to provide the Earth
*     ephemeris, the routine iau_APCS13 can be used instead of the
*     present routine.  This computes the Earth ephemeris using the
*     SOFA routine iau_EPV00.
*
*  5) This is one of several routines that inserts into the ASTROM
*     array star−independent parameters needed for the chain of
*     astrometric transformations ICRS <−> GCRS <−> CIRS <−> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <−> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <−> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <−> observed
*        iau_APCS iau_APCS13    space           ICRS <−> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <−> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession−nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <−> GCRS
*     transformation), and atmospheric refraction.
*
*  6) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_CP       copy p−vector
*     iau_PM       modulus of p−vector
*     iau_PN       decompose p−vector into modulus and direction
*     iau_IR       initialize r−matrix to identity
*
*−
```

```
      SUBROUTINE iau_APCS13 ( DATE1, DATE2, PV, ASTROM )
*+
*  - - - - - - - - - - - -
*   i a u _ A P C S 1 3
*  - - - - - - - - - - - -
*
*  For an observer whose geocentric position and velocity are known,
*  prepare star-independent astrometry parameters for transformations
*  between ICRS and GCRS.  The Earth ephemeris is from SOFA models.
*
*  The parameters produced by this routine are required in the space
*  motion, parallax, light deflection and aberration parts of the
*  astrometric transformation chain.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d      TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 1)
*     PV       d(3,2) observer's geocentric pos/vel (Note 3)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     unchanged
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) All the vectors are with respect to BCRS axes.
*
*  3) The observer's position and velocity PV are geocentric but with
```

```
*     respect to BCRS axes, and in units of m and m/s.  No assumptions
*     are made about proximity to the Earth, and the routine can be
*     used for deep space applications as well as Earth orbit and
*     terrestrial.
*
*  4) In cases where the caller wishes to supply his own Earth
*     ephemeris, the routine iau_APCS can be used instead of the present
*     routine.
*
*  5) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  6) The context array ASTROM produced by this routine is used by
*     iau_ATCIQ* and iau_ATICQ*.
*
*  Called:
*     iau_EPV00    Earth position and velocity
*     iau_APCS     astrometry parameters, ICRS-GCRS, space observer
*
*-
```

```
      SUBROUTINE iau_APER ( THETA, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P E R
*  - - - - - - - - -
*
*  In the star-independent astrometry parameters, update only the
*  Earth rotation angle, supplied by the caller explicitly.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     THETA    d      Earth rotation angle (radians, Note 2)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      not used
*               (2-4)    not used
*               (5-7)    not used
*               (8)      not used
*               (9-11)   not used
*               (12)     not used
*               (13-21)  not used
*               (22)     longitude + s’ (radians)
*               (23)     not used
*               (24)     not used
*               (25)     not used
*               (26)     not used
*               (27)     not used
*               (28)     not used
*               (29)     not used
*               (30)     not used
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      unchanged
*               (2-4)    unchanged
*               (5-7)    unchanged
*               (8)      unchanged
*               (9-11)   unchanged
*               (12)     unchanged
*               (13-21)  unchanged
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     "local" Earth rotation angle (radians)
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) This routine exists to enable sidereal-tracking applications to
*     avoid wasteful recomputation of the bulk of the astrometry
*     parameters:  only the Earth rotation is updated.
*
*  2) For targets expressed as equinox based positions, such as
*     classical geocentric apparent (RA,Dec), the supplied THETA can be
*     Greenwich apparent sidereal time rather than Earth rotation
*     angle.
*
*  3) The routine iau_APER13 can be used instead of the present routine,
*     and starts from UT1 rather than ERA itself.
*
*  4) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
```

```
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*-
```

```
      SUBROUTINE iau_APER13 ( UT11, UT12, ASTROM )
*+
*  - - - - - - - - - - -
*   i a u _ A P E R 1 3
*  - - - - - - - - - - -
*
*  In the star-independent astrometry parameters, update only the
*  Earth rotation angle.  The caller provides UT1 (n.b. not UTC).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UT11     d      UT1 as a 2-part...
*     UT12     d      ...Julian Date (Note 1)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      not used
*               (2-4)    not used
*               (5-7)    not used
*               (8)      not used
*               (9-11)   not used
*               (12)     not used
*               (13-21)  not used
*               (22)     longitude + s' (radians)
*               (23)     not used
*               (24)     not used
*               (25)     not used
*               (26)     not used
*               (27)     not used
*               (28)     not used
*               (29)     not used
*               (30)     not used
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      unchanged
*               (2-4)    unchanged
*               (5-7)    unchanged
*               (8)      unchanged
*               (9-11)   unchanged
*               (12)     unchanged
*               (13-21)  unchanged
*               (22)     unchanged
*               (23)     unchanged
*               (24)     unchanged
*               (25)     unchanged
*               (26)     unchanged
*               (27)     unchanged
*               (28)     "local" Earth rotation angle (radians)
*               (29)     unchanged
*               (30)     unchanged
*
*  Notes:
*
*  1) The UT1 date (n.b. not UTC) UT11+UT12 is a Julian Date,
*     apportioned in any convenient way between the arguments UT11 and
*     UT12.  For example, JD(UT1)=2450123.7 could be expressed in any
*     of these ways, among others:
*
*            UT11           UT12
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  The date & time method is
```

```
*     best matched to the algorithm used:  maximum precision is
*     delivered when the UT11 argument is for 0hrs UT1 on the day in
*     question and the UT12 argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) If the caller wishes to provide the Earth rotation angle itself,
*     the routine iau_APER can be used instead.  One use of this
*     technique is to substitute Greenwich apparent sidereal time and
*     thereby to support equinox based transformations directly.
*
*  3) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  Called:
*     iau_APER     astrometry parameters: update ERA
*     iau_ERA00    Earth rotation angle, IAU 2000
*
*-
```

```
      SUBROUTINE iau_APIO ( SP, THETA, ELONG, PHI, HM, XP, YP,
     :                      REFA, REFB, ASTROM )
*+
*  - - - - - - - - -
*   i a u _ A P I O
*  - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between CIRS and observed coordinates.
*  The caller supplies the Earth orientation information and the
*  refraction constants as well as the site coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     SP       d      the TIO locator s' (radians, Note 1)
*     THETA    d      Earth rotation angle (radians)
*     ELONG    d      longitude (radians, east +ve, Note 2)
*     PHI      d      geodetic latitude (radians, Note 2)
*     HM       d      height above ellipsoid (m, geodetic Note 2)
*     XP,YP    d      polar motion coordinates (radians, Note 3)
*     REFA     d      refraction constant A (radians, Note 4)
*     REFB     d      refraction constant B (radians, Note 4)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      unchanged
*               (2-4)    unchanged
*               (5-7)    unchanged
*               (8)      unchanged
*               (9-11)   unchanged
*               (12)     unchanged
*               (13-21)  unchanged
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Notes:
*
*  1) SP, the TIO locator s', is a tiny quantity needed only by the most
*     precise applications.  It can either be set to zero or predicted
*     using the SOFA routine iau_SP00.
*
*  2) The geographical coordinates are with respect to the WGS84
*     reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*     longitude required by the present routine is east-positive
*     (i.e. right-handed), in accordance with geographical convention.
*
*  3) The polar motion XP,YP can be obtained from IERS bulletins.  The
*     values are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians 0 and 90 deg west respectively.  For many applications,
*     XP and YP can be set to zero.
*
*     Internally, the polar motion is stored in a form rotated onto the
*     local meridian.
*
*  4) The refraction constants REFA and REFB are for use in a
*     dZ = A*tan(Z)+B*tan^3(Z) model, where Z is the observed
*     (i.e. refracted) zenith distance and dZ is the amount of
*     refraction.
*
```

```
*  5) It is advisable to take great care with units, as even unlikely
*     values of the input parameters are accepted and processed in
*     accordance with the models used.
*
*  6) In cases where the caller does not wish to provide the Earth
*     rotation information and refraction constants, the routine
*     iau_APIO13 can be used instead of the present routine.  This
*     starts from UTC and weather readings etc. and computes suitable
*     values using other SOFA routines.
*
*  7) This is one of several routines that inserts into the ASTROM
*     array star-independent parameters needed for the chain of
*     astrometric transformations ICRS <-> GCRS <-> CIRS <-> observed.
*
*     The various routines support different classes of observer and
*     portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <-> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <-> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <-> observed
*        iau_APCS iau_APCS13    space           ICRS <-> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <-> observed
*
*     Those with names ending in "13" use contemporary SOFA models to
*     compute the various ephemerides.  The others accept ephemerides
*     supplied by the caller.
*
*     The transformation from ICRS to GCRS covers space motion,
*     parallax, light deflection, and aberration.  From GCRS to CIRS
*     comprises frame bias and precession-nutation.  From CIRS to
*     observed takes account of Earth rotation, polar motion, diurnal
*     aberration and parallax (unless subsumed into the ICRS <-> GCRS
*     transformation), and atmospheric refraction.
*
*  8) The context array ASTROM produced by this routine is used by
*     iau_ATIOQ and iau_ATOIQ.
*
*  Called:
*     iau_PVTOB    position/velocity of terrestrial station
*     iau_APER     astrometry parameters: update ERA
*
*-
```

```
      SUBROUTINE iau_APIO13 ( UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
     :                        PHPA, TC, RH, WL, ASTROM, J )
*+
*  - - - - - - - - - - -
*   i a u _ A P I O 1 3
*  - - - - - - - - - - -
*
*  For a terrestrial observer, prepare star-independent astrometry
*  parameters for transformations between CIRS and observed coordinates.
*  The caller supplies UTC, site coordinates, ambient air conditions and
*  observing wavelength.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTC1     d      UTC as a 2-part...
*     UTC2     d      ...quasi Julian Date (Notes 1,2)
*     DUT1     d      UT1-UTC (seconds)
*     ELONG    d      longitude (radians, east +ve, Note 3)
*     PHI      d      geodetic latitude (radians, Note 3)
*     HM       d      height above ellipsoid (m, geodetic Notes 4,6)
*     XP,YP    d      polar motion x-coordinates (radians, Note 5)
*     PHPA     d      pressure at the observer (hPa = mB, Note 6)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0-1)
*     WL       d      wavelength (micrometers, Note 7)
*
*  Returned:
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      unchanged
*               (2-4)    unchanged
*               (5-7)    unchanged
*               (8)      unchanged
*               (9-11)   unchanged
*               (12)     unchanged
*               (13-21)  unchanged
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*     J        i         status: +1 = dubious year (Note 2)
*                                 0 = OK
*                                -1 = unacceptable date
*
*  Notes:
*
*  1)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2-part quasi Julian Date, as
*      it implements the leap-second-ambiguity convention just
*      described.
*
*  2)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the future
*      to be trusted.  See iau_DAT for further details.
*
```

```
*  3)  UT1−UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1−UTC within +/− 0.9s.  n.b. This
*      practice is under review, and in the future UT1−UTC may grow
*      essentially without limit.
*
*  4)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east−positive
*      (i.e. right−handed), in accordance with geographical convention.
*
*  5)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*      Internally, the polar motion is stored in a form rotated onto
*      the local meridian.
*
*  6)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB), is
*      available, an adequate estimate of hm can be obtained from the
*      expression
*
*            hm = −29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea−level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( −hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to the
*      pressure and that an accurate phpa value is important for
*      precise work.
*
*  7)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  8)  It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  9)  In cases where the caller wishes to supply his own Earth
*      rotation information and refraction constants, the routine
*      iau_APC can be used instead of the present routine.
*
*  10) This is one of several routines that inserts into the ASTROM
*      array star−independent parameters needed for the chain of
*      astrometric transformations ICRS <−> GCRS <−> CIRS <−> observed.
*
*      The various routines support different classes of observer and
*      portions of the transformation chain:
*
*            routines           observer        transformation
*
*        iau_APCG iau_APCG13    geocentric      ICRS <−> GCRS
*        iau_APCI iau_APCI13    terrestrial     ICRS <−> CIRS
*        iau_APCO iau_APCO13    terrestrial     ICRS <−> observed
*        iau_APCS iau_APCS13    space           ICRS <−> GCRS
*        iau_APER iau_APER13    terrestrial     update Earth rotation
*        iau_APIO iau_APIO13    terrestrial     CIRS <−> observed
*
*      Those with names ending in "13" use contemporary SOFA models to
*      compute the various ephemerides.  The others accept ephemerides
*      supplied by the caller.
*
*      The transformation from ICRS to GCRS covers space motion,
```

```
*      parallax, light deflection, and aberration.  From GCRS to CIRS
*      comprises frame bias and precession-nutation.  From CIRS to
*      observed takes account of Earth rotation, polar motion, diurnal
*      aberration and parallax (unless subsumed into the ICRS <-> GCRS
*      transformation), and atmospheric refraction.
*
*  11) The context array ASTROM produced by this routine is used by
*      iau_ATIOQ and iau_ATOIQ.
*
*  Called:
*     iau_UTCTAI   UTC to TAI
*     iau_TAITT    TAI to TT
*     iau_UTCUT1   UTC to UT1
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_REFCO    refraction constants for given ambient conditions
*     iau_APIO     astrometry parameters, CIRS-observed
*
*-
```

```
      SUBROUTINE iau_ATCI13 ( RC, DC, PR, PD, PX, RV, DATE1, DATE2,
     :                        RI, DI, EO )
*+
*  - - - - - - - - - - - -
*   i a u _ A T C I 1 3
*  - - - - - - - - - - - -
*
*  Transform ICRS star data, epoch J2000.0, to CIRS.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC       d      ICRS right ascension at J2000.0 (radians, Note 1)
*     DC       d      ICRS declination at J2000.0 (radians, Note 1)
*     PR       d      RA proper motion (radians/year; Note 2)
*     PD       d      Dec proper motion (radians/year)
*     PX       d      parallax (arcsec)
*     RV       d      radial velocity (km/s, +ve if receding)
*     DATE1    d      TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 3)
*
*  Returned:
*     RI,DI    d      CIRS geocentric RA,Dec (radians)
*     EO       d      equation of the origins (ERA-GST, Note 5)
*
*  Notes:
*
*  1) Star data for an epoch other than J2000.0 (for example from the
*     Hipparcos catalog, which has an epoch of J1991.25) will require a
*     preliminary call to iau_PMSAFE before use.
*
*  2) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  3) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  4) The available accuracy is better than 1 milliarcsecond, limited
*     mainly by the precession-nutation model that is used, namely
*     IAU 2000A/2006.  Very close to solar system bodies, additional
*     errors of up to several milliarcseconds can occur because of
*     unmodeled light deflection;  however, the Sun’s contribution is
*     taken into account, to first order.  The accuracy limitations of
*     the SOFA routine iau_EPV00 (used to compute Earth position and
*     velocity) can contribute aberration errors of up to
*     5 microarcseconds.  Light deflection at the Sun’s limb is
*     uncertain at the 0.4 mas level.
*
*  5) Should the transformation to (equinox based) apparent place be
```

```
*     required rather than (CIO based) intermediate place, subtract the
*     equation of the origins from the returned right ascension:
*     RA = RI - EO. (The iau_ANP routine can then be applied, as
*     required, to keep the result in the conventional 0-2pi range.)
*
*  Called:
*     iau_APCI13   astrometry parameters, ICRS-CIRS, 2013
*     iau_ATCIQ    quick ICRS to CIRS
*
*-
```

```
      SUBROUTINE iau_ATCIQ ( RC, DC, PR, PD, PX, RV, ASTROM, RI, DI )
*+
*  - - - - - - - - - -
*   i a u _ A T C I Q
*  - - - - - - - - - -
*
*  Quick ICRS, epoch J2000.0, to CIRS transformation, given precomputed
*  star-independent astrometry parameters.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are to be transformed for one date.  The
*  star-independent parameters can be obtained by calling one of the
*  routines iau_APCI[13], iau_APCG[13], iau_APCO[13] or iau_APCS[13].
*
*  If the parallax and proper motions are zero the iau_ATCIQZ routine
*  can be used instead.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC,DC    d      ICRS RA,Dec at J2000.0 (radians)
*     PR       d      RA proper motion (radians/year; Note 3)
*     PD       d      Dec proper motion (radians/year)
*     PX       d      parallax (arcsec)
*     RV       d      radial velocity (km/s, +ve if receding)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s’ (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Returned:
*     RI,DI    d      CIRS RA,Dec (radians)
*
*  Notes:
*
*  1) All the vectors are with respect to BCRS axes.
*
*  2) Star data for an epoch other than J2000.0 (for example from the
*     Hipparcos catalog, which has an epoch of J1991.25) will require a
*     preliminary call to iau_PMSAFE before use.
*
*  3) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  Called:
*     iau_PMPX     proper motion and parallax
*     iau_LDSUN    light deflection by the Sun
*     iau_AB       stellar aberration
*     iau_RXP      product of r-matrix and pv-vector
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*-
```

```
      SUBROUTINE iau_ATCIQN ( RC, DC, PR, PD, PX, RV, ASTROM, N, B,
     :                        RI, DI )
*+
*  - - - - - - - - - - - -
*   i a u _ A T C I Q N
*  - - - - - - - - - - - -
*
*  Quick ICRS, epoch J2000.0, to CIRS transformation, given precomputed
*  star-independent astrometry parameters plus a list of light-
*  deflecting bodies.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are to be transformed for one date.  The
*  star-independent parameters can be obtained by calling one of the
*  routines iau_APCI[13], iau_APCG[13], iau_APCO[13] or iau_APCS[13].
*
*  If the only light-deflecting body to be taken into account is the
*  Sun, the iau_ATCIQ routine can be used instead.  If in addition the
*  parallax and proper motions are zero, the iau_ATCIQZ routine can be
*  used.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC,DC    d      ICRS RA,Dec at J2000.0 (radians, Note 1)
*     PR       d      RA proper motion (radians/year; Note 2)
*     PD       d      Dec proper motion (radians/year)
*     PX       d      parallax (arcsec)
*     RV       d      radial velocity (km/s, +ve if receding)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*     N        i       number of bodies (Note 3)
*     B        d(8,N)  data for each of the NB bodies (Notes 3,4):
*               (1,I)    mass of the body (solar masses, Note 5)
*               (2,I)    deflection limiter (Note 6)
*               (3-5,I)  barycentric position of the body (au)
*               (6-8,I)  barycentric velocity of the body (au/day)
*
*  Returned:
*     RI,DI    d      CIRS RA,Dec (radians)
*
*  Notes:
*
*  1) Star data for an epoch other than J2000.0 (for example from the
*     Hipparcos catalog, which has an epoch of J1991.25) will require a
*     preliminary call to iau_PMSAFE before use.
*
*  2) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  3) The array B contains N entries, one for each body to be
*     considered.  If N = 0, no gravitational light deflection will be
*     applied, not even for the Sun.
*
*  4) The array B should include an entry for the Sun as well as for any
```

```
*     planet or other body to be taken into account.  The entries should
*     be in the order in which the light passes the body.
*
*  5) In the entry in the B array for body I, the mass parameter B(1,I)
*     can, as required, be adjusted in order to allow for such effects
*     as quadrupole field.
*
*  6) The deflection limiter parameter B(2,I) is phi^2/2, where phi is
*     the angular separation (in radians) between star and body at which
*     limiting is applied.  As phi shrinks below the chosen threshold,
*     the deflection is artificially reduced, reaching zero for phi = 0.
*     Example values suitable for a terrestrial observer, together with
*     masses, are as follows:
*
*        body I     B(1,I)         B(2,I)
*
*        Sun        1D0            6D-6
*        Jupiter    0.00095435D0   3D-9
*        Saturn     0.00028574D0   3D-10
*
*  7) For efficiency, validation of the B array is omitted.  The
*     supplied masses must be greater than zero, the position and
*     velocity vectors must be right, and the deflection limiter
*     greater than zero.
*
*  Called:
*     iau_PMPX     proper motion and parallax
*     iau_LDN      light deflection by n bodies
*     iau_AB       stellar aberration
*     iau_RXP      product of r-matrix and pv-vector
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*-
```

```
      SUBROUTINE iau_ATCIQZ ( RC, DC, ASTROM, RI, DI )
*+
*  - - - - - - - - - - -
*   i a u _ A T C I Q Z
*  - - - - - - - - - - -
*
*  Quick ICRS to CIRS transformation, given precomputed star-independent
*  astrometry parameters, and assuming zero parallax and proper motion.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are to be transformed for one date.  The
*  star-independent parameters can be obtained by calling one of the
*  routines iau_APCI[13], iau_APCG[13], iau_APCO[13] or iau_APCS[13].
*
*  The corresponding routine for the case of non-zero parallax and
*  proper motion is iau_ATCIQ.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC,DC    d      ICRS astrometric RA,Dec (radians)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Returned:
*     RI,DI    d      CIRS RA,Dec (radians)
*
*  Note:
*
*     All the vectors are with respect to BCRS axes.
*
*  References:
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013).
*
*     Klioner, Sergei A., "A practical relativistic model for micro-
*     arcsecond astrometry in space", Astr. J. 125, 1580-1597 (2003).
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_LDSUN    light deflection due to Sun
*     iau_AB       stellar aberration
*     iau_RXP      product of r-matrix and p-vector
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range +/- pi
*
*-
```

```
      SUBROUTINE iau_ATCO13 ( RC, DC, PR, PD, PX, RV,
     :                        UTC1, UTC2, DUT1, ELONG, PHI, HM, XP, YP,
     :                        PHPA, TC, RH, WL,
     :                        AOB, ZOB, HOB, DOB, ROB, EO, J )
*+
*  - - - - - - - - - - -
*   i a u _ A T C O 1 3
*  - - - - - - - - - - -
*
*  ICRS RA,Dec to observed place.  The caller supplies UTC, site
*  coordinates, ambient air conditions and observing wavelength.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC,DC    d      ICRS right ascension at J2000.0 (radians, Note 1)
*     PR       d      RA proper motion (radians/year; Note 2)
*     PD       d      Dec proper motion (radians/year)
*     PX       d      parallax (arcsec)
*     RV       d      radial velocity (km/s, +ve if receding)
*     UTC1     d      UTC as a 2-part...
*     UTC2     d      ...quasi Julian Date (Notes 3-4)
*     DUT1     d      UT1-UTC (seconds, Note 5)
*     ELONG    d      longitude (radians, east +ve, Note 6)
*     PHI      d      latitude (geodetic, radians, Note 6)
*     HM       d      height above ellipsoid (m, geodetic, Notes 6,8)
*     XP,YP    d      polar motion coordinates (radians, Note 7)
*     PHPA     d      pressure at the observer (hPa = mB, Note 8)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0-1)
*     WL       d      wavelength (micrometers, Note 9)
*
*  Returned:
*     AOB      d      observed azimuth (radians: N=0,E=90)
*     ZOB      d      observed zenith distance (radians)
*     HOB      d      observed hour angle (radians)
*     DOB      d      observed declination (radians)
*     ROB      d      observed right ascension (CIO-based, radians)
*     EO       d      equation of the origins (ERA-GST)
*     J        i      status: +1 = dubious year (Note 4)
*                              0 = OK
*                             -1 = unacceptable date
*
*  Notes:
*
*  1)  Star data for an epoch other than J2000.0 (for example from the
*      Hipparcos catalog, which has an epoch of J1991.25) will require
*      a preliminary call to iau_PMSAFE before use.
*
*  2)  The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  3)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2-part quasi Julian Date, as
*      it implements the leap-second-ambiguity convention just
*      described.
*
*  4)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the
*      future to be trusted.  See iau_DAT for further details.
*
```

```
*  5)  UT1-UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1-UTC within +/- 0.9s.  n.b. This
*      practice is under review, and in the future UT1-UTC may grow
*      essentially without limit.
*
*  6)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east-positive
*      (i.e. right-handed), in accordance with geographical convention.
*
*  7)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*  8)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB),
*      is available, an adequate estimate of hm can be obtained from
*      the expression
*
*            hm = -29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea-level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( -hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to
*      the pressure and that an accurate phpa value is important for
*      precise work.
*
*  9)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  10) The accuracy of the result is limited by the corrections for
*      refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*      Providing the meteorological parameters are known accurately and
*      there are no gross local effects, the predicted observed
*      coordinates should be within 0.05 arcsec (optical) or 1 arcsec
*      (radio) for a zenith distance of less than 70 degrees, better
*      than 30 arcsec (optical or radio) at 85 degrees and better than
*      20 arcmin (optical) or 30 arcmin (radio) at the horizon.
*
*      Without refraction, the complementary routines iau_ATCO13 and
*      iau_ATOC13 are self-consistent to better than 1 microarcsecond
*      all over the celestial sphere.  With refraction included,
*      consistency falls off at high zenith distances, but is still
*      better than 0.05 arcsec at 85 degrees.
*
*  11) "Observed" Az,ZD means the position that would be seen by a
*      perfect geodetically aligned theodolite.  (Zenith distance is
*      used rather than altitude in order to reflect the fact that no
*      allowance is made for depression of the horizon.)  This is
*      related to the observed HA,Dec via the standard rotation, using
*      the geodetic latitude (corrected for polar motion), while the
*      observed HA and RA are related simply through the Earth rotation
*      angle and the site longitude.  "Observed" RA,Dec or HA,Dec thus
*      means the position that would be seen by a perfect equatorial
*      with its polar axis aligned to the Earth's axis of rotation.
*
*  12) It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  Called:
*     iau_APCO13   astrometry parameters, ICRS-observed, 2013
```

```
*     iau_ATCIQ    quick ICRS to CIRS
*     iau_ATIOQ    quick CIRS to observed
*
*-
```

```
      SUBROUTINE iau_ATIC13 ( RI, DI, DATE1, DATE2, RC, DC, EO )
*+
*  - - - - - - - - - - -
*   i a u _ A T I C 1 3
*  - - - - - - - - - - -
*
*  Transform star RA,Dec from geocentric CIRS to ICRS astrometric.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RI,DI    d      CIRS geocentric RA,Dec (radians)
*     DATE1    d      TDB as a 2-part...
*     DATE2    d      ...Julian Date (Note 1)
*
*  Returned:
*     RC,DC    d      ICRS astrometric RA,Dec (radians)
*     EO       d      equation of the origins (ERA-GST, Note 4)
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.  For most
*     applications of this routine the choice will not be at all
*     critical.
*
*     TT can be used instead of TDB without any significant impact on
*     accuracy.
*
*  2) Iterative techniques are used for the aberration and light
*     deflection corrections so that the routines iau_ATIC13 (or
*     iau_ATICQ) and iau_ATCI13 (or iau_ATCIQ) are accurate inverses;
*     even at the edge of the Sun's disk the discrepancy is only about
*     1 nanoarcsecond.
*
*  3) The available accuracy is better than 1 milliarcsecond, limited
*     mainly by the precession-nutation model that is used, namely
*     IAU 2000A/2006.  Very close to solar system bodies, additional
*     errors of up to several milliarcseconds can occur because of
*     unmodeled light deflection;  however, the Sun's contribution is
*     taken into account, to first order.  The accuracy limitations of
*     the SOFA routine iau_EPV00 (used to compute Earth position and
*     velocity) can contribute aberration errors of up to
*     5 microarcseconds.  Light deflection at the Sun's limb is
*     uncertain at the 0.4 mas level.
*
*  4) Should the transformation to (equinox based) J2000.0 mean place
*     be required rather than (CIO based) ICRS coordinates, subtract the
*     equation of the origins from the returned right ascension:
*     RA = RI - EO.  (The iau_ANP routine can then be applied, as
*     required, to keep the result in the conventional 0-2pi range.)
*
*  Called:
```

```
*     iau_APCI13   astrometry parameters, ICRS-CIRS, 2013
*     iau_ATICQ    quick CIRS to ICRS astrometric
*
*-
```

```
      SUBROUTINE iau_ATICQ ( RI, DI, ASTROM, RC, DC )
*+
*  - - - - - - - - - -
*   i a u _ A T I C Q
*  - - - - - - - - - -
*
*  Quick CIRS RA,Dec to ICRS astrometric place, given the star-
*  independent astrometry parameters.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are all to be transformed for one date.
*  The star-independent astrometry parameters can be obtained by
*  calling one of the routines iau_APCI[13], iau_APCG[13], iau_APCO[13]
*  or iau_APCS[13].
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RI,DI    d      CIRS RA,Dec (radians)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Returned:
*     RC,DC    d      ICRS astrometric RA,Dec (radians)
*
*  Notes:
*
*  1) Only the Sun is taken into account in the light deflection
*     correction.
*
*  2) Iterative techniques are used for the aberration and light
*     deflection corrections so that the routines iau_ATIC13 (or
*     iau_ATICQ) and iau_ATCI13 (or iau_ATCIQ) are accurate inverses;
*     even at the edge of the Sun's disk the discrepancy is only about
*     1 nanoarcsecond.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_ZP       zero p-vector
*     iau_AB       stellar aberration
*     iau_LDSUN    light deflection by the Sun
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range +/- pi
*
*-
```

```
      SUBROUTINE iau_ATICQN ( RI, DI, ASTROM, N, B, RC, DC )
*+
*  - - - - - - - - - - -
*   i a u _ A T I C Q N
*  - - - - - - - - - - -
*
*  Quick CIRS to ICRS astrometric place transformation, given the
*  star-independent astrometry parameters plus a list of light-
*  deflecting bodies.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are all to be transformed for one date.
*  The star-independent astrometry parameters can be obtained by
*  calling one of the routines iau_APCI[13], iau_APCG[13], iau_APCO[13]
*  or iau_APCS[13].
*
*  If the only light-deflecting body to be taken into account is the
*  Sun, the iau_ATICQ routine can be used instead.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RI,DI    d      CIRS RA,Dec (radians)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*     N        i       number of bodies (Note 3)
*     B        d(8,N)  data for each of the NB bodies (Notes 3,4):
*               (1,I)    mass of the body (solar masses, Note 5)
*               (2,I)    deflection limiter (Note 6)
*               (3-5,I)  barycentric position of the body (au)
*               (6-8,I)  barycentric velocity of the body (au/day)
*
*  Returned:
*     RC,DC    d      ICRS astrometric RA,Dec (radians)
*
*  Notes:
*
*  1) Iterative techniques are used for the aberration and light
*     deflection corrections so that the routines iau_ATICQN and
*     iau_ATCIQN are accurate inverses; even at the edge of the Sun's
*     disk the discrepancy is only about 1 nanoarcsecond.
*
*  2) If the only light-deflecting body to be taken into account is the
*     Sun, the iau_ATICQ routine can be used instead.
*
*  3) The array B contains N entries, one for each body to be
*     considered.  If N = 0, no gravitational light deflection will be
*     applied, not even for the Sun.
*
*  4) The array B should include an entry for the Sun as well as for any
*     planet or other body to be taken into account.  The entries should
*     be in the order in which the light passes the body.
*
*  5) In the entry in the B array for body I, the mass parameter B(1,I)
```

```
*     can, as required, be adjusted in order to allow for such effects
*     as quadrupole field.
*
*  6) The deflection limiter parameter B(2,I) is phi^2/2, where phi is
*     the angular separation (in radians) between star and body at which
*     limiting is applied.  As phi shrinks below the chosen threshold,
*     the deflection is artificially reduced, reaching zero for phi = 0.
*     Example values suitable for a terrestrial observer, together with
*     masses, are as follows:
*
*        body I     B(1,I)         B(2,I)
*
*        Sun        1D0            6D-6
*        Jupiter    0.00095435D0   3D-9
*        Saturn     0.00028574D0   3D-10
*
*  7) For efficiency, validation of the contents of the B array is
*     omitted.  The supplied masses must be greater than zero, the
*     position and velocity vectors must be right, and the deflection
*     limiter greater than zero.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_ZP       zero p-vector
*     iau_AB       stellar aberration
*     iau_LDN      light deflection by n bodies
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range +/- pi
*
*-
```

```
      SUBROUTINE iau_ATIO13 ( RI, DI, UTC1, UTC2, DUT1,
     :                        ELONG, PHI, HM, XP, YP, PHPA, TC, RH, WL,
     :                        AOB, ZOB, HOB, DOB, ROB, J )
*+
*  − − − − − − − − − − −
*   i a u _ A T I O 1 3
*  − − − − − − − − − − −
*
*  CIRS RA,Dec to observed place.  The caller supplies UTC, site
*  coordinates, ambient air conditions and observing wavelength.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RI       d      CIRS right ascension (CIO−based, radians)
*     DI       d      CIRS declination (radians)
*     UTC1     d      UTC as a 2−part...
*     UTC2     d      ...quasi Julian Date (Notes 1,2)
*     DUT1     d      UT1−UTC (seconds, Note 3)
*     ELONG    d      longitude (radians, east +ve, Note 4)
*     PHI      d      geodetic latitude (radians, Note 4)
*     HM       d      height above ellipsoid (m, geodetic Notes 4,6)
*     XP,YP    d      polar motion coordinates (radians, Note 5)
*     PHPA     d      pressure at the observer (hPa = mB, Note 6)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0−1)
*     WL       d      wavelength (micrometers, Note 7)
*
*  Returned:
*     AOB      d      observed azimuth (radians: N=0,E=90)
*     ZOB      d      observed zenith distance (radians)
*     HOB      d      observed hour angle (radians)
*     DOB      d      observed declination (radians)
*     ROB      d      observed right ascension (CIO−based, radians)
*     J        i      status: +1 = dubious year (Note 2)
*                              0 = OK
*                             −1 = unacceptable date
*
*  Notes:
*
*  1)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2−part quasi Julian Date, as
*      it implements the leap−second−ambiguity convention just
*      described.
*
*  2)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the
*      future to be trusted.  See iau_DAT for further details.
*
*  3)  UT1−UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1−UTC within +/− 0.9s.  n.b. This
*      practice is under review, and in the future UT1−UTC may grow
*      essentially without limit.
*
*  4)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east−positive
*      (i.e. right−handed), in accordance with geographical convention.
*
```

```
*  5)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*  6)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB), is
*      available, an adequate estimate of hm can be obtained from the
*      expression
*
*            hm = -29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea-level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( -hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to
*      the pressure and that an accurate phpa value is important for
*      precise work.
*
*  7)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  8)  "Observed" Az,ZD means the position that would be seen by a
*      perfect geodetically aligned theodolite.  (Zenith distance is
*      used rather than altitude in order to reflect the fact that no
*      allowance is made for depression of the horizon.)  This is
*      related to the observed HA,Dec via the standard rotation, using
*      the geodetic latitude (corrected for polar motion), while the
*      observed HA and RA are related simply through the Earth rotation
*      angle and the site longitude.  "Observed" RA,Dec or HA,Dec thus
*      means the position that would be seen by a perfect equatorial
*      with its polar axis aligned to the Earth's axis of rotation.
*
*  9)  The accuracy of the result is limited by the corrections for
*      refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*      Providing the meteorological parameters are known accurately and
*      there are no gross local effects, the predicted astrometric
*      coordinates should be within 0.05 arcsec (optical) or 1 arcsec
*      (radio) for a zenith distance of less than 70 degrees, better
*      than 30 arcsec (optical or radio) at 85 degrees and better
*      than 20 arcmin (optical) or 30 arcmin (radio) at the horizon.
*
*  10) The complementary routines iau_ATIO13 and iau_ATOI13 are self-
*      consistent to better than 1 microarcsecond all over the
*      celestial sphere.
*
*  11) It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  Called:
*     iau_APIO13   astrometry parameters, CIRS-observed, 2013
*     iau_ATIOQ    quick CIRS to observed
*
*-
```

```
      SUBROUTINE iau_ATIOQ ( RI, DI, ASTROM, AOB, ZOB, HOB, DOB, ROB )
*+
*  - - - - - - - - - -
*   i a u _ A T I O Q
*  - - - - - - - - - -
*
*  Quick CIRS to observed place transformation.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are all to be transformed for one date.
*  The star-independent astrometry parameters can be obtained by
*  calling iau_APIO[13] or iau_APCO[13].
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RI       d      CIRS right ascension
*     DI       d      CIRS declination
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s’ (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Returned:
*     AOB      d      observed azimuth (radians: N=0,E=90)
*     ZOB      d      observed zenith distance (radians)
*     HOB      d      observed hour angle (radians)
*     DOB      d      observed declination (CIO-based, radians)
*     ROB      d      observed right ascension (CIO-based, radians)
*
*  Notes:
*
*  1) This routine returns zenith distance rather than altitude in
*     order to reflect the fact that no allowance is made for
*     depression of the horizon.
*
*  2) The accuracy of the result is limited by the corrections for
*     refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*     Providing the meteorological parameters are known accurately and
*     there are no gross local effects, the predicted observed
*     coordinates should be within 0.05 arcsec (optical) or 1 arcsec
*     (radio) for a zenith distance of less than 70 degrees, better
*     than 30 arcsec (optical or radio) at 85 degrees and better than
*     20 arcmin (optical) or 30 arcmin (radio) at the horizon.
*
*     Without refraction, the complementary routines iau_ATIOQ and
*     iau_ATOIQ are self-consistent to better than 1 microarcsecond all
*     over the celestial sphere.  With refraction included, consistency
*     falls off at high zenith distances, but is still better than
*     0.05 arcsec at 85 degrees.
*
*  3) It is advisable to take great care with units, as even unlikely
*     values of the input parameters are accepted and processed in
*     accordance with the models used.
*
*  4) The CIRS RA,Dec is obtained from a star catalog mean place by
```

```
*     allowing for space motion, parallax, the Sun’s gravitational lens
*     effect, annual aberration and precession−nutation.  For star
*     positions in the ICRS, these effects can be applied by means of
*     the iau_ATCI13 (etc.) routines.  Starting from classical "mean
*     place" systems, additional transformations will be needed first.
*
*  5) "Observed" Az,El means the position that would be seen by a
*     perfect geodetically aligned theodolite.  This is obtained from
*     the CIRS RA,Dec by allowing for Earth orientation and diurnal
*     aberration, rotating from equator to horizon coordinates, and then
*     adjusting for refraction.  The HA,Dec is obtained by rotating back
*     into equatorial coordinates, and is the position that would be
*     seen by a perfect equatorial with its polar axis aligned to the
*     Earth’s axis of rotation.  Finally, the RA is obtained by
*     subtracting the HA from the local ERA.
*
*  6) The star−independent CIRS−to−observed−place parameters in ASTROM
*     may be computed with iau_APIO[13] or iau_APCO[13].  If nothing has
*     changed significantly except the time, iau_APER[13] may be used
*     to perform the requisite adjustment to the ASTROM array.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_C2S      p−vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*−
```

```
      SUBROUTINE iau_ATOC13 ( TYPE, OB1, OB2, UTC1, UTC2, DUT1,
     :                        ELONG, PHI, HM, XP, YP, PHPA, TC, RH, WL,
     :                        RC, DC, J )
*+
*  - - - - - - - - - - -
*   i a u _ A T O C 1 3
*  - - - - - - - - - - -
*
*  Observed place at a groundbased site to to ICRS astrometric RA,Dec.
*  The caller supplies UTC, site coordinates, ambient air conditions
*  and observing wavelength.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TYPE     c*(*)  type of coordinates - 'R', 'H' or 'A' (Notes 1,2)
*     OB1      d      observed Az, HA or RA (radians; Az is N=0,E=90)
*     OB2      d      observed ZD or Dec (radians)
*     UTC1     d      UTC as a 2-part...
*     UTC2     d      ...quasi Julian Date (Notes 3,4)
*     DUT1     d      UT1-UTC (seconds, Note 5)
*     ELONG    d      longitude (radians, east +ve, Note 6)
*     PHI      d      geodetic latitude (radians, Note 6)
*     HM       d      height above ellipsoid (m, geodetic Notes 6,8)
*     XP,YP    d      polar motion coordinates (radians, Note 7)
*     PHPA     d      pressure at the observer (hPa = mB, Note 8)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0-1)
*     WL       d      wavelength (micrometers, Note 9)
*
*  Returned:
*     RC,DC    d      ICRS astrometric RA,Dec (radians)
*     J        i      status: +1 = dubious year (Note 4)
*                              0 = OK
*                             -1 = unacceptable date
*
*  Notes:
*
*  1)  "Observed" Az,ZD means the position that would be seen by a
*      perfect geodetically aligned theodolite.  (Zenith distance is
*      used rather than altitude in order to reflect the fact that no
*      allowance is made for depression of the horizon.)  This is
*      related to the observed HA,Dec via the standard rotation, using
*      the geodetic latitude (corrected for polar motion), while the
*      observed HA and RA are related simply through the Earth rotation
*      angle and the site longitude.  "Observed" RA,Dec or HA,Dec thus
*      means the position that would be seen by a perfect equatorial
*      with its polar axis aligned to the Earth's axis of rotation.
*
*  2)  Only the first character of the TYPE argument is significant.
*      'R' or 'r' indicates that OB1 and OB2 are the observed right
*      ascension and declination;  'H' or 'h' indicates that they are
*      hour angle (west +ve) and declination;  anything else ('A' or
*      'a' is recommended) indicates that OB1 and OB2 are azimuth
*      (north zero, east 90 deg) and zenith distance.
*
*  3)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2-part quasi Julian Date, as
*      it implements the leap-second-ambiguity convention just
*      described.
```

```
*
*  4)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the
*      future to be trusted.  See iau_DAT for further details.
*
*  5)  UT1-UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1-UTC within +/- 0.9s.  n.b. This
*      practice is under review, and in the future UT1-UTC may grow
*      essentially without limit.
*
*  6)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east-positive
*      (i.e. right-handed), in accordance with geographical convention.
*
*  7)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*  8)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB), is
*      available, an adequate estimate of hm can be obtained from the
*      expression
*
*            hm = -29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea-level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( -hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to
*      the pressure and that an accurate phpa value is important for
*      precise work.
*
*  9)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  10) The accuracy of the result is limited by the corrections for
*      refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*      Providing the meteorological parameters are known accurately and
*      there are no gross local effects, the predicted astrometric
*      coordinates should be within 0.05 arcsec (optical) or 1 arcsec
*      (radio) for a zenith distance of less than 70 degrees, better
*      than 30 arcsec (optical or radio) at 85 degrees and better
*      than 20 arcmin (optical) or 30 arcmin (radio) at the horizon.
*
*      Without refraction, the complementary routines iau_ATCO13 and
*      iau_ATOC13 are self-consistent to better than 1 microarcsecond
*      all over the celestial sphere.  With refraction included,
*      consistency falls off at high zenith distances, but is still
*      better than 0.05 arcsec at 85 degrees.
*
*  11) It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  Called:
*     iau_APCO13   astrometry parameters, ICRS-observed
*     iau_ATOIQ    quick observed to CIRS
*     iau_ATICQ    quick CIRS to ICRS
*
*-
```

```
      SUBROUTINE iau_ATOI13 ( TYPE, OB1, OB2, UTC1, UTC2, DUT1,
     :                        ELONG, PHI, HM, XP, YP, PHPA, TC, RH, WL,
     :                        RI, DI, J )
*+
*  − − − − − − − − − − −
*   i a u _ A T O I 1 3
*  − − − − − − − − − − −
*
*  Observed place to CIRS.  The caller supplies UTC, site coordinates,
*  ambient air conditions and observing wavelength.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TYPE     c*(*)  type of coordinates − ’R’, ’H’ or ’A’ (Notes 1,2)
*     OB1      d      observed Az, HA or RA (radians; Az is N=0,E=90)
*     OB2      d      observed ZD or Dec (radians)
*     UTC1     d      UTC as a 2−part...
*     UTC2     d      ...quasi Julian Date (Notes 3,4)
*     DUT1     d      UT1−UTC (seconds, Note 5)
*     ELONG    d      longitude (radians, east +ve, Note 6)
*     PHI      d      geodetic latitude (radians, Note 6)
*     HM       d      height above the ellipsoid (meters, Notes 6,8)
*     XP,YP    d      polar motion coordinates (radians, Note 7)
*     PHPA     d      pressure at the observer (hPa = mB, Note 8)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0−1)
*     WL       d      wavelength (micrometers, Note 9)
*
*  Returned:
*     RI       d      CIRS right ascension (CIO−based, radians)
*     DI       d      CIRS declination (radians)
*     J        i      status: +1 = dubious year (Note 2)
*                              0 = OK
*                             −1 = unacceptable date
*
*  Notes:
*
*  1)  "Observed" Az,ZD means the position that would be seen by a
*      perfect geodetically aligned theodolite.  (Zenith distance is
*      used rather than altitude in order to reflect the fact that no
*      allowance is made for depression of the horizon.)  This is
*      related to the observed HA,Dec via the standard rotation, using
*      the geodetic latitude (corrected for polar motion), while the
*      observed HA and RA are related simply through the Earth rotation
*      angle and the site longitude.  "Observed" RA,Dec or HA,Dec thus
*      means the position that would be seen by a perfect equatorial
*      with its polar axis aligned to the Earth’s axis of rotation.
*
*  2)  Only the first character of the TYPE argument is significant.
*      ’R’ or ’r’ indicates that OB1 and OB2 are the observed right
*      ascension and declination;  ’H’ or ’h’ indicates that they are
*      hour angle (west +ve) and declination;  anything else (’A’ or
*      ’a’ is recommended) indicates that OB1 and OB2 are azimuth
*      (north zero, east 90 deg) and zenith distance.
*
*  3)  UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*      convenient way between the two arguments, for example where UTC1
*      is the Julian Day Number and UTC2 is the fraction of a day.
*
*      However, JD cannot unambiguously represent UTC during a leap
*      second unless special measures are taken.  The convention in the
*      present routine is that the JD day represents UTC days whether
*      the length is 86399, 86400 or 86401 SI seconds.
*
*      Applications should use the routine iau_DTF2D to convert from
*      calendar date and time of day into 2−part quasi Julian Date, as
*      it implements the leap−second−ambiguity convention just
*      described.
```

```
*
*  4)  The warning status "dubious year" flags UTCs that predate the
*      introduction of the time scale or that are too far in the
*      future to be trusted.  See iau_DAT for further details.
*
*  5)  UT1-UTC is tabulated in IERS bulletins.  It increases by exactly
*      one second at the end of each positive UTC leap second,
*      introduced in order to keep UT1-UTC within +/- 0.9s.  n.b. This
*      practice is under review, and in the future UT1-UTC may grow
*      essentially without limit.
*
*  6)  The geographical coordinates are with respect to the WGS84
*      reference ellipsoid.  TAKE CARE WITH THE LONGITUDE SIGN:  the
*      longitude required by the present routine is east-positive
*      (i.e. right-handed), in accordance with geographical convention.
*
*  7)  The polar motion XP,YP can be obtained from IERS bulletins.  The
*      values are the coordinates (in radians) of the Celestial
*      Intermediate Pole with respect to the International Terrestrial
*      Reference System (see IERS Conventions 2003), measured along the
*      meridians 0 and 90 deg west respectively.  For many applications,
*      XP and YP can be set to zero.
*
*  8)  If hm, the height above the ellipsoid of the observing station
*      in meters, is not known but phpa, the pressure in hPa (=mB), is
*      available, an adequate estimate of hm can be obtained from the
*      expression
*
*            hm = -29.3 * tsl * log ( phpa / 1013.25 );
*
*      where tsl is the approximate sea-level air temperature in K
*      (See Astrophysical Quantities, C.W.Allen, 3rd edition, section
*      52).  Similarly, if the pressure phpa is not known, it can be
*      estimated from the height of the observing station, hm, as
*      follows:
*
*            phpa = 1013.25 * exp ( -hm / ( 29.3 * tsl ) );
*
*      Note, however, that the refraction is nearly proportional to
*      the pressure and that an accurate phpa value is important for
*      precise work.
*
*  9)  The argument WL specifies the observing wavelength in
*      micrometers.  The transition from optical to radio is assumed to
*      occur at 100 micrometers (about 3000 GHz).
*
*  10) The accuracy of the result is limited by the corrections for
*      refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*      Providing the meteorological parameters are known accurately and
*      there are no gross local effects, the predicted astrometric
*      coordinates should be within 0.05 arcsec (optical) or 1 arcsec
*      (radio) for a zenith distance of less than 70 degrees, better
*      than 30 arcsec (optical or radio) at 85 degrees and better
*      than 20 arcmin (optical) or 30 arcmin (radio) at the horizon.
*
*      Without refraction, the complementary routines iau_ATIO13 and
*      iau_ATOI13 are self-consistent to better than 1 microarcsecond
*      all over the celestial sphere.  With refraction included,
*      consistency falls off at high zenith distances, but is still
*      better than 0.05 arcsec at 85 degrees.
*
*  11) It is advisable to take great care with units, as even unlikely
*      values of the input parameters are accepted and processed in
*      accordance with the models used.
*
*  Called:
*     iau_APIO13   astrometry parameters, CIRS-observed, 2013
*     iau_ATOIQ    quick observed to CIRS
*
*-
```

```
      SUBROUTINE iau_ATOIQ ( TYPE, OB1, OB2, ASTROM, RI, DI )
*+
*  - - - - - - - - - -
*   i a u _ A T O I Q
*  - - - - - - - - - -
*
*  Quick observed place to CIRS, given the star-independent astrometry
*  parameters.
*
*  Use of this routine is appropriate when efficiency is important and
*  where many star positions are all to be transformed for one date.
*  The star-independent astrometry parameters can be obtained by calling
*  iau_APIO[13] or iau_APCO[13].
*
*  Status:  support routine.
*
*  Given:
*     TYPE     c*(*)  type of coordinates: 'R', 'H' or 'A' (Note 2)
*     OB1      d      observed Az, HA or RA (radians; Az is N=0,E=90)
*     OB2      d      observed ZD or Dec (radians)
*     ASTROM   d(30)  star-independent astrometry parameters:
*               (1)      PM time interval (SSB, Julian years)
*               (2-4)    SSB to observer (vector, au)
*               (5-7)    Sun to observer (unit vector)
*               (8)      distance from Sun to observer (au)
*               (9-11)   v: barycentric observer velocity (vector, c)
*               (12)     sqrt(1-|v|^2): reciprocal of Lorenz factor
*               (13-21)  bias-precession-nutation matrix
*               (22)     longitude + s' (radians)
*               (23)     polar motion xp wrt local meridian (radians)
*               (24)     polar motion yp wrt local meridian (radians)
*               (25)     sine of geodetic latitude
*               (26)     cosine of geodetic latitude
*               (27)     magnitude of diurnal aberration vector
*               (28)     "local" Earth rotation angle (radians)
*               (29)     refraction constant A (radians)
*               (30)     refraction constant B (radians)
*
*  Returned:
*     RI       d      CIRS right ascension (CIO-based, radians)
*     DI       d      CIRS declination (radians)
*
*  Notes:
*
*  1) "Observed" Az,El means the position that would be seen by a
*     perfect geodetically aligned theodolite.  This is related to the
*     observed HA,Dec via the standard rotation, using the geodetic
*     latitude (corrected for polar motion), while the observed HA and
*     RA are related simply through the Earth rotation angle and the
*     site longitude.  "Observed" RA,Dec or HA,Dec thus means the
*     position that would be seen by a perfect equatorial with its polar
*     axis aligned to the Earth's axis of rotation.  By removing from
*     the observed place the effects of atmospheric refraction and
*     diurnal aberration, the CIRS RA,Dec is obtained.
*
*  2) Only the first character of the type argument is significant.
*     'R' or 'r' indicates that OB1 and OB2 are the observed right
*     ascension and declination;  'H' or 'h' indicates that they are
*     hour angle (west +ve) and declination;  anything else ('A' or
*     'a' is recommended) indicates that OB1 and OB2 are azimuth (north
*     zero, east 90 deg) and zenith distance.  (Zenith distance is used
*     rather than altitude in order to reflect the fact that no
*     allowance is made for depression of the horizon.)
*
*  3) The accuracy of the result is limited by the corrections for
*     refraction, which use a simple A*tan(z) + B*tan^3(z) model.
*     Providing the meteorological parameters are known accurately and
*     there are no gross local effects, the predicted observed
*     coordinates should be within 0D05 arcsec (optical) or 1 arcsec
*     (radio) for a zenith distance of less than 70 degrees, better
*     than 30 arcsec (optical or radio) at 85 degrees and better than
*     20 arcmin (optical) or 30 arcmin (radio) at the horizon.
```

```
*
*     Without refraction, the complementary routines iau_ATIOQ and
*     iau_ATOIQ are self-consistent to better than 1 microarcsecond all
*     over the celestial sphere.  With refraction included, consistency
*     falls off at high zenith distances, but is still better than
*     0.05 arcsec at 85 degrees.
*
*  4) It is advisable to take great care with units, as even unlikely
*     values of the input parameters are accepted and processed in
*     accordance with the models used.
*
*  5) The star-independent astrometry parameters in ASTROM may be
*     computed with iau_APIO13 (or iau_APIO).  If nothing has changed
*     significantly except the time, iau_APER13 (or iau_APER) may be
*     used to perform the requisite adjustment to the ASTROM array.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*-
```

```
      SUBROUTINE iau_BI00 ( DPSIBI, DEPSBI, DRA )
*+
*  - - - - - - - - -
*   i a u _ B I 0 0
*  - - - - - - - - -
*
*  Frame bias components of IAU 2000 precession-nutation models (part of
*  MHB2000 with additions).
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Returned:
*     DPSIBI,DEPSBI  d   longitude and obliquity corrections
*     DRA            d   the ICRS RA of the J2000.0 mean equinox
*
*  Notes:
*
*  1) The frame bias corrections in longitude and obliquity (radians)
*     are required in order to correct for the offset between the GCRS
*     pole and the J2000.0 mean pole.  They define, with respect to the
*     GCRS frame, a J2000.0 mean pole that is consistent with the rest
*     of the IAU 2000A precession-nutation model.
*
*  2) In addition to the displacement of the pole, the complete
*     description of the frame bias requires also an offset in right
*     ascension.  This is not part of the IAU 2000A model, and is from
*     Chapront et al. (2002).  It is returned in radians.
*
*  3) This is a supplemented implementation of one aspect of the IAU
*     2000A nutation model, formally adopted by the IAU General Assembly
*     in 2000, namely MHB2000 (Mathews et al. 2002).
*
*  References:
*
*     Chapront, J., Chapront-Touze, M. & Francou, G., Astron.Astrophys.,
*     387, 700, 2002.
*
*     Mathews, P.M., Herring, T.A., Buffet, B.A., "Modeling of nutation
*     and precession   New nutation series for nonrigid Earth and
*     insights into the Earth’s interior", J.Geophys.Res., 107, B4,
*     2002.  The MHB2000 code itself was obtained on 9th September 2002
*     from ftp://maia.usno.navy.mil/conv2000/chapter5/IAU2000A.
*
*-
```

```
      SUBROUTINE iau_BP00 ( DATE1, DATE2, RB, RP, RBP )
*+
*  - - - - - - - - -
*   i a u _ B P 0 0
*  - - - - - - - - -
*
*  Frame bias and precession, IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RB             d(3,3)    frame bias matrix (Note 2)
*     RP             d(3,3)    precession matrix (Note 3)
*     RBP            d(3,3)    bias-precession matrix (Note 4)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RB transforms vectors from GCRS to mean J2000.0 by
*     applying frame bias.
*
*  3) The matrix RP transforms vectors from J2000.0 mean equator and
*     equinox to mean equator and equinox of date by applying
*     precession.
*
*  4) The matrix RBP transforms vectors from GCRS to mean equator and
*     equinox of date by applying frame bias then precession.  It is the
*     product RP x RB.
*
*  Called:
*     iau_BI00     frame bias components, IAU 2000
*     iau_PR00     IAU 2000 precession adjustments
*     iau_IR       initialize r-matrix to identity
*     iau_RX       rotate around X-axis
*     iau_RY       rotate around Y-axis
*     iau_RZ       rotate around Z-axis
*     iau_RXR      product of two r-matrices
*     iau_CR       copy r-matrix
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
```

*
*_

```
      SUBROUTINE iau_BP06 ( DATE1, DATE2, RB, RP, RBP )
*+
*  - - - - - - - - -
*   i a u _ B P 0 6
*  - - - - - - - - -
*
*  Frame bias and precession, IAU 2006.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RB           d(3,3)    frame bias matrix (Note 2)
*     RP           d(3,3)    precession matrix (Note 3)
*     RBP          d(3,3)    bias-precession matrix (Note 4)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RB transforms vectors from GCRS to mean J2000.0 by
*     applying frame bias.
*
*  3) The matrix RP transforms vectors from mean J2000.0 to mean of date
*     by applying precession.
*
*  4) The matrix RBP transforms vectors from GCRS to mean of date by
*     applying frame bias then precession.  It is the product RP x RB.
*
*  Called:
*     iau_PFW06    bias-precession F-W angles, IAU 2006
*     iau_FW2M     F-W angles to r-matrix
*     iau_PMAT06   PB matrix, IAU 2006
*     iau_TR       transpose r-matrix
*     iau_RXR      product of two r-matrices
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_BPN2XY ( RBPN, X, Y )
*+
*  - - - - - - - - - - -
*   i a u _ B P N 2 X Y
*  - - - - - - - - - - -
*
*  Extract from the bias-precession-nutation matrix the X,Y coordinates
*  of the Celestial Intermediate Pole.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RBPN      d(3,3)    celestial-to-true matrix (Note 1)
*
*  Returned:
*     X,Y         d       Celestial Intermediate Pole (Note 2)
*
*  Notes:
*
*  1) The matrix RBPN transforms vectors from GCRS to true equator (and
*     CIO or equinox) of date, and therefore the Celestial Intermediate
*     Pole unit vector is the bottom row of the matrix.
*
*  2) X,Y are components of the Celestial Intermediate Pole unit vector
*     in the Geocentric Celestial Reference System.
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*-
```

```
      SUBROUTINE iau_C2I00A ( DATE1, DATE2, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I 0 0 A
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date using the
*  IAU 2000A precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  3) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_C2I00B routine.
*
*  Called:
*     iau_PNM00A   classical NPB matrix, IAU 2000A
*     iau_C2IBPN   celestial-to-intermediate matrix, given NPB matrix
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
```

*

* _

```
      SUBROUTINE iau_C2I00B ( DATE1, DATE2, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I 0 0 B
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date using the
*  IAU 2000B precession-nutation model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0       -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  3) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_C2I00A routine.
*
*  Called:
*     iau_PNM00B   classical NPB matrix, IAU 2000B
*     iau_C2IBPN   celestial-to-intermediate matrix, given NPB matrix
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
```

*
*_

```
      SUBROUTINE iau_C2I06A ( DATE1, DATE2, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I 0 6 A
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date using the
*  IAU 2006 precession and IAU 2000A nutation models.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*     iau_C2IXYS   celestial-to-intermediate matrix, given X,Y and s
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_C2IBPN ( DATE1, DATE2, RBPN, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I B P N
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date given
*  the bias-precession-nutation matrix.  IAU 2000.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*     RBPN         d(3,3)    celestial-to-true matrix (Note 2)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DATE1          DATE2
*
*          2450123.7D0        0D0        (JD method)
*           2451545D0      -1421.3D0     (J2000 method)
*          2400000.5D0     50123.2D0     (MJD method)
*          2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RBPN transforms vectors from GCRS to true equator (and
*     CIO or equinox) of date.  Only the CIP (bottom row) is used.
*
*  3) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  4) Although its name does not include "00", this routine is in fact
*     specific to the IAU 2000 models.
*
*  Called:
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_C2IXY    celestial-to-intermediate matrix, given X,Y
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
```

```
*           intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2IXY ( DATE1, DATE2, X, Y, RC2I )
*+
*  - - - - - - - - - -
*   i a u _ C 2 I X Y
*  - - - - - - - - - -
*
*  Form the celestial to intermediate-frame-of-date matrix for a given
*  date when the CIP X,Y coordinates are known.  IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     X,Y            d      Celestial Intermediate Pole (Note 2)
*
*  Returned:
*     RC2I         d(3,3)   celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  4) Although its name does not include "00", this routine is in fact
*     specific to the IAU 2000 models.
*
*  Called:
*     iau_C2IXYS   celestial-to-intermediate matrix, given X,Y and s
*     iau_S00      the CIO locator s, given X,Y, IAU 2000A
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2IXYS ( X, Y, S, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I X Y S
*  - - - - - - - - - - -
*
*  Form the celestial to intermediate-frame-of-date matrix given the CIP
*  X,Y and the CIO locator s.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     X,Y        d       Celestial Intermediate Pole (Note 1)
*     S          d       the CIO locator s (Note 2)
*
*  Returned:
*     RC2I     d(3,3)    celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  2) The CIO locator s (in radians) positions the Celestial
*     Intermediate Origin on the equator of the CIP.
*
*  3) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RY       rotate around Y-axis
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2S ( P, THETA, PHI )
*+
*  - - - - - - - -
*   i a u _ C 2 S
*  - - - - - - - -
*
*  P-vector to spherical coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)      p-vector
*
*  Returned:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*
*  Notes:
*
*  1) P can have any magnitude; only its direction is used.
*
*  2) If P is null, zero THETA and PHI are returned.
*
*  3) At either pole, zero THETA is returned.
*
*-
```

```
      SUBROUTINE iau_C2T00A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T 0 0 A
*  - - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1 and
*  the polar motion, using the IAU 2000A nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 3)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth rotation angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  3) The matrix RC2T transforms from celestial to terrestrial
*     coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), RC2I is the
*     celestial-to-intermediate matrix, ERA is the Earth rotation angle
*     and RPOM is the polar motion matrix.
*
*  4) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_C2T00B routine.
*
*  Called:
*     iau_C2I00A   celestial-to-intermediate matrix, IAU 2000A
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_POM00    polar motion matrix
```

```
*     iau_C2TCIO   form CIO−based celestial−to−terrestrial matrix
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*−
```

```
      SUBROUTINE iau_C2T00B ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T 0 0 B
*  - - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1 and
*  the polar motion, using the IAU 2000B nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 3)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth rotation angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  3) The matrix RC2T transforms from celestial to terrestrial
*     coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), RC2I is the
*     celestial-to-intermediate matrix, ERA is the Earth rotation angle
*     and RPOM is the polar motion matrix.
*
*  4) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_C2T00A routine.
*
*  Called:
*     iau_C2I00B   celestial-to-intermediate matrix, IAU 2000B
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_POM00    polar motion matrix
*     iau_C2TCIO   form CIO-based celestial-to-terrestrial matrix
```

```
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2T06A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T 0 6 A
*  - - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1 and
*  the polar motion, using the IAU 2006 precession and IAU 2000A
*  nutation models.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 3)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth rotation angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  3) The matrix RC2T transforms from celestial to terrestrial
*     coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), RC2I is the
*     celestial-to-intermediate matrix, ERA is the Earth rotation angle
*     and RPOM is the polar motion matrix.
*
*  Called:
*     iau_C2I06A   celestial-to-intermediate matrix, IAU 2006/2000A
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_POM00    polar motion matrix
*     iau_C2TCIO   form CIO-based celestial-to-terrestrial matrix
*
```

```
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*-
```

```
      SUBROUTINE iau_C2TCEO ( RC2I, ERA, RPOM, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T C E O
*  - - - - - - - - - - -
*
*  Assemble the celestial to terrestrial matrix from CIO-based
*  components (the celestial-to-intermediate matrix, the Earth Rotation
*  Angle and the polar motion matrix).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  obsolete routine.
*
*  Given:
*     RC2I     d(3,3)    celestial-to-intermediate matrix
*     ERA        d       Earth rotation angle
*     RPOM     d(3,3)    polar-motion matrix
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix
*
*  Notes:
*
*  1) The name of the present routine, iau_C2TCEO, reflects the original
*     name of the celestial intermediate origin (CIO), which before the
*     adoption of IAU 2006 Resolution 2 was called the "celestial
*     ephemeris origin" (CEO).
*
*  2) When the name change from CEO to CIO occurred, a new SOFA routine
*     called iau_C2TCIO was introduced as the successor to the existing
*     iau_C2TCEO.  The present routine is merely a front end to the new
*     one.
*
*  3) The present routine is included in the SOFA collection only to
*     support existing applications.  It should not be used in new
*     applications.
*
*  Called:
*     iau_C2TCIO   form CIO-based celestial-to-terrestrial matrix
*
*-
```

```
      SUBROUTINE iau_C2TCIO ( RC2I, ERA, RPOM, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T C I O
*  - - - - - - - - - - -
*
*  Assemble the celestial to terrestrial matrix from CIO-based
*  components (the celestial-to-intermediate matrix, the Earth Rotation
*  Angle and the polar motion matrix).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC2I     d(3,3)    celestial-to-intermediate matrix
*     ERA        d       Earth rotation angle (radians)
*     RPOM     d(3,3)    polar-motion matrix
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix
*
*  Notes:
*
*  1) This routine constructs the rotation matrix that transforms
*     vectors in the celestial system into vectors in the terrestrial
*     system.  It does so starting from precomputed components, namely
*     the matrix which rotates from celestial coordinates to the
*     intermediate frame, the Earth rotation angle and the polar motion
*     matrix.  One use of the present routine is when generating a
*     series of celestial-to-terrestrial matrices where only the Earth
*     Rotation Angle changes, avoiding the considerable overhead of
*     recomputing the precession-nutation more often than necessary to
*     achieve given accuracy objectives.
*
*  2) The relationship between the arguments is as follows:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003).
*
*  Called:
*     iau_CR       copy r-matrix
*     iau_RZ       rotate around Z-axis
*     iau_RXR      product of two r-matrices
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*-
```

```
      SUBROUTINE iau_C2TEQX ( RBPN, GST, RPOM, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T E Q X
*  - - - - - - - - - - -
*
*  Assemble the celestial to terrestrial matrix from equinox-based
*  components (the celestial-to-true matrix, the Greenwich Apparent
*  Sidereal Time and the polar motion matrix).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RBPN     d(3,3)    celestial-to-true matrix
*     GST        d       Greenwich (apparent) Sidereal Time (radians)
*     RPOM     d(3,3)    polar-motion matrix
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 2)
*
*  Notes:
*
*  1) This routine constructs the rotation matrix that transforms
*     vectors in the celestial system into vectors in the terrestrial
*     system.  It does so starting from precomputed components, namely
*     the matrix which rotates from celestial coordinates to the
*     true equator and equinox of date, the Greenwich Apparent Sidereal
*     Time and the polar motion matrix.  One use of the present routine
*     is when generating a series of celestial-to-terrestrial matrices
*     where only the Sidereal Time changes, avoiding the considerable
*     overhead of recomputing the precession-nutation more often than
*     necessary to achieve given accuracy objectives.
*
*  2) The relationship between the arguments is as follows:
*
*        [TRS]  =  RPOM * R_3(GST) * RBPN * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003).
*
*  Called:
*     iau_CR       copy r-matrix
*     iau_RZ       rotate around Z-axis
*     iau_RXR      product of two r-matrices
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2TPE ( TTA, TTB, UTA, UTB, DPSI, DEPS, XP, YP,
     :                       RC2T )
*+
*  - - - - - - - - - -
*   i a u _ C 2 T P E
*  - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1, the
*  nutation and the polar motion.  IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB     d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB     d       UT1 as a 2-part Julian Date (Note 1)
*     DPSI,DEPS   d       nutation (Note 2)
*     XP,YP       d       coordinates of the pole (radians, Note 3)
*
*  Returned:
*     RC2T      d(3,3)    celestial-to-terrestrial matrix (Note 4)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth rotation angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) The caller is responsible for providing the nutation components;
*     they are in longitude and obliquity, in radians and are with
*     respect to the equinox and ecliptic of date.  For high-accuracy
*     applications, free core nutation should be included as well as
*     any other relevant corrections to the position of the CIP.
*
*  3) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  4) The matrix RC2T transforms from celestial to terrestrial
*     coordinates:
*
*        [TRS]  =  RPOM * R_3(GST) * RBPN * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), RBPN is the
*     bias-precession-nutation matrix, GST is the Greenwich (apparent)
*     Sidereal Time and RPOM is the polar motion matrix.
*
```

```
*  5) Although its name does not include "00", this routine is in fact
*     specific to the IAU 2000 models.
*
*  Called:
*     iau_PN00     bias/precession/nutation results, IAU 2000
*     iau_GMST00   Greenwich mean sidereal time, IAU 2000
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_EE00     equation of the equinoxes, IAU 2000
*     iau_POM00    polar motion matrix
*     iau_C2TEQX   form equinox-based celestial-to-terrestrial matrix
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_C2TXY ( TTA, TTB, UTA, UTB, X, Y, XP, YP, RC2T )
*+
*  - - - - - - - - - -
*   i a u _ C 2 T X Y
*  - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1, the
*  CIP coordinates and the polar motion.  IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     X,Y        d       Celestial Intermediate Pole (Note 2)
*     XP,YP      d       coordinates of the pole (radians, Note 3)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 4)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth rotation angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  4) The matrix RC2T transforms from celestial to terrestrial
*     coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2003), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  5) Although its name does not include "00", this routine is in fact
*     specific to the IAU 2000 models.
*
*  Called:
*     iau_C2IXY    celestial-to-intermediate matrix, given X,Y
```

```
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_SP00     the TIO locator s', IERS 2000
*     iau_POM00    polar motion matrix
*     iau_C2TCIO   form CIO-based celestial-to-terrestrial matrix
*
* Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_CAL2JD ( IY, IM, ID, DJM0, DJM, J )
*+
*  - - - - - - - - - - -
*   i a u _ C A L 2 J D
*  - - - - - - - - - - -
*
*  Gregorian Calendar to Julian Date.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     IY,IM,ID    i     year, month, day in Gregorian calendar (Note 1)
*
*  Returned:
*     DJM0        d     MJD zero-point: always 2400000.5
*     DJM         d     Modified Julian Date for 0 hrs
*     J           i     status:
*                           0 = OK
*                          -1 = bad year   (Note 3: JD not computed)
*                          -2 = bad month  (JD not computed)
*                          -3 = bad day    (JD computed)
*
*  Notes:
*
*  1) The algorithm used is valid from -4800 March 1, but this
*     implementation rejects dates before -4799 January 1.
*
*  2) The Julian Date is returned in two pieces, in the usual SOFA
*     manner, which is designed to preserve time resolution.  The
*     Julian Date is available as a single number by adding DJM0 and
*     DJM.
*
*  3) In early eras the conversion is from the "Proleptic Gregorian
*     Calendar";  no account is taken of the date(s) of adoption of
*     the Gregorian Calendar, nor is the AD/BC numbering convention
*     observed.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 12.92 (p604).
*
*-
```

```
      SUBROUTINE iau_CP ( P, C )
*+
*  - - - - - - - 
*   i a u _ C P
*  - - - - - - - 
*
*  Copy a p-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)     p-vector to be copied
*
*  Returned:
*     C        d(3)     copy
*
*-
```

```
      SUBROUTINE iau_CPV ( PV, C )
*+
*  - - - - - - - -
*   i a u _ C P V
*  - - - - - - - -
*
*  Copy a position/velocity vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PV       d(3,2)    position/velocity vector to be copied
*
*  Returned:
*     C        d(3,2)    copy
*
*  Called:
*     iau_CP       copy p-vector
*
*-
```

```
      SUBROUTINE iau_CR ( R, C )
*+
*  - - - - - - -
*   i a u _ C R
*  - - - - - - -
*
*  Copy an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix to be copied
*
*  Returned:
*     C        d(3,3)    copy
*
*  Called:
*     iau_CP       copy p-vector
*
*-
```

```
      SUBROUTINE iau_D2DTF ( SCALE, NDP, D1, D2, IY, IM, ID, IHMSF, J )
*+
*  − − − − − − − − − −
*   i a u _ D 2 D T F
*  − − − − − − − − − −
*
*  Format for output a 2−part Julian Date (or in the case of UTC a
*  quasi−JD form that includes special provision for leap seconds).
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     SCALE      c*(*)  time scale ID (Note 1)
*     NDP        i      resolution (Note 2)
*     D1,D2      d      time as a 2−part Julian Date (Notes 3,4)
*
*  Returned:
*     IY,IM,ID   i      year, month, day in Gregorian calendar (Note 5)
*     IHMSF      i(4)   hours, minutes, seconds, fraction (Note 1)
*     J          i      status: +1 = dubious year (Note 5)
*                                0 = OK
*                               −1 = unacceptable date (Note 6)
*
*  Notes:
*
*  1) SCALE identifies the time scale.  Only the value ’UTC’ (in upper
*     case) is significant, and enables handling of leap seconds (see
*     Note 4).
*
*  2) NDP is the number of decimal places in the seconds field, and can
*     have negative as well as positive values, such as:
*
*     NDP         resolution
*     −4            1 00 00
*     −3            0 10 00
*     −2            0 01 00
*     −1            0 00 10
*      0            0 00 01
*      1            0 00 00.1
*      2            0 00 00.01
*      3            0 00 00.001
*
*     The limits are platform dependent, but a safe range is −5 to +9.
*
*  3) D1+D2 is Julian Date, apportioned in any convenient way between
*     the two arguments, for example where D1 is the Julian Day Number
*     and D2 is the fraction of a day.  In the case of UTC, where the
*     use of JD is problematical, special conventions apply:  see the
*     next note.
*
*  4) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The SOFA internal convention is that
*     the quasi−JD day represents UTC days whether the length is 86399,
*     86400 or 86401 SI seconds.  In the 1960−1972 era there were
*     smaller jumps (in either direction) each time the linear UTC(TAI)
*     expression was changed, and these "mini−leaps" are also included
*     in the SOFA convention.
*
*  5) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  6) For calendar conventions and limitations, see iau_CAL2JD.
*
*  Called:
*     iau_JD2CAL   JD to Gregorian calendar
*     iau_D2TF     decompose days to hms
*     iau_DAT      delta(AT) = TAI−UTC
*
```

*_

```
      SUBROUTINE iau_D2TF ( NDP, DAYS, SIGN, IHMSF )
*+
*  - - - - - - - - -
*   i a u _ D 2 T F
*  - - - - - - - - -
*
*  Decompose days to hours, minutes, seconds, fraction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     NDP       i        resolution (Note 1)
*     DAYS      d        interval in days
*
*  Returned:
*     SIGN      c        '+' or '-'
*     IHMSF     i(4)     hours, minutes, seconds, fraction
*
*  Notes:
*
*  1) NDP is interpreted as follows:
*
*     NDP         resolution
*      :      ...0000 00 00
*     -7         1000 00 00
*     -6          100 00 00
*     -5           10 00 00
*     -4            1 00 00
*     -3            0 10 00
*     -2            0 01 00
*     -1            0 00 10
*      0            0 00 01
*      1            0 00 00.1
*      2            0 00 00.01
*      3            0 00 00.001
*      :            0 00 00.000...
*
*  2) The largest positive useful value for NDP is determined by the
*     size of DAYS, the format of DOUBLE PRECISION floating-point
*     numbers on the target platform, and the risk of overflowing
*     IHMSF(4).  On a typical platform, for DAYS up to 1D0, the
*     available floating-point precision might correspond to NDP=12.
*     However, the practical limit is typically NDP=9, set by the
*     capacity of a 32-bit IHMSF(4).
*
*  3) The absolute value of DAYS may exceed 1D0.  In cases where it
*     does not, it is up to the caller to test for and handle the
*     case where DAYS is very nearly 1D0 and rounds up to 24 hours,
*     by testing for IHMSF(1)=24 and setting IHMSF(1-4) to zero.
*
*-
```

```
      SUBROUTINE iau_DAT ( IY, IM, ID, FD, DELTAT, J )
*+
*  - - - - - - - -
*   i a u _ D A T
*  - - - - - - - -
*
*  For a given UTC date, calculate Delta(AT) = TAI-UTC.
*
*     :------------------------------------------:
*     :                                          :
*     :                 IMPORTANT                :
*     :                                          :
*     :  A new version of this routine must be   :
*     :  produced whenever a new leap second is  :
*     :  announced.  There are five items to     :
*     :  change on each such occasion:           :
*     :                                          :
*     :  1) The parameter NDAT must be           :
*     :     increased by 1.                      :
*     :                                          :
*     :  2) The set of DATA statements that      :
*     :     initialize the arrays IDAT and       :
*     :     DATS must be extended by one line.   :
*     :                                          :
*     :  3) The parameter IYV must be set to     :
*     :     the current year.                    :
*     :                                          :
*     :  4) The "Latest leap second" comment     :
*     :     below must be set to the new leap    :
*     :     second date.                         :
*     :                                          :
*     :  5) The "This revision" comment, later,  :
*     :     must be set to the current date.     :
*     :                                          :
*     :  Change (3) must also be carried out     :
*     :  whenever the routine is re-issued,      :
*     :  even if no leap seconds have been       :
*     :  added.                                  :
*     :                                          :
*     :  Latest leap second:  2016 December 31   :
*     :                                          :
*     :__________________________________________:
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  user-replaceable support routine.
*
*  Given:
*     IY       i     UTC:  year (Notes 1 and 2)
*     IM       i           month (Note 2)
*     ID       i           day (Notes 2 and 3)
*     FD       d           fraction of day (Note 4)
*
*  Returned:
*     DELTAT   d     TAI minus UTC, seconds
*     J        i     status (Note 5):
*                       1 = dubious year (Note 1)
*                       0 = OK
*                      -1 = bad year
*                      -2 = bad month
*                      -3 = bad day (Note 3)
*                      -4 = bad fraction (Note 4)
*                      -5 = internal error (Note 5)
*
*  Notes:
*
*  1) UTC began at 1960 January 1.0 (JD 2436934.5) and it is improper
*     to call the routine with an earlier date.  If this is attempted,
*     zero is returned together with a warning status.
*
*     Because leap seconds cannot, in principle, be predicted in
```

```
*     advance, a reliable check for dates beyond the valid range is
*     impossible.  To guard against gross errors, a year five or more
*     after the release year of the present routine (see parameter IYV)
*     is considered dubious.  In this case a warning status is returned
*     but the result is computed in the normal way.
*
*     For both too-early and too-late years, the warning status is J=+1.
*     This is distinct from the error status J=-1, which signifies a
*     year so early that JD could not be computed.
*
*  2) If the specified date is for a day which ends with a leap second,
*     the TAI-UTC value returned is for the period leading up to the
*     leap second.  If the date is for a day which begins as a leap
*     second ends, the TAI-UTC returned is for the period following the
*     leap second.
*
*  3) The day number must be in the normal calendar range, for example
*     1 through 30 for April.  The "almanac" convention of allowing
*     such dates as January 0 and December 32 is not supported in this
*     routine, in order to avoid confusion near leap seconds.
*
*  4) The fraction of day is used only for dates before the introduction
*     of leap seconds, the first of which occurred at the end of 1971.
*     It is tested for validity (0 to 1 is the valid range) even if not
*     used;  if invalid, zero is used and status J=-4 is returned.  For
*     many applications, setting FD to zero is acceptable;  the
*     resulting error is always less than 3 ms (and occurs only
*     pre-1972).
*
*  5) The status value returned in the case where there are multiple
*     errors refers to the first error detected.  For example, if the
*     month and day are 13 and 32 respectively, J=-2 (bad month) will be
*     returned.  The "internal error" status refers to a case that is
*     impossible but causes some compilers to issue a warning.
*
*  6) In cases where a valid result is not available, zero is returned.
*
*  References:
*
*  1) For dates from 1961 January 1 onwards, the expressions from the
*     file ftp://maia.usno.navy.mil/ser7/tai-utc.dat are used.
*
*  2) The 5ms timestep at 1961 January 1 is taken from 2.58.1 (p87) of
*     the 1992 Explanatory Supplement.
*
*  Called:
*     iau_CAL2JD   Gregorian calendar to JD
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_DTDB ( DATE1, DATE2,
     :                                     UT, ELONG, U, V )
*+
*  - - - - - - - - -
*   i a u _ D T D B
*  - - - - - - - - -
*
*  An approximation to TDB-TT, the difference between barycentric
*  dynamical time and terrestrial time, for an observer on the Earth.
*
*  The different time scales - proper, coordinate and realized - are
*  related to each other:
*
*            TAI             <-  physically realized
*             :
*          offset            <-  observed (nominally +32.184s)
*             :
*            TT              <-  terrestrial time
*             :
*    rate adjustment (L_G)   <-  definition of TT
*             :
*            TCG             <-  time scale for GCRS
*             :
*      "periodic" terms      <-  iau_DTDB is an implementation
*             :
*    rate adjustment (L_C)   <-  function of solar-system ephemeris
*             :
*            TCB             <-  time scale for BCRS
*             :
*    rate adjustment (-L_B)  <-  definition of TDB
*             :
*            TDB             <-  TCB scaled to track TT
*             :
*      "periodic" terms      <-  -iau_DTDB is an approximation
*             :
*            TT              <-  terrestrial time
*
*  Adopted values for the various constants can be found in the IERS
*  Conventions (McCarthy & Petit 2003).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2     d    date, TDB (Notes 1-3)
*     UT              d    universal time (UT1, fraction of one day)
*     ELONG           d    longitude (east positive, radians)
*     U               d    distance from Earth spin axis (km)
*     V               d    distance north of equatorial plane (km)
*
*  Returned:
*    iau_DTDB         d    TDB-TT (seconds)
*
*  Notes:
*
*  1) The date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the arguments DATE1 and DATE2.  For
*     example, JD(TDB)=2450123.7 could be expressed in any of these
*     ways, among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
```

```
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.
*
*     Although the date is, formally, barycentric dynamical time (TDB),
*     the terrestrial dynamical time (TT) can be used with no practical
*     effect on the accuracy of the prediction.
*
*  2) TT can be regarded as a coordinate time that is realized as an
*     offset of 32.184s from International Atomic Time, TAI.  TT is a
*     specific linear transformation of geocentric coordinate time TCG,
*     which is the time scale for the Geocentric Celestial Reference
*     System, GCRS.
*
*  3) TDB is a coordinate time, and is a specific linear transformation
*     of barycentric coordinate time TCB, which is the time scale for
*     the Barycentric Celestial Reference System, BCRS.
*
*  4) The difference TCG−TCB depends on the masses and positions of the
*     bodies of the solar system and the velocity of the Earth.  It is
*     dominated by a rate difference, the residual being of a periodic
*     character.  The latter, which is modeled by the present routine,
*     comprises a main (annual) sinusoidal term of amplitude
*     approximately 0.00166 seconds, plus planetary terms up to about
*     20 microseconds, and lunar and diurnal terms up to 2 microseconds.
*     These effects come from the changing transverse Doppler effect
*     and gravitational red−shift as the observer (on the Earth’s
*     surface) experiences variations in speed (with respect to the
*     BCRS) and gravitational potential.
*
*  5) TDB can be regarded as the same as TCB but with a rate adjustment
*     to keep it close to TT, which is convenient for many applications.
*     The history of successive attempts to define TDB is set out in
*     Resolution 3 adopted by the IAU General Assembly in 2006, which
*     defines a fixed TDB(TCB) transformation that is consistent with
*     contemporary solar−system ephemerides.  Future ephemerides will
*     imply slightly changed transformations between TCG and TCB, which
*     could introduce a linear drift between TDB and TT;  however, any
*     such drift is unlikely to exceed 1 nanosecond per century.
*
*  6) The geocentric TDB−TT model used in the present routine is that of
*     Fairhead & Bretagnon (1990), in its full form.  It was originally
*     supplied by Fairhead (private communications with P.T.Wallace,
*     1990) as a Fortran subroutine.  The present routine contains an
*     adaptation of the Fairhead code.  The numerical results are
*     essentially unaffected by the changes, the differences with
*     respect to the Fairhead & Bretagnon original being at the 1D−20 s
*     level.
*
*     The topocentric part of the model is from Moyer (1981) and
*     Murray (1983), with fundamental arguments adapted from
*     Simon et al. 1994.  It is an approximation to the expression
*     ( v / c ) . ( r / c ), where v is the barycentric velocity of
*     the Earth, r is the geocentric position of the observer and
*     c is the speed of light.
*
*     By supplying zeroes for U and V, the topocentric part of the
*     model can be nullified, and the routine will return the Fairhead
*     & Bretagnon result alone.
*
*  7) During the interval 1950−2050, the absolute accuracy is better
*     than +/− 3 nanoseconds relative to time ephemerides obtained by
*     direct numerical integrations based on the JPL DE405 solar system
*     ephemeris.
*
*  8) It must be stressed that the present routine is merely a model,
*     and that numerical integration of solar−system ephemerides is the
*     definitive method for predicting the relationship between TCG and
*     TCB and hence between TT and TDB.
*
*  References:
*
*     Fairhead, L., & Bretagnon, P., Astron.Astrophys., 229, 240−247
```

```
*     (1990).
*
*     IAU 2006 Resolution 3.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Moyer, T.D., Cel.Mech., 23, 33 (1981).
*
*     Murray, C.A., Vectorial Astrometry, Adam Hilger (1983).
*
*     Seidelmann, P.K. et al., Explanatory Supplement to the
*     Astronomical Almanac, Chapter 2, University Science Books (1992).
*
*     Simon, J.L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G. & Laskar, J., Astron.Astrophys., 282, 663-683 (1994).
*
*-
```

```
      SUBROUTINE iau_DTF2D ( SCALE, IY, IM, ID, IHR, IMN, SEC,
     :                       D1, D2, J )
*+
*  - - - - - - - - - - -
*   i a u _ D T F 2 D
*  - - - - - - - - - - -
*
*  Encode date and time fields into 2-part Julian Date (or in the case
*  of UTC a quasi-JD form that includes special provision for leap
*  seconds).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     SCALE      c*(*)  time scale ID (Note 1)
*     IY,IM,ID   i      year, month, day in Gregorian calendar (Note 2)
*     IHR,IMN    i      hour, minute
*     SEC        d      seconds
*
*  Returned:
*     D1,D2      d      2-part Julian Date (Notes 3,4)
*     J          i      status: +3 = both of next two
*                               +2 = time is after end of day (Note 5)
*                               +1 = dubious year (Note 6)
*                                0 = OK
*                               -1 = bad year
*                               -2 = bad month
*                               -3 = bad day
*                               -4 = bad hour
*                               -5 = bad minute
*                               -6 = bad second (<0)
*
*  Notes:
*
*  1) SCALE identifies the time scale.  Only the value 'UTC' (in upper
*     case) is significant, and enables handling of leap seconds (see
*     Note 4).
*
*  2) For calendar conventions and limitations, see iau_CAL2JD.
*
*  3) The sum of the results, D1+D2, is Julian Date, where normally D1
*     is the Julian Day Number and D2 is the fraction of a day.  In the
*     case of UTC, where the use of JD is problematical, special
*     conventions apply:  see the next note.
*
*  4) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The SOFA internal convention is that
*     the quasi-JD day represents UTC days whether the length is 86399,
*     86400 or 86401 SI seconds.  In the 1960-1972 era there were
*     smaller jumps (in either direction) each time the linear UTC(TAI)
*     expression was changed, and these "mini-leaps" are also included
*     in the SOFA convention.
*
*  5) The warning status "time is after end of day" usually means that
*     the SEC argument is greater than 60D0.  However, in a day ending
*     in a leap second the limit changes to 61D0 (or 59D0 in the case of
*     a negative leap second).
*
*  6) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  7) Only in the case of continuous and regular time scales (TAI, TT,
*     TCG, TCB and TDB) is the result D1+D2 a Julian Date, strictly
*     speaking.  In the other cases (UT1 and UTC) the result must be
*     used with circumspection;  in particular the difference between
*     two such results cannot be interpreted as a precise time
*     interval.
*
```

```
*  Called:
*     iau_CAL2JD   Gregorian calendar to JD
*     iau_DAT      delta(AT) = TAI-UTC
*     iau_JD2CAL   JD to Gregorian calendar
*
*-
```

```
      SUBROUTINE iau_ECEQ06 ( DATE1, DATE2, DL, DB, DR, DD )
*+
*  - - - - - - - - - - -
*   i a u _ E C E Q 0 6
*  - - - - - - - - - - -
*
*  Transformation from ecliptic coordinates (mean equinox and ecliptic
*  of date) to ICRS RA,Dec, using the IAU 2006 precession model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2  d      TT as a 2-part Julian Date (Note 1)
*     DL,DB        d      ecliptic longitude and latitude (radians)
*
*  Returned:
*     DR,DD        d      ICRS right ascension and declination (radians)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) No assumptions are made about whether the coordinates represent
*     starlight and embody astrometric effects such as parallax or
*     aberration.
*
*  3) The transformation is approximately that from ecliptic longitude
*     and latitude (mean equinox and ecliptic of date) to mean J2000.0
*     right ascension and declination, with only frame bias (always less
*     than 25 mas) to disturb this classical picture.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_ECM06    J2000.0 to ecliptic rotation matrix, IAU 2006
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_C2S      unit vector to spherical coordinates
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*
*-
```

```
      SUBROUTINE iau_ECM06 ( DATE1, DATE2, RM )
*+
*  - - - - - - - - - -
*   i a u _ E C M 0 6
*  - - - - - - - - - -
*
*  ICRS equatorial to ecliptic rotation matrix, IAU 2006.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2  d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RM           d(3,3) ICRS to ecliptic rotation matrix
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix is in the sense
*
*        E_ep = RM x P_ICRS,
*
*     where P_ICRS is a vector with respect to ICRS right ascension
*     and declination axes and E_ep is the same vector with respect to
*     the (inertial) ecliptic and equinox of date.
*
*  3) P_ICRS is a free vector, merely a direction, typically of unit
*     magnitude, and not bound to any particular spatial origin, such as
*     the Earth, Sun or SSB.  No assumptions are made about whether it
*     represents starlight and embodies astrometric effects such as
*     parallax or aberration.  The transformation is approximately that
*     between mean J2000.0 right ascension and declination and ecliptic
*     longitude and latitude, with only frame bias (always less than
*     25 mas) to disturb this classical picture.
*
*  Called:
*     iau_OBL06    mean obliquity, IAU 2006
*     iau_PMAT06   PB matrix, IAU 2006
*     iau_IR       initialize r-matrix to identity
*     iau_RX       rotate around X-axis
*     iau_RXR      product of two r-matrices
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE00 ( DATE1, DATE2, EPSA, DPSI )
*+
*  - - - - - - - - -
*   i a u _ E E 0 0
*  - - - - - - - - -
*
*  The equation of the equinoxes, compatible with IAU 2000 resolutions,
*  given the nutation in longitude and the mean obliquity.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     EPSA           d      mean obliquity (Note 2)
*     DPSI           d      nutation in longitude (Note 3)
*
*  Returned:
*     iau_EE00       d      equation of the equinoxes (Note 4)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The obliquity, in radians, is mean of date.
*
*  3) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  4) The result is compatible with the IAU 2000 resolutions.  For
*     further details, see IERS Conventions 2003 and Capitaine et al.
*     (2002).
*
*  Called:
*     iau_EECT00   equation of the equinoxes complementary terms
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE00A ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ E E 0 0 A
*  - - - - - - - - - -
*
*  Equation of the equinoxes, compatible with IAU 2000 resolutions.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EE00A      d      equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  3) The result is compatible with the IAU 2000 resolutions.  For
*     further details, see IERS Conventions 2003 and Capitaine et al.
*     (2002).
*
*  Called:
*     iau_PR00     IAU 2000 precession adjustments
*     iau_OBL80    mean obliquity, IAU 1980
*     iau_NUT00A   nutation, IAU 2000A
*     iau_EE00     equation of the equinoxes, IAU 2000
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE00B ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ E E 0 0 B
*  - - - - - - - - - -
*
*  Equation of the equinoxes, compatible with IAU 2000 resolutions but
*  using the truncated nutation model IAU 2000B.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d     TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EE00B     d     equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  3) The result is compatible with the IAU 2000 resolutions except that
*     accuracy has been compromised for the sake of speed.  For further
*     details, see McCarthy & Luzum (2001), IERS Conventions 2003 and
*     Capitaine et al. (2003).
*
*  Called:
*     iau_PR00     IAU 2000 precession adjustments
*     iau_OBL80    mean obliquity, IAU 1980
*     iau_NUT00B   nutation, IAU 2000B
*     iau_EE00     equation of the equinoxes, IAU 2000
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D.D. & Luzum, B.J., "An abridged model of the
*     precession-nutation of the celestial pole", Celestial Mechanics &
*     Dynamical Astronomy, 85, 37-49 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE06A ( DATE1, DATE2 )
*+
*  - - - - - - - - - - -
*   i a u _ E E 0 6 A
*  - - - - - - - - - - -
*
*  Equation of the equinoxes, compatible with IAU 2000 resolutions and
*  IAU 2006/2000A precession-nutation.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EE06A      d      equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  Called:
*     iau_ANPM     normalize angle into range +/- pi
*     iau_GST06A   Greenwich apparent sidereal time, IAU 2006/2000A
*     iau_GMST06   Greenwich mean sidereal time, IAU 2006
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EECT00 ( DATE1, DATE2 )
*+
*  - - - - - - - - - - -
*   i a u _ E E C T 0 0
*  - - - - - - - - - - -
*
*  Equation of the equinoxes complementary terms, consistent with
*  IAU 2000 resolutions.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EECT00    d    complementary terms (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The "complementary terms" are part of the equation of the
*     equinoxes (EE), classically the difference between apparent and
*     mean Sidereal Time:
*
*        GAST = GMST + EE
*
*     with:
*
*        EE = dpsi * cos(eps)
*
*     where dpsi is the nutation in longitude and eps is the obliquity
*     of date.  However, if the rotation of the Earth were constant in
*     an inertial frame the classical formulation would lead to apparent
*     irregularities in the UT1 timescale traceable to side-effects of
*     precession-nutation.  In order to eliminate these effects from
*     UT1, "complementary terms" were introduced in 1994 (IAU, 1994) and
*     took effect from 1997 (Capitaine and Gontier, 1993):
*
*        GAST = GMST + CT + EE
*
*     By convention, the complementary terms are included as part of the
*     equation of the equinoxes rather than as part of the mean Sidereal
*     Time.  This slightly compromises the "geometrical" interpretation
*     of mean sidereal time but is otherwise inconsequential.
*
*     The present routine computes CT in the above expression,
*     compatible with IAU 2000 resolutions (Capitaine et al., 2002, and
*     IERS Conventions 2003).
*
*  Called:
*     iau_FAL03    mean anomaly of the Moon
```

```
*     iau_FALP03   mean anomaly of the Sun
*     iau_FAF03    mean argument of the latitude of the Moon
*     iau_FAD03    mean elongation of the Moon from the Sun
*     iau_FAOM03   mean longitude of the Moon’s ascending node
*     iau_FAVE03   mean longitude of Venus
*     iau_FAE03    mean longitude of Earth
*     iau_FAPA03   general accumulated precession in longitude
*
*  References:
*
*     Capitaine, N. & Gontier, A.−M., Astron.Astrophys., 275,
*     645−650 (1993)
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astron.Astrophys.,
*     406, 1135−1149 (2003)
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*−
```

```
      SUBROUTINE iau_EFORM ( N, A, F, J )
*+
*  - - - - - - - - - -
*   i a u _ E F O R M
*  - - - - - - - - - -
*
*  Earth reference ellipsoids.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     N         i     ellipsoid identifier (Note 1)
*
*  Returned:
*     A         d     equatorial radius (meters, Note 2)
*     F         d     flattening (Note 2)
*     J         i     status:  0 = OK
*                             -1 = illegal identifier (Note 3)
*
*  Notes:
*
*  1) The identifier N is a number that specifies the choice of
*     reference ellipsoid.  The following are supported:
*
*        N   ellipsoid
*
*        1    WGS84
*        2    GRS80
*        3    WGS72
*
*     The number N has no significance outside the SOFA software.
*
*  2) The ellipsoid parameters are returned in the form of equatorial
*     radius in meters (A) and flattening (F).  The latter is a number
*     around 0.00335, i.e. around 1/298.
*
*  3) For the case where an unsupported N value is supplied, zero A and
*     F are returned, as well as error status.
*
*  References:
*
*     Department of Defense World Geodetic System 1984, National Imagery
*     and Mapping Agency Technical Report 8350.2, Third Edition, p3-2.
*
*     Moritz, H., Bull. Geodesique 66-2, 187 (1992).
*
*     The Department of Defense World Geodetic System 1972, World
*     Geodetic System Committee, May 1974.
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     p220.
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EO06A ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ E O 0 6 A
*  - - - - - - - - - -
*
*  Equation of the origins, IAU 2006 precession and IAU 2000A nutation.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EO06A      d      equation of the origins in radians
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The equation of the origins is the distance between the true
*     equinox and the celestial intermediate origin and, equivalently,
*     the difference between Earth rotation angle and Greenwich
*     apparent sidereal time (ERA-GST).  It comprises the precession
*     (since J2000.0) in right ascension plus the equation of the
*     equinoxes (including the small correction terms).
*
*  Called:
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*     iau_EORS     equation of the origins, given NPB matrix and s
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EORS ( RNPB, S )
*+
*  - - - - - - - - -
*   i a u _ E O R S
*  - - - - - - - - -
*
*  Equation of the origins, given the classical NPB matrix and the
*  quantity s.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RNPB    d(3,3)    classical nutation x precession x bias matrix
*     S         d       the quantity s (the CIO locator)
*
*  Returned:
*     iau_EORS  d       the equation of the origins in radians.
*
*  Notes:
*
*  1)  The equation of the origins is the distance between the true
*      equinox and the celestial intermediate origin and, equivalently,
*      the difference between Earth rotation angle and Greenwich
*      apparent sidereal time (ERA-GST).  It comprises the precession
*      (since J2000.0) in right ascension plus the equation of the
*      equinoxes (including the small correction terms).
*
*  2)  The algorithm is from Wallace & Capitaine (2006).
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EPB ( DJ1, DJ2 )
*+
*  - - - - - - - -
*   i a u _ E P B
*  - - - - - - - -
*
*  Julian Date to Besselian Epoch.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DJ1,DJ2   d         Julian Date (see note)
*
*  The result is the Besselian Epoch.
*
*  Note:
*
*     The Julian Date is supplied in two pieces, in the usual SOFA
*     manner, which is designed to preserve time resolution.  The
*     Julian Date is available as a single number by adding DJ1 and
*     DJ2.  The maximum resolution is achieved if DJ1 is 2451545D0
*     (J2000.0).
*
*  Reference:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*
*-
```

```
      SUBROUTINE iau_EPB2JD ( EPB, DJM0, DJM )
*+
*  - - - - - - - - - - -
*   i a u _ E P B 2 J D
*  - - - - - - - - - - -
*
*  Besselian Epoch to Julian Date.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPB         d     Besselian Epoch (e.g. 1957.3D0)
*
*  Returned:
*     DJM0        d     MJD zero-point: always 2400000.5
*     DJM         d     Modified Julian Date
*
*  Note:
*
*     The Julian Date is returned in two pieces, in the usual SOFA
*     manner, which is designed to preserve time resolution.  The
*     Julian Date is available as a single number by adding DJM0 and
*     DJM.
*
*  Reference:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EPJ ( DJ1, DJ2 )
*+
*  - - - - - - - -
*   i a u _ E P J
*  - - - - - - - -
*
*  Julian Date to Julian Epoch.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DJ1,DJ2   d         Julian Date (see note)
*
*  The result is the Julian Epoch.
*
*  Note:
*
*     The Julian Date is supplied in two pieces, in the usual SOFA
*     manner, which is designed to preserve time resolution.  The
*     Julian Date is available as a single number by adding DJ1 and
*     DJ2.  The maximum resolution is achieved if DJ1 is 2451545D0
*     (J2000.0).
*
*  Reference:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*
*-
```

```
      SUBROUTINE iau_EPJ2JD ( EPJ, DJM0, DJM )
*+
*  - - - - - - - - - - -
*   i a u _ E P J 2 J D
*  - - - - - - - - - - -
*
*  Julian Epoch to Julian Date.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ         d     Julian Epoch (e.g. 1996.8D0)
*
*  Returned:
*     DJM0        d     MJD zero-point: always 2400000.5
*     DJM         d     Modified Julian Date
*
*  Note:
*
*     The Julian Date is returned in two pieces, in the usual SOFA
*     manner, which is designed to preserve time resolution.  The
*     Julian Date is available as a single number by adding DJM0 and
*     DJM.
*
*  Reference:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*
*-
```

```
      SUBROUTINE iau_EPV00 ( DATE1, DATE2, PVH, PVB, JSTAT )
*+
*  - - - - - - - - - -
*   i a u _ E P V 0 0
*  - - - - - - - - - -
*
*  Earth position and velocity, heliocentric and barycentric, with
*  respect to the Barycentric Celestial Reference System.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1    d        TDB date part A (Note 1)
*     DATE2    d        TDB date part B (Note 1)
*
*  Returned:
*     PVH      d(3,2)   heliocentric Earth position/velocity (au,au/day)
*     PVB      d(3,2)   barycentric Earth position/velocity (au,au/day)
*     JSTAT    i        status: 0 = OK
*                              +1 = warning: date outside 1900-2100 AD
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     However, the accuracy of the result is more likely to be
*     limited by the algorithm itself than the way the epoch has been
*     expressed.
*
*  2) On return, the arrays PVH and PVB contain the following:
*
*        PVH(1,1)  x       }
*        PVH(2,1)  y       } heliocentric position, au
*        PVH(3,1)  z       }
*
*        PVH(1,2)  xdot    }
*        PVH(2,2)  ydot    } heliocentric velocity, au/d
*        PVH(3,2)  zdot    }
*
*        PVB(1,1)  x       }
*        PVB(2,1)  y       } barycentric position, au
*        PVB(3,1)  z       }
*
*        PVB(1,2)  xdot    }
*        PVB(2,2)  ydot    } barycentric velocity, au/d
*        PVB(3,2)  zdot    }
*
*     The vectors are with respect to the Barycentric Celestial
*     Reference System.  The time unit is one day in TDB.
*
*  3) The routine is a SIMPLIFIED SOLUTION from the planetary theory
*     VSOP2000 (X. Moisson, P. Bretagnon, 2001, Celes. Mechanics &
*     Dyn. Astron., 80, 3/4, 205-213) and is an adaptation of original
```

```
*     Fortran code supplied by P. Bretagnon (private comm., 2000).
*
*  4) Comparisons over the time span 1900-2100 with this simplified
*     solution and the JPL DE405 ephemeris give the following results:
*
*                                RMS    max
*           Heliocentric:
*              position error    3.7   11.2   km
*              velocity error    1.4    5.0   mm/s
*
*           Barycentric:
*              position error    4.6   13.4   km
*              velocity error    1.4    4.9   mm/s
*
*     Comparisons with the JPL DE406 ephemeris show that by 1800 and
*     2200 the position errors are approximately double their 1900-2100
*     size.  By 1500 and 2500 the deterioration is a factor of 10 and by
*     1000 and 3000 a factor of 60.  The velocity accuracy falls off at
*     about half that rate.
*
*-
```

```
      SUBROUTINE iau_EQEC06 ( DATE1, DATE2, DR, DD, DL, DB )
*+
*  - - - - - - - - - - -
*   i a u _ E Q E C 0 6
*  - - - - - - - - - - -
*
*  Transformation from ICRS equatorial coordinates to ecliptic
*  coordinates (mean equinox and ecliptic of date) using IAU 2006
*  precession model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2  d      TT as a 2-part Julian Date (Note 1)
*     DR,DD        d      ICRS right ascension and declination (radians)
*
*  Returned:
*     DL,DB        d      ecliptic longitude and latitude (radians)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) No assumptions are made about whether the coordinates represent
*     starlight and embody astrometric effects such as parallax or
*     aberration.
*
*  3) The transformation is approximately that from mean J2000.0 right
*     ascension and declination to ecliptic longitude and latitude (mean
*     equinox and ecliptic of date), with only frame bias (always less
*     than 25 mas) to disturb this classical picture.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_ECM06    J2000.0 to ecliptic rotation matrix, IAU 2006
*     iau_RXP      product of r-matrix and p-vector
*     iau_C2S      unit vector to spherical coordinates
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EQEQ94 ( DATE1, DATE2 )
*+
*  - - - - - - - - - - -
*   i a u _ E Q E Q 9 4
*  - - - - - - - - - - -
*
*  Equation of the equinoxes, IAU 1994 model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2     d         TDB date (Note 1)
*
*  Returned:
*     iau_EQEQ94      d         equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  Called:
*     iau_ANPM     normalize angle into range +/- pi
*     iau_NUT80    nutation, IAU 1980
*     iau_OBL80    mean obliquity, IAU 1980
*
*  References:
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*     Capitaine, N. & Gontier, A.-M., Astron.Astrophys., 275,
*     645-650 (1993)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_ERA00 ( DJ1, DJ2 )
*+
*  - - - - - - - - - -
*   i a u _ E R A 0 0
*  - - - - - - - - - -
*
*  Earth rotation angle (IAU 2000 model).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DJ1,DJ2     d      UT1 as a 2-part Julian Date (see note)
*
*  The result is the Earth rotation angle (radians), in the range 0 to
*  2pi.
*
*  Notes:
*
*  1) The UT1 date DJ1+DJ2 is a Julian Date, apportioned in any
*     convenient way between the arguments DJ1 and DJ2.  For example,
*     JD(UT1)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DJ1            DJ2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  The date & time method is
*     best matched to the algorithm used:  maximum accuracy (or, at
*     least, minimum noise) is delivered when the DJ1 argument is for
*     0hrs UT1 on the day in question and the DJ2 argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The algorithm is adapted from Expression 22 of Capitaine et al.
*     2000.  The time argument has been expressed in days directly,
*     and, to retain precision, integer contributions have been
*     eliminated.  The same formulation is given in IERS Conventions
*     (2003), Chap. 5, Eq. 14.
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine N., Guinot B. and McCarthy D.D, 2000, Astron.
*     Astrophys., 355, 398-405.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAD03 ( T )
*+
*  - - - - - - - - - -
*   i a u _ F A D 0 3
*  - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean elongation of the Moon from the Sun.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAD03   d    D, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAE03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A E 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Earth.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAE03   d    mean longitude of Earth, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAF03 ( T )
*+
*  - - - - - - - - - -
*   i a u _ F A F 0 3
*  - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of the Moon minus mean longitude of the ascending
*  node.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAF03   d    F, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAJU03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A J U 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Jupiter.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAJU03  d    mean longitude of Jupiter, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAL03 ( T )
*+
*  - - - - - - - - - -
*   i a u _ F A L 0 3
*  - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean anomaly of the Moon.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAL03   d    l, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FALP03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A L P 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean anomaly of the Sun.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FALP03  d    l', radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAMA03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A M A 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Mars.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAMA03  d    mean longitude of Mars, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAME03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A M E 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Mercury.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAME03  d    mean longitude of Mercury, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FANE03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A N E 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Neptune.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FANE03  d    mean longitude of Neptune, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is adapted from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAOM03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A O M 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of the Moon's ascending node.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAOM03  d    Omega, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAPA03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A P A 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  general accumulated precession in longitude.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAPA03  d    general precession in longitude, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003).  It
*     is taken from Kinoshita & Souchay (1990) and comes originally from
*     Lieske et al. (1977).
*
*  References:
*
*     Kinoshita, H. and Souchay J. 1990, Celest.Mech. and Dyn.Astron.
*     48, 187
*
*     Lieske, J.H., Lederle, T., Fricke, W. & Morando, B. 1977,
*     Astron.Astrophys. 58, 1-16
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FASA03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A S A 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Saturn.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FASA03  d    mean longitude of Saturn, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAUR03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A U R 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Uranus.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAUR03  d    mean longitude of Uranus, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     is adapted from Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_FAVE03 ( T )
*+
*  - - - - - - - - - - -
*   i a u _ F A V E 0 3
*  - - - - - - - - - - -
*
*  Fundamental argument, IERS Conventions (2003):
*  mean longitude of Venus.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     T           d    TDB, Julian centuries since J2000.0 (Note 1)
*
*  Returned:
*     iau_FAVE03  d    mean longitude of Venus, radians (Note 2)
*
*  Notes:
*
*  1) Though T is strictly TDB, it is usually more convenient to use TT,
*     which makes no significant difference.
*
*  2) The expression used is as adopted in IERS Conventions (2003) and
*     comes from Souchay et al. (1999) after Simon et al. (1994).
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*-
```

```
      SUBROUTINE iau_FK52H ( R5, D5, DR5, DD5, PX5, RV5,
     :                       RH, DH, DRH, DDH, PXH, RVH )
*+
*  - - - - - - - - - -
*   i a u _ F K 5 2 H
*  - - - - - - - - - -
*
*  Transform FK5 (J2000.0) star data into the Hipparcos system.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given (all FK5, equinox J2000.0, epoch J2000.0):
*     R5        d      RA (radians)
*     D5        d      Dec (radians)
*     DR5       d      proper motion in RA (dRA/dt, rad/Jyear)
*     DD5       d      proper motion in Dec (dDec/dt, rad/Jyear)
*     PX5       d      parallax (arcsec)
*     RV5       d      radial velocity (km/s, positive = receding)
*
*  Returned (all Hipparcos, epoch J2000.0):
*     RH        d      RA (radians)
*     DH        d      Dec (radians)
*     DRH       d      proper motion in RA (dRA/dt, rad/Jyear)
*     DDH       d      proper motion in Dec (dDec/dt, rad/Jyear)
*     PXH       d      parallax (arcsec)
*     RVH       d      radial velocity (km/s, positive = receding)
*
*  Notes:
*
*  1) This routine transforms FK5 star positions and proper motions into
*     the system of the Hipparcos catalog.
*
*  2) The proper motions in RA are dRA/dt rather than cos(Dec)*dRA/dt,
*     and are per year rather than per century.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure rotation
*     and spin;  zonal errors in the FK5 catalog are not taken into
*     account.
*
*  4) See also iau_H2FK5, iau_FK5HZ, iau_HFK5Z.
*
*  Called:
*     iau_STARPV   star catalog data to space motion pv-vector
*     iau_FK5HIP   FK5 to Hipparcos rotation and spin
*     iau_RXP      product of r-matrix and p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PPP      p-vector plus p-vector
*     iau_PVSTAR   space motion pv-vector to star catalog data
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron.Astrophys., 354, 732-739 (2000).
*
*-
```

```
      SUBROUTINE iau_FK5HIP ( R5H, S5H )
*+
*  - - - - - - - - - - -
*   i a u _ F K 5 H I P
*  - - - - - - - - - - -
*
*  FK5 to Hipparcos rotation and spin.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Returned:
*     R5H     d(3,3)   r-matrix: FK5 rotation wrt Hipparcos (Note 2)
*     S5H     d(3)     r-vector: FK5 spin wrt Hipparcos (Note 3)
*
*  Notes:
*
*  1) This routine models the FK5 to Hipparcos transformation as a
*     pure rotation and spin;  zonal errors in the FK5 catalogue are
*     not taken into account.
*
*  2) The r-matrix R5H operates in the sense:
*
*           P_Hipparcos = R5H x P_FK5
*
*     where P_FK5 is a p-vector in the FK5 frame, and P_Hipparcos is
*     the equivalent Hipparcos p-vector.
*
*  3) The r-vector S5H represents the time derivative of the FK5 to
*     Hipparcos rotation.  The units are radians per year (Julian,
*     TDB).
*
*  Called:
*     iau_RV2M     r-vector to r-matrix
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron.Astrophys., 354, 732-739 (2000).
*
*-
```

```
      SUBROUTINE iau_FK5HZ ( R5, D5, DATE1, DATE2, RH, DH )
*+
*  - - - - - - - - - - -
*   i a u _ F K 5 H Z
*  - - - - - - - - - - -
*
*  Transform an FK5 (J2000.0) star position into the system of the
*  Hipparcos catalogue, assuming zero Hipparcos proper motion.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     R5              d      FK5 RA (radians), equinox J2000.0, at date
*     D5              d      FK5 Dec (radians), equinox J2000.0, at date
*     DATE1,DATE2     d      TDB date (Notes 1,2)
*
*  Returned:
*     RH              d      Hipparcos RA (radians)
*     DH              d      Hipparcos Dec (radians)
*
*  Notes:
*
*  1) This routine converts a star position from the FK5 system to
*     the Hipparcos system, in such a way that the Hipparcos proper
*     motion is zero.  Because such a star has, in general, a non-zero
*     proper motion in the FK5 system, the routine requires the date
*     at which the position in the FK5 system was determined.
*
*  2) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure
*     rotation and spin;  zonal errors in the FK5 catalogue are
*     not taken into account.
*
*  4) The position returned by this routine is in the Hipparcos
*     reference system but at date DATE1+DATE2.
*
*  5) See also iau_FK52H, iau_H2FK5, iau_HFK5Z.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_FK5HIP   FK5 to Hipparcos rotation and spin
*     iau_SXP      multiply p-vector by scalar
*     iau_RV2M     r-vector to r-matrix
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_C2S      p-vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
```

```
*
* _
```

```
      SUBROUTINE iau_FW2M ( GAMB, PHIB, PSI, EPS, R )
*+
*  - - - - - - - - -
*   i a u _ F W 2 M
*  - - - - - - - - -
*
*  Form rotation matrix given the Fukushima-Williams angles.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     GAMB       d      F-W angle gamma_bar (radians)
*     PHIB       d      F-W angle phi_bar (radians)
*     PSI        d      F-W angle psi (radians)
*     EPS        d      F-W angle epsilon (radians)
*
*  Returned:
*     R        d(3,3)   rotation matrix
*
*  Notes:
*
*  1) Naming the following points:
*
*           e = J2000.0 ecliptic pole,
*           p = GCRS pole,
*           E = ecliptic pole of date,
*     and   P = CIP,
*
*     the four Fukushima-Williams angles are as follows:
*
*        GAMB = gamma = epE
*        PHIB = phi = pE
*        PSI = psi = pEP
*        EPS = epsilon = EP
*
*  2) The matrix representing the combined effects of frame bias,
*     precession and nutation is:
*
*        NxPxB = R_1(-EPS).R_3(-PSI).R_1(PHIB).R_3(GAMB)
*
*  3) Three different matrices can be constructed, depending on the
*     supplied angles:
*
*     o  To obtain the nutation x precession x frame bias matrix,
*        generate the four precession angles, generate the nutation
*        components and add them to the psi_bar and epsilon_A angles,
*        and call the present routine.
*
*     o  To obtain the precession x frame bias matrix, generate the
*        four precession angles and call the present routine.
*
*     o  To obtain the frame bias matrix, generate the four precession
*        angles for date J2000.0 and call the present routine.
*
*     The nutation-only and precession-only matrices can if necessary
*     be obtained by combining these three appropriately.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RX       rotate around X-axis
*
*  Reference:
*
*     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
*
*-
```

```
      SUBROUTINE iau_FW2XY ( GAMB, PHIB, PSI, EPS, X, Y )
*+
*  - - - - - - - - - -
*   i a u _ F W 2 X Y
*  - - - - - - - - - -
*
*  CIP X,Y given Fukushima-Williams bias-precession-nutation angles.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     GAMB       d      F-W angle gamma_bar (radians)
*     PHIB       d      F-W angle phi_bar (radians)
*     PSI        d      F-W angle psi (radians)
*     EPS        d      F-W angle epsilon (radians)
*
*  Returned:
*     X,Y        d      CIP unit vector X,Y
*
*  Notes:
*
*  1) Naming the following points:
*
*           e = J2000.0 ecliptic pole,
*           p = GCRS pole
*           E = ecliptic pole of date,
*     and   P = CIP,
*
*     the four Fukushima-Williams angles are as follows:
*
*        GAMB = gamma = epE
*        PHIB = phi = pE
*        PSI = psi = pEP
*        EPS = epsilon = EP
*
*  2) The matrix representing the combined effects of frame bias,
*     precession and nutation is:
*
*        NxPxB = R_1(-EPSA).R_3(-PSI).R_1(PHIB).R_3(GAMB)
*
*     The returned values x,y are elements (3,1) and (3,2) of the
*     matrix.  Near J2000.0, they are essentially angles in radians.
*
*  Called:
*     iau_FW2M     F-W angles to r-matrix
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*
*  Reference:
*
*     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
*
*-
```

```
      SUBROUTINE iau_G2ICRS ( DL, DB, DR, DD )
*+
*  - - - - - - - - - - -
*   i a u _ G 2 I C R S
*  - - - - - - - - - - -
*
*  Transformation from Galactic Coordinates to ICRS.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DL       d      galactic longitude (radians)
*     DB       d      galactic latitude (radians)
*
*  Returned:
*     DR       d      ICRS right ascension (radians)
*     DD       d      ICRS declination (radians)
*
*  Notes:
*
*  1) The IAU 1958 system of Galactic coordinates was defined with
*     respect to the now obsolete reference system FK4 B1950.0.  When
*     interpreting the system in a modern context, several factors have
*     to be taken into account:
*
*     . The inclusion in FK4 positions of the E-terms of aberration.
*
*     . The distortion of the FK4 proper motion system by differential
*       Galactic rotation.
*
*     . The use of the B1950.0 equinox rather than the now-standard
*       J2000.0.
*
*     . The frame bias between ICRS and the J2000.0 mean place system.
*
*     The Hipparcos Catalogue (Perryman & ESA 1997) provides a rotation
*     matrix that transforms directly between ICRS and Galactic
*     coordinates with the above factors taken into account.  The
*     matrix is derived from three angles, namely the ICRS coordinates
*     of the Galactic pole and the longitude of the ascending node of
*     the galactic equator on the ICRS equator.  They are given in
*     degrees to five decimal places and for canonical purposes are
*     regarded as exact.  In the Hipparcos Catalogue the matrix elements
*     are given to 10 decimal places (about 20 microarcsec).  In the
*     present SOFA routine the matrix elements have been recomputed from
*     the canonical three angles and are given to 30 decimal places.
*
*  2) The inverse transformation is performed by the routine iau_ICRS2G.
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*     iau_S2C      spherical coordinates to unit vector
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_C2S      p-vector to spherical
*
*  Reference:
*     Perryman M.A.C. & ESA, 1997, ESA SP-1200, The Hipparcos and Tycho
*     catalogues.  Astrometric and photometric star catalogues
*     derived from the ESA Hipparcos Space Astrometry Mission.  ESA
*     Publications Division, Noordwijk, Netherlands.
*
*-
```

```
      SUBROUTINE iau_GC2GD ( N, XYZ, ELONG, PHI, HEIGHT, J )
*+
*  - - - - - - - - - -
*   i a u _ G C 2 G D
*  - - - - - - - - - -
*
*  Transform geocentric coordinates to geodetic using the specified
*  reference ellipsoid.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical transformation.
*
*  Given:
*     N         i     ellipsoid identifier (Note 1)
*     XYZ       d(3)  geocentric vector (Note 2)
*
*  Returned:
*     ELONG     d     longitude (radians, east +ve, Note 3)
*     PHI       d     latitude (geodetic, radians, Note 3)
*     HEIGHT    d     height above ellipsoid (geodetic, Notes 2,3)
*     J         i     status:  0 = OK
*                             -1 = illegal identifier (Note 3)
*                             -2 = internal error (Note 3)
*
*  Notes:
*
*  1) The identifier N is a number that specifies the choice of
*     reference ellipsoid.  The following are supported:
*
*        N   ellipsoid
*
*        1    WGS84
*        2    GRS80
*        3    WGS72
*
*     The number N has no significance outside the SOFA software.
*
*  2) The geocentric vector (XYZ, given) and height (HEIGHT, returned)
*     are in meters.
*
*  3) An error status J=-1 means that the identifier N is illegal.  An
*     error status J=-2 is theoretically impossible.  In all error
*     cases, all three results are set to -1D9.
*
*  4) The inverse transformation is performed in the routine iau_GD2GC.
*
*  Called:
*     iau_EFORM    Earth reference ellipsoids
*     iau_GC2GDE   geocentric to geodetic transformation, general
*
*-
```

```
      SUBROUTINE iau_GC2GDE ( A, F, XYZ, ELONG, PHI, HEIGHT, J )
*+
*  - - - - - - - - - - -
*   i a u _ G C 2 G D E
*  - - - - - - - - - - -
*
*  Transform geocentric coordinates to geodetic for a reference
*  ellipsoid of specified form.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     A         d      equatorial radius (Notes 2,4)
*     F         d      flattening (Note 3)
*     XYZ       d(3)   geocentric vector (Note 4)
*
*  Returned:
*     ELONG     d      longitude (radians, east +ve)
*     PHI       d      latitude (geodetic, radians)
*     HEIGHT    d      height above ellipsoid (geodetic, Note 4)
*     J         i      status:  0 = OK
*                              -1 = illegal F
*                              -2 = illegal A
*
*  Notes:
*
*  1) This routine is closely based on the GCONV2H subroutine by
*     Toshio Fukushima (see reference).
*
*  2) The equatorial radius, A, can be in any units, but meters is
*     the conventional choice.
*
*  3) The flattening, F, is (for the Earth) a value around 0.00335,
*     i.e. around 1/298.
*
*  4) The equatorial radius, A, and the geocentric vector, XYZ,
*     must be given in the same units, and determine the units of
*     the returned height, HEIGHT.
*
*  5) If an error occurs (J<0), ELONG, PHI and HEIGHT are unchanged.
*
*  6) The inverse transformation is performed in the routine iau_GD2GCE.
*
*  7) The transformation for a standard ellipsoid (such as WGS84) can
*     more conveniently be performed by calling iau_GC2GD, which uses a
*     numerical code (1 for WGS84) to identify the required A and F
*     values.
*
*  Reference:
*
*     Fukushima, T., "Transformation from Cartesian to geodetic
*     coordinates accelerated by Halley's method", J.Geodesy (2006)
*     79: 689-693
*
*-
```

```
      SUBROUTINE iau_GD2GC ( N, ELONG, PHI, HEIGHT, XYZ, J )
*+
*  - - - - - - - - - -
*   i a u _ G D 2 G C
*  - - - - - - - - - -
*
*  Transform geodetic coordinates to geocentric using the specified
*  reference ellipsoid.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical transformation.
*
*  Given:
*     N         i     ellipsoid identifier (Note 1)
*     ELONG     d     longitude (radians, east +ve)
*     PHI       d     latitude (geodetic, radians, Note 3)
*     HEIGHT    d     height above ellipsoid (geodetic, Notes 2,3)
*
*  Returned:
*     XYZ       d(3)  geocentric vector (Note 2)
*     J         i     status:  0 = OK
*                             -1 = illegal identifier (Note 3)
*                             -2 = illegal case (Note 3)
*
*  Notes:
*
*  1) The identifier N is a number that specifies the choice of
*     reference ellipsoid.  The following are supported:
*
*        N   ellipsoid
*
*        1    WGS84
*        2    GRS80
*        3    WGS72
*
*     The number N has no significance outside the SOFA software.
*
*  2) The height (HEIGHT, given) and the geocentric vector (XYZ,
*     returned) are in meters.
*
*  3) No validation is performed on the arguments ELONG, PHI and HEIGHT.
*     An error status J=-1 means that the identifier N is illegal.  An
*     error status J=-2 protects against cases that would lead to
*     arithmetic exceptions.  In all error cases, XYZ is set to zeros.
*
*  4) The inverse transformation is performed in the routine iau_GC2GD.
*
*  Called:
*     iau_EFORM    Earth reference ellipsoids
*     iau_GD2GCE   geodetic to geocentric transformation, general
*     iau_ZP       zero p-vector
*
*-
```

```
      SUBROUTINE iau_GD2GCE ( A, F, ELONG, PHI, HEIGHT, XYZ, J )
*+
*  - - - - - - - - - - -
*   i a u _ G D 2 G C E
*  - - - - - - - - - - -
*
*  Transform geodetic coordinates to geocentric for a reference
*  ellipsoid of specified form.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     A         d     equatorial radius (Notes 1,4)
*     F         d     flattening (Notes 2,4)
*     ELONG     d     longitude (radians, east +ve)
*     PHI       d     latitude (geodetic, radians, Note 4)
*     HEIGHT    d     height above ellipsoid (geodetic, Notes 3,4)
*
*  Returned:
*     XYZ       d(3)  geocentric vector (Note 3)
*     J         i     status:  0 = OK
*                             -1 = illegal case (Note 4)
*
*  Notes:
*
*  1) The equatorial radius, A, can be in any units, but meters is
*     the conventional choice.
*
*  2) The flattening, F, is (for the Earth) a value around 0.00335,
*     i.e. around 1/298.
*
*  3) The equatorial radius, A, and the height, HEIGHT, must be
*     given in the same units, and determine the units of the
*     returned geocentric vector, XYZ.
*
*  4) No validation is performed on individual arguments.  The error
*     status J=-1 protects against (unrealistic) cases that would lead
*     to arithmetic exceptions.  If an error occurs, XYZ is unchanged.
*
*  5) The inverse transformation is performed in the routine iau_GC2GDE.
*
*  6) The transformation for a standard ellipsoid (such as WGS84) can
*     more conveniently be performed by calling iau_GD2GC, which uses a
*     numerical code (1 for WGS84) to identify the required A and F
*     values.
*
*  References:
*
*     Green, R.M., Spherical Astronomy, Cambridge University Press,
*     (1985) Section 4.5, p96.
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 4.22, p202.
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GMST00 ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G M S T 0 0
*  - - - - - - - - - - -
*
*  Greenwich Mean Sidereal Time (model consistent with IAU 2000
*  resolutions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GMST00   d      Greenwich mean sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*             Part A         Part B
*
*          2450123.7D0        0D0        (JD method)
*           2451545D0      -1421.3D0     (J2000 method)
*          2400000.5D0     50123.2D0     (MJD method)
*          2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession.  If UT1 is used for
*     both purposes, errors of order 100 microarcseconds result.
*
*  3) This GMST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation and equation of the equinoxes.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  5) The algorithm is from Capitaine et al. (2003) and IERS Conventions
*     2003.
*
*  Called:
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
```

*_

```
      DOUBLE PRECISION FUNCTION iau_GMST06 ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G M S T 0 6
*  - - - - - - - - - - -
*
*  Greenwich mean sidereal time (consistent with IAU 2006 precession).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GMST06   d      Greenwich mean sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     rotation angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession.  If UT1 is used for
*     both purposes, errors of order 100 microarcseconds result.
*
*  3) This GMST is compatible with the IAU 2006 precession and must not
*     be used with other precession models.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  Called:
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  Reference:
*
*     Capitaine, N., Wallace, P.T. & Chapront, J., 2005,
*     Astron.Astrophys. 432, 355
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GMST82 ( DJ1, DJ2 )
*+
*  - - - - - - - - - - -
*   i a u _ G M S T 8 2
*  - - - - - - - - - - -
*
*  Universal Time to Greenwich Mean Sidereal Time (IAU 1982 model).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DJ1, DJ2     d      UT1 Julian Date (see note)
*
*  Returned:
*     iau_GMST82   d      Greenwich mean sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 epoch DJ1+DJ2 is a Julian Date, apportioned in any
*     convenient way between the arguments DJ1 and DJ2.  For example,
*     JD(UT1)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DJ1            DJ2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  The date & time method is
*     best matched to the algorithm used:  maximum accuracy (or, at
*     least, minimum noise) is delivered when the DJ1 argument is for
*     0hrs UT1 on the day in question and the DJ2 argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The algorithm is based on the IAU 1982 expression.  This is always
*     described as giving the GMST at 0 hours UT1.  In fact, it gives the
*     difference between the GMST and the UT, the steady 4-minutes-per-day
*     drawing-ahead of ST with respect to UT.  When whole days are ignored,
*     the expression happens to equal the GMST at 0 hours UT1 each day.
*
*  3) In this routine, the entire UT1 (the sum of the two arguments DJ1
*     and DJ2) is used directly as the argument for the standard formula,
*     the constant term of which is adjusted by 12 hours to take account
*     of the noon phasing of Julian Date.  The UT1 is then added, but
*     omitting whole days to conserve accuracy.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Transactions of the International Astronomical Union,
*     XVIII B, 67 (1983).
*
*     Aoki et al., Astron.Astrophys., 105, 359-361 (1982).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST00A ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G S T 0 0 A
*  - - - - - - - - - - -
*
*  Greenwich Apparent Sidereal Time (consistent with IAU 2000
*  resolutions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GST00A   d      Greenwich apparent sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession-nutation.  If UT1 is
*     used for both purposes, errors of order 100 microarcseconds
*     result.
*
*  3) This GAST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  5) The algorithm is from Capitaine et al. (2003) and IERS Conventions
*     2003.
*
*  Called:
*     iau_GMST00   Greenwich mean sidereal time, IAU 2000
*     iau_EE00A    equation of the equinoxes, IAU 2000A
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
```

```
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST00B ( UTA, UTB )
*+
*  - - - - - - - - - - -
*   i a u _ G S T 0 0 B
*  - - - - - - - - - - -
*
*  Greenwich Apparent Sidereal Time (consistent with IAU 2000
*  resolutions but using the truncated nutation model IAU 2000B).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GST00B   d      Greenwich apparent sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 date UTA+UTB is a Julian Date, apportioned in any
*     convenient way between the argument pair.  For example,
*     JD=2450123.7 could be expressed in any of these ways, among
*     others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The result is compatible with the IAU 2000 resolutions, except
*     that accuracy has been compromised for the sake of speed and
*     convenience in two respects:
*
*     . UT is used instead of TDB (or TT) to compute the precession
*       component of GMST and the equation of the equinoxes.  This
*       results in errors of order 0.1 mas at present.
*
*     . The IAU 2000B abridged nutation model (McCarthy & Luzum, 2001)
*       is used, introducing errors of up to 1 mas.
*
*  3) This GAST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  5) The algorithm is from Capitaine et al. (2003) and IERS Conventions
*     2003.
*
*  Called:
*     iau_GMST00   Greenwich mean sidereal time, IAU 2000
*     iau_EE00B    equation of the equinoxes, IAU 2000B
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions to
```

```
*     implement the IAU 2000 definition of UT1", Astronomy &
*     Astrophysics, 406, 1135-1149 (2003)
*
*     McCarthy, D.D. & Luzum, B.J., "An abridged model of the
*     precession-nutation of the celestial pole", Celestial Mechanics &
*     Dynamical Astronomy, 85, 37-49 (2003)
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST06 ( UTA, UTB, TTA, TTB, RNPB )
*+
*  - - - - - - - - - -
*   i a u _ G S T 0 6
*  - - - - - - - - - -
*
*  Greenwich apparent sidereal time, IAU 2006, given the NPB matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*     RNPB       d(3,3)   nutation x precession x bias matrix
*
*  Returned:
*     iau_GST06    d      Greenwich apparent sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     rotation angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession-nutation.  If UT1 is
*     used for both purposes, errors of order 100 microarcseconds
*     result.
*
*  3) Although the routine uses the IAU 2006 series for s+XY/2, it is
*     otherwise independent of the precession-nutation model and can in
*     practice be used with any equinox-based NPB matrix.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  Called:
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ERA00    Earth rotation angle, IAU 2000
*     iau_EORS     equation of the origins, given NPB matrix and s
*
*  Reference:
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST06A ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G S T 0 6 A
*  - - - - - - - - - - -
*
*  Greenwich apparent sidereal time (consistent with IAU 2000 and 2006
*  resolutions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GST06A   d      Greenwich apparent sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     rotation angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession-nutation.  If UT1 is
*     used for both purposes, errors of order 100 microarcseconds
*     result.
*
*  3) This GAST is compatible with the IAU 2000/2006 resolutions and
*     must be used only in conjunction with IAU 2006 precession and
*     IAU 2000A nutation.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  Called:
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_GST06    Greenwich apparent ST, IAU 2006, given NPB matrix
*
*  Reference:
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST94 ( UTA, UTB )
*+
*  - - - - - - - - - -
*   i a u _ G S T 9 4
*  - - - - - - - - - -
*
*  Greenwich Apparent Sidereal Time (consistent with IAU 1982/94
*  resolutions).
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*
*  Returned:
*     iau_GST94    d      Greenwich apparent sidereal time (radians)
*
*  Notes:
*
*  1) The UT1 date UTA+UTB is a Julian Date, apportioned in any
*     convenient way between the argument pair.  For example,
*     JD=2450123.7 could be expressed in any of these ways, among
*     others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The result is compatible with the IAU 1982 and 1994 resolutions,
*     except that accuracy has been compromised for the sake of
*     convenience in that UT is used instead of TDB (or TT) to compute
*     the equation of the equinoxes.
*
*  3) This GAST must be used only in conjunction with contemporaneous
*     IAU standards such as 1976 precession, 1980 obliquity and 1982
*     nutation.  It is not compatible with the IAU 2000 resolutions.
*
*  4) The result is returned in the range 0 to 2pi.
*
*  Called:
*     iau_GMST82   Greenwich mean sidereal time, IAU 1982
*     iau_EQEQ94   equation of the equinoxes, IAU 1994
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*-
```

```
      SUBROUTINE iau_H2FK5 ( RH, DH, DRH, DDH, PXH, RVH,
     :                       R5, D5, DR5, DD5, PX5, RV5 )
*+
*  - - - - - - - - - -
*   i a u _ H 2 F K 5
*  - - - - - - - - - -
*
*  Transform Hipparcos star data into the FK5 (J2000.0) system.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given (all Hipparcos, epoch J2000.0):
*     RH         d      RA (radians)
*     DH         d      Dec (radians)
*     DRH        d      proper motion in RA (dRA/dt, rad/Jyear)
*     DDH        d      proper motion in Dec (dDec/dt, rad/Jyear)
*     PXH        d      parallax (arcsec)
*     RVH        d      radial velocity (km/s, positive = receding)
*
*  Returned (all FK5, equinox J2000.0, epoch J2000.0):
*     R5         d      RA (radians)
*     D5         d      Dec (radians)
*     DR5        d      proper motion in RA (dRA/dt, rad/Jyear)
*     DD5        d      proper motion in Dec (dDec/dt, rad/Jyear)
*     PX5        d      parallax (arcsec)
*     RV5        d      radial velocity (km/s, positive = receding)
*
*  Notes:
*
*  1) This routine transforms Hipparcos star positions and proper
*     motions into FK5 J2000.0.
*
*  2) The proper motions in RA are dRA/dt rather than cos(Dec)*dRA/dt,
*     and are per year rather than per century.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure rotation
*     and spin;  zonal errors in the FK5 catalog are not taken into
*     account.
*
*  4) See also iau_FK52H, iau_FK5HZ, iau_HFK5Z.
*
*  Called:
*     iau_STARPV   star catalog data to space motion pv-vector
*     iau_FK5HIP   FK5 to Hipparcos rotation and spin
*     iau_RV2M     r-vector to r-matrix
*     iau_RXP      product of r-matrix and p-vector
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PMP      p-vector minus p-vector
*     iau_PVSTAR   space motion pv-vector to star catalog data
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron.Astrophys., 354, 732-739 (2000).
*
*-
```

```
      SUBROUTINE iau_HD2AE ( HA, DEC, PHI, AZ, EL )
*+
*  − − − − − − − − − −
*   i a u _ H D 2 A E
*  − − − − − − − − − −
*
*  Equatorial to horizon coordinates:  transform hour angle and
*  declination to azimuth and altitude.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     HA       d     hour angle (local)
*     DEC      d     declination
*     PHI      d     site latitude
*
*  Returned:
*     AZ       d     azimuth
*     EL       d     altitude (informally, elevation)
*
*  Notes:
*
*  1)  All the arguments are angles in radians.
*
*  2)  Azimuth is returned in the range 0−2pi;  north is zero, and east
*      is +pi/2.  Altitude is returned in the range +/− pi/2.
*
*  3)  The latitude PHI is pi/2 minus the angle between the Earth’s
*      rotation axis and the adopted zenith.  In many applications it
*      will be sufficient to use the published geodetic latitude of the
*      site.  In very precise (sub−arcsecond) applications, PHI can be
*      corrected for polar motion.
*
*  4)  The returned azimuth AZ is with respect to the rotational north
*      pole, as opposed to the ITRS pole, and for sub−arcsecond accuracy
*      will need to be adjusted for polar motion if it is to be with
*      respect to north on a map of the Earth’s surface.
*
*  5)  Should the user wish to work with respect to the astronomical
*      zenith rather than the geodetic zenith, PHI will need to be
*      adjusted for deflection of the vertical (often tens of
*      arcseconds), and the zero point of HA will also be affected.
*
*  6)  The transformation is the same as Vh = Rz(pi)*Ry(pi/2−phi)*Ve,
*      where Vh and Ve are lefthanded unit vectors in the (az,el) and
*      (ha,dec) systems respectively and Ry and Rz are rotations about
*      first the y−axis and then the z−axis.  (n.b. Rz(pi) simply
*      reverses the signs of the x and y components.)  For efficiency,
*      the algorithm is written out rather than calling other utility
*      functions.  For applications that require even greater
*      efficiency, additional savings are possible if constant terms
*      such as functions of latitude are computed once and for all.
*
*  7)  Again for efficiency, no range checking of arguments is carried
*      out.
*
*  Last revision:   2018 January 2
*
*  SOFA release 2018−01−30
*
*  Copyright (C) 2018 IAU SOFA Board.  See notes at end.
*
*−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

      IMPLICIT NONE

      DOUBLE PRECISION HA, DEC, PHI, AZ, EL

      DOUBLE PRECISION D2PI
```

```
      PARAMETER ( D2PI = 6.283185307179586476925287D0 )

      DOUBLE PRECISION SH, CH, SD, CD, SP, CP, X, Y, Z, R, A


*  Useful trig functions.
      SH = SIN(HA)
      CH = COS(HA)
      SD = SIN(DEC)
      CD = COS(DEC)
      SP = SIN(PHI)
      CP = COS(PHI)

*  Az,Alt unit vector.
      X = - CH*CD*SP + SD*CP
      Y = - SH*CD
      Z = CH*CD*CP + SD*SP

*  To spherical.
      R = SQRT(X*X + Y*Y)
      IF ( R.EQ.0D0 ) THEN
         A = 0D0
      ELSE
         A = ATAN2(Y,X)
      END IF
      IF ( A.LT.0D0 ) A = A+D2PI
      AZ = A
      EL = ATAN2(Z,R)

*  Finished.

*+----------------------------------------------------------------------
*
*  Copyright (C) 2018
*  Standards Of Fundamental Astronomy Board
*  of the International Astronomical Union.
*
*  =====================
*  SOFA Software License
*  =====================
*
*  NOTICE TO USER:
*
*  BY USING THIS SOFTWARE YOU ACCEPT THE FOLLOWING SIX TERMS AND
*  CONDITIONS WHICH APPLY TO ITS USE.
*
*  1. The Software is owned by the IAU SOFA Board ("SOFA").
*
*  2. Permission is granted to anyone to use the SOFA software for any
*     purpose, including commercial applications, free of charge and
*     without payment of royalties, subject to the conditions and
*     restrictions listed below.
*
*  3. You (the user) may copy and distribute SOFA source code to others,
*     and use and adapt its code and algorithms in your own software,
*     on a world-wide, royalty-free basis.  That portion of your
*     distribution that does not consist of intact and unchanged copies
*     of SOFA source code files is a "derived work" that must comply
*     with the following requirements:
*
*     a) Your work shall be marked or carry a statement that it
*        (i) uses routines and computations derived by you from
*        software provided by SOFA under license to you; and
*        (ii) does not itself constitute software provided by and/or
*        endorsed by SOFA.
*
*     b) The source code of your derived work must contain descriptions
*        of how the derived work is based upon, contains and/or differs
*        from the original SOFA software.
*
*     c) The names of all routines in your derived work shall not
*        include the prefix "iau" or "sofa" or trivial modifications
*        thereof such as changes of case.
```

```
*
*     d) The origin of the SOFA components of your derived work must
*        not be misrepresented;  you must not claim that you wrote the
*        original software, nor file a patent application for SOFA
*        software or algorithms embedded in the SOFA software.
*
*     e) These requirements must be reproduced intact in any source
*        distribution and shall apply to anyone to whom you have
*        granted a further right to modify the source code of your
*        derived work.
*
*     Note that, as originally distributed, the SOFA software is
*     intended to be a definitive implementation of the IAU standards,
*     and consequently third-party modifications are discouraged.  All
*     variations, no matter how minor, must be explicitly marked as
*     such, as explained above.
*
*  4. You shall not cause the SOFA software to be brought into
*     disrepute, either by misuse, or use for inappropriate tasks, or
*     by inappropriate modification.
*
*  5. The SOFA software is provided "as is" and SOFA makes no warranty
*     as to its use or performance.   SOFA does not and cannot warrant
*     the performance or results which the user may obtain by using the
*     SOFA software.  SOFA makes no warranties, express or implied, as
*     to non-infringement of third party rights, merchantability, or
*     fitness for any particular purpose.  In no event will SOFA be
*     liable to the user for any consequential, incidental, or special
*     damages, including any lost profits or lost savings, even if a
*     SOFA representative has been advised of such damages, or for any
*     claim by any third party.
*
*  6. The provision of any version of the SOFA software under the terms
*     and conditions specified herein does not imply that future
*     versions will also be made available under the same terms and
*     conditions.
*
*  In any published work or commercial product which uses the SOFA
*  software directly, acknowledgement (see www.iausofa.org) is
*  appreciated.
*
*  Correspondence concerning SOFA software should be addressed as
*  follows:
*
*      By email:  sofa@ukho.gov.uk
*      By post:   IAU SOFA Center
*                 HM Nautical Almanac Office
*                 UK Hydrographic Office
*                 Admiralty Way, Taunton
*                 Somerset, TA1 2DN
*                 United Kingdom
*
*-----------------------------------------------------------------------

      END
```

```
      DOUBLE PRECISION FUNCTION iau_HD2PA ( HA, DEC, PHI )
*+
*  - - - - - - - - - -
*   i a u _ H D 2 P A
*  - - - - - - - - - -
*
*  Parallactic angle for a given hour angle and declination.
*
*  Given:
*     HA          d     hour angle
*     DEC         d     declination
*     PHI         d     site latitude
*
*  Returned:
*     iau_HD2PA   d     parallactic angle
*
*  Notes:
*
*  1)  All the arguments are angles in radians.
*
*  2)  The parallactic angle at a point in the sky is the position angle
*      of the vertical, i.e. the angle between the directions to the
*      north celestial pole and to the zenith respectively.
*
*  3)  The result is returned in the range -pi to +pi.
*
*  4)  At the pole itself a zero result is returned.
*
*  5)  The latitude PHI is pi/2 minus the angle between the Earth's
*      rotation axis and the adopted zenith.  In many applications it
*      will be sufficient to use the published geodetic latitude of the
*      site.  In very precise (sub-arcsecond) applications, PHI can be
*      corrected for polar motion.
*
*  6)  Should the user wish to work with respect to the astronomical
*      zenith rather than the geodetic zenith, PHI will need to be
*      adjusted for deflection of the vertical (often tens of
*      arcseconds), and the zero point of HA will also be affected.
*
*  Reference:
*     Smart, W.M., "Spherical Astronomy", Cambridge University Press,
*     6th edition (Green, 1977), p49.
*
*-
```

```
      SUBROUTINE iau_HFK5Z ( RH, DH, DATE1, DATE2, R5, D5, DR5, DD5 )
*+
*  - - - - - - - - - -
*   i a u _ H F K 5 Z
*  - - - - - - - - - -
*
*  Transform a Hipparcos star position into FK5 J2000.0, assuming
*  zero Hipparcos proper motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RH              d      Hipparcos RA (radians)
*     DH              d      Hipparcos Dec (radians)
*     DATE1,DATE2     d      TDB date (Note 1)
*
*  Returned (all FK5, equinox J2000.0, date DATE1+DATE2):
*     R5              d      RA (radians)
*     D5              d      Dec (radians)
*     DR5             d      FK5 RA proper motion (rad/year, Note 4)
*     DD5             d      Dec proper motion (rad/year, Note 4)
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure
*     rotation and spin;  zonal errors in the FK5 catalogue are
*     not taken into account.
*
*  4) It was the intention that Hipparcos should be a close
*     approximation to an inertial frame, so that distant objects
*     have zero proper motion;  such objects have (in general)
*     non-zero proper motion in FK5, and this routine returns those
*     fictitious proper motions.
*
*  5) The position returned by this routine is in the FK5 J2000.0
*     reference system but at date DATE1+DATE2.
*
*  6) See also iau_FK52H, iau_H2FK5, iau_FK5ZHZ.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_FK5HIP   FK5 to Hipparcos rotation and spin
*     iau_RXP      product of r-matrix and p-vector
*     iau_SXP      multiply p-vector by scalar
*     iau_RXR      product of two r-matrices
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PV2S     pv-vector to spherical
```

```
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
*
*-
```

```
      SUBROUTINE iau_ICRS2G ( DR, DD, DL, DB  )
*+
*  - - - - - - - - - - -
*   i a u _ I C R S 2 G
*  - - - - - - - - - - -
*
*  Transformation from ICRS to Galactic Coordinates.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DR       d      ICRS right ascension (radians)
*     DD       d      ICRS declination (radians)
*
*  Returned:
*     DL       d      galactic longitude (radians)
*     DB       d      galactic latitude (radians)
*
*  Notes:
*
*  1) The IAU 1958 system of Galactic coordinates was defined with
*     respect to the now obsolete reference system FK4 B1950.0.  When
*     interpreting the system in a modern context, several factors have
*     to be taken into account:
*
*     . The inclusion in FK4 positions of the E-terms of aberration.
*
*     . The distortion of the FK4 proper motion system by differential
*       Galactic rotation.
*
*     . The use of the B1950.0 equinox rather than the now-standard
*       J2000.0.
*
*     . The frame bias between ICRS and the J2000.0 mean place system.
*
*     The Hipparcos Catalogue (Perryman & ESA 1997) provides a rotation
*     matrix that transforms directly between ICRS and Galactic
*     coordinates with the above factors taken into account.  The
*     matrix is derived from three angles, namely the ICRS coordinates
*     of the Galactic pole and the longitude of the ascending node of
*     the galactic equator on the ICRS equator.  They are given in
*     degrees to five decimal places and for canonical purposes are
*     regarded as exact.  In the Hipparcos Catalogue the matrix elements
*     are given to 10 decimal places (about 20 microarcsec).  In the
*     present SOFA routine the matrix elements have been recomputed from
*     the canonical three angles and are given to 30 decimal places.
*
*  2) The inverse transformation is performed by the routine iau_G2ICRS.
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*     iau_S2C      spherical coordinates to unit vector
*     iau_RXP      product of r-matrix and p-vector
*     iau_C2S      p-vector to spherical
*
*  Reference:
*     Perryman M.A.C. & ESA, 1997, ESA SP-1200, The Hipparcos and Tycho
*     catalogues.  Astrometric and photometric star catalogues
*     derived from the ESA Hipparcos Space Astrometry Mission.  ESA
*     Publications Division, Noordwijk, Netherlands.
*
*-
```

```
      SUBROUTINE iau_IR ( R )
*+
*  - - - - - - -
*   i a u _ I R
*  - - - - - - -
*
*  Initialize an r-matrix to the identity matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Returned:
*     R        d(3,3)    r-matrix
*
*-
```

```
      SUBROUTINE iau_JD2CAL ( DJ1, DJ2, IY, IM, ID, FD, J )
*+
*  - - - - - - - - - - -
*   i a u _ J D 2 C A L
*  - - - - - - - - - - -
*
*  Julian Date to Gregorian year, month, day, and fraction of a day.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DJ1,DJ2     d      Julian Date (Notes 1, 2)
*
*  Returned:
*     IY          i      year
*     IM          i      month
*     ID          i      day
*     FD          d      fraction of day
*     J           i      status:
*                           0 = OK
*                          -1 = unacceptable date (Note 1)
*
*  Notes:
*
*  1) The earliest valid date is -68569.5 (-4900 March 1).  The
*     largest value accepted is 10^9.
*
*  2) The Julian Date is apportioned in any convenient way between
*     the arguments DJ1 and DJ2.  For example, JD=2450123.7 could
*     be expressed in any of these ways, among others:
*
*             DJ1            DJ2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*  3) In early eras the conversion is from the "Proleptic Gregorian
*     Calendar";  no account is taken of the date(s) of adoption of
*     the Gregorian Calendar, nor is the AD/BC numbering convention
*     observed.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 12.92 (p604).
*
*-
```

```
      SUBROUTINE iau_JDCALF ( NDP, DJ1, DJ2, IYMDF, J )
*+
*  − − − − − − − − − − −
*   i a u _ J D C A L F
*  − − − − − − − − − − −
*
*  Julian Date to Gregorian Calendar, expressed in a form convenient
*  for formatting messages:  rounded to a specified precision, and with
*  the fields stored in a single array.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     NDP         i     number of decimal places of days in fraction
*     DJ1,DJ2     d     DJ1+DJ2 = Julian Date (Note 1)
*
*  Returned:
*     IYMDF       i(4)  year, month, day, fraction in Gregorian
*                       calendar
*     J           i     status:
*                          −1 = date out of range
*                           0 = OK
*                          +1 = NDP not 0−9 (interpreted as 0)
*
*  Notes:
*
*  1) The Julian Date is apportioned in any convenient way between
*     the arguments DJ1 and DJ2.  For example, JD=2450123.7 could
*     be expressed in any of these ways, among others:
*
*             DJ1            DJ2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      −1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*  2) In early eras the conversion is from the "Proleptic Gregorian
*     Calendar";  no account is taken of the date(s) of adoption of
*     the Gregorian Calendar, nor is the AD/BC numbering convention
*     observed.
*
*  3) Refer to the routine iau_JD2CAL.
*
*  4) NDP should be 4 or less if internal overflows are to be
*     avoided on machines which use 16−bit integers.
*
*  Called:
*     iau_JD2CAL   JD to Gregorian calendar
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 12.92 (p604).
*
*−
```

```
      SUBROUTINE iau_LD ( BM, P, Q, E, EM, DLIM, P1 )
*+
*  - - - - - - -
*   i a u _ L D
*  - - - - - - -
*
*  Apply light deflection by a solar-system body, as part of
*  transforming coordinate direction into natural direction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     BM       d      mass of the gravitating body (solar masses)
*     P        d(3)   direction from observer to source (unit vector)
*     Q        d(3)   direction from body to source (unit vector)
*     E        d(3)   direction from body to observer (unit vector)
*     EM       d      distance from body to observer (au)
*     DLIM     d      deflection limiter (Note 4)
*
*  Returned:
*     P1       d(3)   observer to deflected source (unit vector)
*
*  Notes:
*
*  1) The algorithm is based on Expr. (70) in Klioner (2003) and
*     Expr. (7.63) in the Explanatory Supplement (Urban & Seidelmann
*     2013), with some rearrangement to minimize the effects of machine
*     precision.
*
*  2) The mass parameter BM can, as required, be adjusted in order to
*     allow for such effects as quadrupole field.
*
*  3) The barycentric position of the deflecting body should ideally
*     correspond to the time of closest approach of the light ray to
*     the body.
*
*  4) The deflection limiter parameter DLIM is phi^2/2, where phi is the
*     angular separation (in radians) between source and body at which
*     limiting is applied.  As phi shrinks below the chosen threshold,
*     the deflection is artificially reduced, reaching zero for phi = 0.
*
*  5) The returned vector P1 is not normalized, but the consequential
*     departure from unit magnitude is always negligible.
*
*  6) To accumulate total light deflection taking into account the
*     contributions from several bodies, call the present routine for
*     each body in succession, in decreasing order of distance from the
*     observer.
*
*  7) For efficiency, validation is omitted.  The supplied vectors must
*     be of unit magnitude, and the deflection limiter non-zero and
*     positive.
*
*  References:
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013).
*
*     Klioner, Sergei A., "A practical relativistic model for micro-
*     arcsecond astrometry in space", Astr. J. 125, 1580-1597 (2003).
*
*  Called:
*     iau_PDP      scalar product of two p-vectors
*     iau_PXP      vector product of two p-vectors
*
*-
```

```
      SUBROUTINE iau_LDN ( N, B, OB, SC, SN )
*+
*  - - - - - - - -
*   i a u _ L D N
*  - - - - - - - -
*
*  For a star, apply light deflection by multiple solar-system bodies,
*  as part of transforming coordinate direction into natural direction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     N     i        number of bodies (Note 1)
*     B     d(8,N)   data for each of the N bodies (Notes 1,2):
*            (1,I)     mass of the body (solar masses, Note 3)
*            (2,I)     deflection limiter (Note 4)
*            (3-5,I)   barycentric position of the body (au)
*            (6-8,I)   barycentric velocity of the body (au/day)
*     OB    d(3)     barycentric position of the observer (au)
*     SC    d(3)     observer to star coordinate direction (unit vector)
*
*  Returned:
*     SN    d(3)     observer to deflected star (unit vector)
*
*  1) The array B contains N entries, one for each body to be
*     considered.  If N = 0, no gravitational light deflection will be
*     applied, not even for the Sun.
*
*  2) The array B should include an entry for the Sun as well as for any
*     planet or other body to be taken into account.  The entries should
*     be in the order in which the light passes the body.
*
*  3) In the entry in the B array for body I, the mass parameter B(1,I)
*     can, as required, be adjusted in order to allow for such effects
*     as quadrupole field.
*
*  4) The deflection limiter parameter B(2,I) is phi^2/2, where phi is
*     the angular separation (in radians) between star and body at which
*     limiting is applied.  As phi shrinks below the chosen threshold,
*     the deflection is artificially reduced, reaching zero for phi = 0.
*     Example values suitable for a terrestrial observer, together with
*     masses, are as follows:
*
*        body I     B(1,I)         B(2,I)
*
*        Sun        1D0            6D-6
*        Jupiter    0.00095435D0   3D-9
*        Saturn     0.00028574D0   3D-10
*
*  5) For cases where the starlight passes the body before reaching the
*     observer, the body is placed back along its barycentric track by
*     the light time from that point to the observer.  For cases where
*     the body is "behind" the observer no such shift is applied.  If
*     a different treatment is preferred, the user has the option of
*     instead using the iau_LD routine.  Similarly, iau_LD can be used
*     for cases where the source is nearby, not a star.
*
*  6) The returned vector SN is not normalized, but the consequential
*     departure from unit magnitude is always negligible.
*
*  7) For efficiency, validation is omitted.  The supplied masses must
*     be greater than zero, the position and velocity vectors must be
*     right, and the deflection limiter greater than zero.
*
*  Reference:
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013), Section 7.2.4.
```

```
*
*  Called:
*     iau_CP       copy p-vector
*     iau_PDP      scalar product of two p-vectors
*     iau_PMP      p-vector minus p-vector
*     iau_PPSP     p-vector plus scaled p-vector
*     iau_PN       decompose p-vector into modulus and direction
*     iau_LD       light deflection by a solar-system body
*
*-
```

```
      SUBROUTINE iau_LDSUN ( P, E, EM, P1 )
*+
*  - - - - - - - - - -
*   i a u _ L D S U N
*  - - - - - - - - - -
*
*  Deflection of starlight by the Sun.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     P        d(3)   direction from observer to star (unit vector)
*     E        d(3)   direction from Sun to observer (unit vector)
*     EM       d      distance from Sun to observer (au)
*
*  Returned:
*     P1       d(3)   observer to deflected star (unit vector)
*
*  Notes:
*
*  1) The source is presumed to be sufficiently distant that its
*     directions seen from the Sun and the observer are essentially
*     the same.
*
*  2) The deflection is restrained when the angle between the star and
*     the center of the Sun is less than a threshold value, falling to
*     zero deflection for zero separation.  The chosen threshold value
*     is within the solar limb for all solar-system applications, and
*     is about 5 arcminutes for the case of a terrestrial observer.
*
*  Called:
*     iau_LD       light deflection by a solar-system body
*
*-
```

```
      SUBROUTINE iau_LTECEQ ( EPJ, DL, DB, DR, DD )
*+
*  - - - - - - - - - - -
*   i a u _ L T E C E Q
*  - - - - - - - - - - -
*
*  Transformation from ecliptic coordinates (mean equinox and ecliptic
*  of date) to ICRS RA,Dec, using a long-term precession model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*     DL,DB     d        ecliptic longitude and latitude (radians)
*
*  Returned:
*     DR,DD     d        ICRS right ascension and declination (radians)
*
*  1) No assumptions are made about whether the coordinates represent
*     starlight and embody astrometric effects such as parallax or
*     aberration.
*
*  2) The transformation is approximately that from ecliptic longitude
*     and latitude (mean equinox and ecliptic of date) to mean J2000.0
*     right ascension and declination, with only frame bias (always less
*     than 25 mas) to disturb this classical picture.
*
*  3) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_LTECM    J2000.0 to ecliptic rotation matrix, long term
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_C2S      unit vector to spherical coordinates
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTECM ( EPJ, RM )
*+
*  - - - - - - - - - -
*   i a u _ L T E C M
*  - - - - - - - - - -
*
*  ICRS equatorial to ecliptic rotation matrix, long-term.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*
*  Returned:
*     RM        d(3,3)   ICRS to ecliptic rotation matrix
*
*  Notes:
*
*  1) The matrix is in the sense
*
*        E_ep = RM x P_ICRS,
*
*     where P_ICRS is a vector with respect to ICRS right ascension
*     and declination axes and E_ep is the same vector with respect to
*     the (inertial) ecliptic and equinox of epoch EPJ.
*
*  2) P_ICRS is a free vector, merely a direction, typically of unit
*     magnitude, and not bound to any particular spatial origin, such as
*     the Earth, Sun or SSB.  No assumptions are made about whether it
*     represents starlight and embodies astrometric effects such as
*     parallax or aberration.  The transformation is approximately that
*     between mean J2000.0 right ascension and declination and ecliptic
*     longitude and latitude, with only frame bias (always less than
*     25 mas) to disturb this classical picture.
*
*  3) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  Called:
*     iau_LTPEQU   equator pole, long term
*     iau_LTPECL   ecliptic pole, long term
*     iau_PXP      vector product
*     iau_PN       normalize vector
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTEQEC ( EPJ, DR, DD, DL, DB )
*+
*  - - - - - - - - - - -
*   i a u _ L T E Q E C
*  - - - - - - - - - - -
*
*  Transformation from ICRS equatorial coordinates to ecliptic
*  coordinates (mean equinox and ecliptic of date), using a long-term
*  precession model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*     DR,DD     d        ICRS right ascension and declination (radians)
*
*  Returned:
*     DL,DB     d        ecliptic longitude and latitude (radians)
*
*  1) No assumptions are made about whether the coordinates represent
*     starlight and embody astrometric effects such as parallax or
*     aberration.
*
*  2) The transformation is approximately that from mean J2000.0 right
*     ascension and declination to ecliptic longitude and latitude
*     (mean equinox and ecliptic of date), with only frame bias (always
*     less than 25 mas) to disturb this classical picture.
*
*  3) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_LTECM    J2000.0 to ecliptic rotation matrix, long term
*     iau_RXP      product of r-matrix and p-vector
*     iau_C2S      unit vector to spherical coordinates
*     iau_ANP      normalize angle into range 0 to 2pi
*     iau_ANPM     normalize angle into range +/- pi
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTP ( EPJ, RP )
*+
*  - - - - - - - -
*   i a u _ L T P
*  - - - - - - - -
*
*  Long-term precession matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*
*  Returned:
*     RP        d(3,3)   precession matrix, J2000.0 to date
*
*  Notes:
*
*  1) The matrix is in the sense
*
*        P_date = RP x P_J2000,
*
*     where P_J2000 is a vector with respect to the J2000.0 mean equator
*     and equinox and P_date is the same vector with respect to the
*     equator and equinox of epoch EPJ.
*
*  2) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  Called:
*     iau_LTPEQU   equator pole, long term
*     iau_LTPECL   ecliptic pole, long term
*     iau_PXP      vector product
*     iau_PN       normalize vector
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTPB ( EPJ, RPB )
*+
*  - - - - - - - - -
*   i a u _ L T P B
*  - - - - - - - - -
*
*  Long-term precession matrix, including ICRS frame bias.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*
*  Returned:
*     RPB       d        precession-bias matrix, J2000.0 to date
*
*  Notes:
*
*  1) The matrix is in the sense
*
*        P_date = RPB x P_ICRS,
*
*     where P_J2000 is a vector in the International Celestial Reference
*     System, and P_date is the vector with respect to the Celestial
*     Intermediate Reference System at that date but with nutation
*     neglected.
*
*  2) A first order frame bias formulation is used, of sub-
*     microarcsecond accuracy compared with a full 3D rotation.
*
*  3) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  Called:
*     iau_LTP      precession matrix, long term
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTPECL ( EPJ, VEC )
*+
*  - - - - - - - - - - -
*   i a u _ L T P E C L
*  - - - - - - - - - - -
*
*  Long-term precession of the ecliptic.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*
*  Returned:
*     VEC       d(3)     ecliptic pole unit vector
*
*  Notes:
*
*  1) The returned vector is with respect to the J2000.0 mean equator
*     and equinox.
*
*  2) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_LTPEQU ( EPJ, VEQ )
*+
*  - - - - - - - - - - -
*   i a u _ L T P E Q U
*  - - - - - - - - - - -
*
*  Long-term precession of the equator.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPJ       d        Julian epoch (TT)
*
*  Returned:
*     VEQ       d(3)     equator pole unit vector
*
*  Notes:
*
*  1) The returned vector is with respect to the J2000.0 mean equator
*     and equinox.
*
*  2) The Vondrak et al. (2011, 2012) 400 millennia precession model
*     agrees with the IAU 2006 precession at J2000.0 and stays within
*     100 microarcseconds during the 20th and 21st centuries.  It is
*     accurate to a few arcseconds throughout the historical period,
*     worsening to a few tenths of a degree at the end of the
*     +/- 200,000 year time span.
*
*  References:
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2011, New precession
*     expressions, valid for long time intervals, Astron.Astrophys. 534,
*     A22
*
*     Vondrak, J., Capitaine, N. and Wallace, P., 2012, New precession
*     expressions, valid for long time intervals (Corrigendum),
*     Astron.Astrophys. 541, C1
*
*-
```

```
      SUBROUTINE iau_NUM00A ( DATE1, DATE2, RMATN )
*+
*  - - - - - - - - - - -
*   i a u _ N U M 0 0 A
*  - - - - - - - - - - -
*
*  Form the matrix of nutation for a given date, IAU 2000A model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RMATN        d(3,3)    nutation matrix
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  3) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_NUM00B routine.
*
*  Called:
*     iau_PN00A    bias/precession/nutation, IAU 2000A
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.222-3 (p114).
*
*-
```

```
      SUBROUTINE iau_NUM00B ( DATE1, DATE2, RMATN )
*+
*  - - - - - - - - - - -
*   i a u _ N U M 0 0 B
*  - - - - - - - - - - -
*
*  Form the matrix of nutation for a given date, IAU 2000B model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RMATN       d(3,3)   nutation matrix
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  3) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_NUM00A routine.
*
*  Called:
*     iau_PN00B    bias/precession/nutation, IAU 2000B
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.222-3 (p114).
*
*-
```

```
      SUBROUTINE iau_NUM06A ( DATE1, DATE2, RMATN )
*+
*  - - - - - - - - - - - -
*   i a u _ N U M 0 6 A
*  - - - - - - - - - - - -
*
*  Form the matrix of nutation for a given date, IAU 2006/2000A model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RMATN        d(3,3)    nutation matrix
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  Called:
*     iau_OBL06    mean obliquity, IAU 2006
*     iau_NUT06A   nutation, IAU 2006/2000A
*     iau_NUMAT    form nutation matrix
*
*  References:
*
*     Capitaine, N., Wallace, P.T. & Chapront, J., 2005, Astron.
*     Astrophys. 432, 355
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_NUMAT ( EPSA, DPSI, DEPS, RMATN )
*+
*  - - - - - - - - - -
*   i a u _ N U M A T
*  - - - - - - - - - -
*
*  Form the matrix of nutation.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPSA          d       mean obliquity of date (Note 1)
*     DPSI,DEPS     d       nutation (Note 2)
*
*  Returned:
*     RMATN       d(3,3)    nutation matrix (Note 3)
*
*  Notes:
*
*
*  1) The supplied mean obliquity EPSA, must be consistent with the
*     precession-nutation models from which DPSI and DEPS were obtained.
*
*  2) The caller is responsible for providing the nutation components;
*     they are in longitude and obliquity, in radians and are with
*     respect to the equinox and ecliptic of date.
*
*  3) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RX       rotate around X-axis
*     iau_RZ       rotate around Z-axis
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.222-3 (p114).
*
*-
```

```
      SUBROUTINE iau_NUT00A ( DATE1, DATE2, DPSI, DEPS )
*+
*  − − − − − − − − − − −
*   i a u _ N U T 0 0 A
*  − − − − − − − − − − −
*
*  Nutation, IAU 2000A model (MHB2000 luni−solar and planetary nutation
*  with free core nutation omitted).
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2−part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d    nutation, luni−solar + planetary (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      −1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components in longitude and obliquity are in radians
*     and with respect to the equinox and ecliptic of date.  The
*     obliquity at J2000.0 is assumed to be the Lieske et al. (1977)
*     value of 84381.448 arcsec.
*
*     Both the luni−solar and planetary nutations are included.  The
*     latter are due to direct planetary nutations and the perturbations
*     of the lunar and terrestrial orbits.
*
*  3) The routine computes the MHB2000 nutation series with the
*     associated corrections for planetary nutations.  It is an
*     implementation of the nutation part of the IAU 2000A precession−
*     nutation model, formally adopted by the IAU General Assembly in
*     2000, namely MHB2000 (Mathews et al. 2002), but with the free core
*     nutation (FCN − see Note 4) omitted.
*
*  4) The full MHB2000 model also contains contributions to the
*     nutations in longitude and obliquity due to the free−excitation of
*     the free−core−nutation during the period 1979−2000.  These FCN
*     terms, which are time−dependent and unpredictable, are NOT
*     included in the present routine and, if required, must be
*     independently computed.  With the FCN corrections included, the
*     present routine delivers a pole which is at current epochs
*     accurate to a few hundred microarcseconds.  The omission of FCN
*     introduces further errors of about that size.
*
*  5) The present routine provides classical nutation.  The MHB2000
*     algorithm, from which it is adapted, deals also with (i) the
*     offsets between the GCRS and mean poles and (ii) the adjustments
*     in longitude and obliquity due to the changed precession rates.
*     These additional functions, namely frame bias and precession
```

```
*     adjustments, are supported by the SOFA routines iau_BI00 and
*     iau_PR00.
*
*  6) The MHB2000 algorithm also provides "total" nutations, comprising
*     the arithmetic sum of the frame bias, precession adjustments,
*     luni-solar nutation and planetary nutation.  These total nutations
*     can be used in combination with an existing IAU 1976 precession
*     implementation, such as iau_PMAT76, to deliver GCRS-to-true
*     predictions of sub-mas accuracy at current epochs.  However, there
*     are three shortcomings in the MHB2000 model that must be taken
*     into account if more accurate or definitive results are required
*     (see Wallace 2002):
*
*       (i) The MHB2000 total nutations are simply arithmetic sums,
*           yet in reality the various components are successive Euler
*           rotations.  This slight lack of rigor leads to cross terms
*           that exceed 1 mas after a century.  The rigorous procedure
*           is to form the GCRS-to-true rotation matrix by applying the
*           bias, precession and nutation in that order.
*
*      (ii) Although the precession adjustments are stated to be with
*           respect to Lieske et al. (1977), the MHB2000 model does
*           not specify which set of Euler angles are to be used and
*           how the adjustments are to be applied.  The most literal and
*           straightforward procedure is to adopt the 4-rotation
*           epsilon_0, psi_A, omega_A, xi_A option, and to add DPSIPR to
*           psi_A and DEPSPR to both omega_A and eps_A.
*
*     (iii) The MHB2000 model predates the determination by Chapront
*           et al. (2002) of a 14.6 mas displacement between the J2000.0
*           mean equinox and the origin of the ICRS frame.  It should,
*           however, be noted that neglecting this displacement when
*           calculating star coordinates does not lead to a 14.6 mas
*           change in right ascension, only a small second-order
*           distortion in the pattern of the precession-nutation effect.
*
*     For these reasons, the SOFA routines do not generate the "total
*     nutations" directly, though they can of course easily be generated
*     by calling iau_BI00, iau_PR00 and the present routine and adding
*     the results.
*
*  7) The MHB2000 model contains 41 instances where the same frequency
*     appears multiple times, of which 38 are duplicates and three are
*     triplicates.  To keep the present code close to the original MHB
*     algorithm, this small inefficiency has not been corrected.
*
*  Called:
*     iau_FAL03    mean anomaly of the Moon
*     iau_FAF03    mean argument of the latitude of the Moon
*     iau_FAOM03   mean longitude of the Moon's ascending node
*     iau_FAME03   mean longitude of Mercury
*     iau_FAVE03   mean longitude of Venus
*     iau_FAE03    mean longitude of Earth
*     iau_FAMA03   mean longitude of Mars
*     iau_FAJU03   mean longitude of Jupiter
*     iau_FASA03   mean longitude of Saturn
*     iau_FAUR03   mean longitude of Uranus
*     iau_FAPA03   general accumulated precession in longitude
*
*  References:
*
*     Chapront, J., Chapront-Touze, M. & Francou, G. 2002,
*     Astron.Astrophys. 387, 700
*
*     Lieske, J.H., Lederle, T., Fricke, W. & Morando, B. 1977,
*     Astron.Astrophys. 58, 1-16
*
*     Mathews, P.M., Herring, T.A., Buffet, B.A. 2002, J.Geophys.Res.
*     107, B4.  The MHB_2000 code itself was obtained on 9th September
*     2002 from ftp//maia.usno.navy.mil/conv2000/chapter5/IAU2000A.
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J. 1994, Astron.Astrophys. 282, 663-683
```

```
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M. 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*     Wallace, P.T., "Software for Implementing the IAU 2000
*     Resolutions", in IERS Workshop 5.1 (2002)
*
*-
```

```
      SUBROUTINE iau_NUT00B ( DATE1, DATE2, DPSI, DEPS )
*+
*  - - - - - - - - - - -
*   i a u _ N U T 0 0 B
*  - - - - - - - - - - -
*
*  Nutation, IAU 2000B model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2   d   TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d   nutation, luni-solar + planetary (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.
*
*  2) The nutation components in longitude and obliquity are in radians
*     and with respect to the equinox and ecliptic of date.  The
*     obliquity at J2000.0 is assumed to be the Lieske et al. (1977)
*     value of 84381.448 arcsec.  (The errors that result from using
*     this routine with the IAU 2006 value of 84381.406 arcsec can be
*     neglected.)
*
*     The nutation model consists only of luni-solar terms, but includes
*     also a fixed offset which compensates for certain long-period
*     planetary terms (Note 7).
*
*  3) This routine is an implementation of the IAU 2000B abridged
*     nutation model formally adopted by the IAU General Assembly in
*     2000.  The routine computes the MHB_2000_SHORT luni-solar nutation
*     series (Luzum 2001), but without the associated corrections for
*     the precession rate adjustments and the offset between the GCRS
*     and J2000.0 mean poles.
*
*  4) The full IAU 2000A (MHB2000) nutation model contains nearly 1400
*     terms.  The IAU 2000B model (McCarthy & Luzum 2003) contains only
*     77 terms, plus additional simplifications, yet still delivers
*     results of 1 mas accuracy at present epochs.  This combination of
*     accuracy and size makes the IAU 2000B abridged nutation model
*     suitable for most practical applications.
*
*     The routine delivers a pole accurate to 1 mas from 1900 to 2100
*     (usually better than 1 mas, very occasionally just outside 1 mas).
*     The full IAU 2000A model, which is implemented in the routine
*     iau_NUT00A (q.v.), delivers considerably greater accuracy at
*     current epochs;  however, to realize this improved accuracy,
*     corrections for the essentially unpredictable free-core-nutation
*     (FCN) must also be included.
```

```
*
*  5) The present routine provides classical nutation.  The
*     MHB_2000_SHORT algorithm, from which it is adapted, deals also
*     with (i) the offsets between the GCRS and mean poles and (ii) the
*     adjustments in longitude and obliquity due to the changed
*     precession rates.  These additional functions, namely frame bias
*     and precession adjustments, are supported by the SOFA routines
*     iau_BI00 and iau_PR00.
*
*  6) The MHB_2000_SHORT algorithm also provides "total" nutations,
*     comprising the arithmetic sum of the frame bias, precession
*     adjustments, and nutation (luni-solar + planetary).  These total
*     nutations can be used in combination with an existing IAU 1976
*     precession implementation, such as iau_PMAT76, to deliver GCRS-to-
*     true predictions of mas accuracy at current epochs.  However, for
*     symmetry with the iau_NUT00A routine (q.v. for the reasons), the
*     SOFA routines do not generate the "total nutations" directly.
*     Should they be required, they could of course easily be generated
*     by calling iau_BI00, iau_PR00 and the present routine and adding
*     the results.
*
*  7) The IAU 2000B model includes "planetary bias" terms that are fixed
*     in size but compensate for long-period nutations.  The amplitudes
*     quoted in McCarthy & Luzum (2003), namely Dpsi = -1.5835 mas and
*     Depsilon = +1.6339 mas, are optimized for the "total nutations"
*     method described in Note 6.  The Luzum (2001) values used in this
*     SOFA implementation, namely -0.135 mas and +0.388 mas, are
*     optimized for the "rigorous" method, where frame bias, precession
*     and nutation are applied separately and in that order.  During the
*     interval 1995-2050, the SOFA implementation delivers a maximum
*     error of 1.001 mas (not including FCN).
*
*  References:
*
*     Lieske, J.H., Lederle, T., Fricke, W., Morando, B., "Expressions
*     for the precession quantities based upon the IAU /1976/ system of
*     astronomical constants", Astron.Astrophys. 58, 1-2, 1-16. (1977)
*
*     Luzum, B., private communication, 2001 (Fortran code
*     MHB_2000_SHORT)
*
*     McCarthy, D.D. & Luzum, B.J., "An abridged model of the
*     precession-nutation of the celestial pole", Cel.Mech.Dyn.Astron.
*     85, 37-49 (2003)
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J., Astron.Astrophys. 282, 663-683 (1994)
*
*-
```

```
      SUBROUTINE iau_NUT06A ( DATE1, DATE2, DPSI, DEPS )
*+
*  - - - - - - - - - - -
*   i a u _ N U T 0 6 A
*  - - - - - - - - - - -
*
*  IAU 2000A nutation with adjustments to match the IAU 2006 precession.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d    nutation, luni-solar + planetary (Note 2)
*
*  Status:  canonical model.
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components in longitude and obliquity are in radians
*     and with respect to the mean equinox and ecliptic of date,
*     IAU 2006 precession model (Hilton et al. 2006, Capitaine et al.
*     2005).
*
*  3) The routine first computes the IAU 2000A nutation, then applies
*     adjustments for (i) the consequences of the change in obliquity
*     from the IAU 1980 ecliptic to the IAU 2006 ecliptic and (ii) the
*     secular variation in the Earth's dynamical form factor J2.
*
*  4) The present routine provides classical nutation, complementing
*     the IAU 2000 frame bias and IAU 2006 precession.  It delivers a
*     pole which is at current epochs accurate to a few tens of
*     microarcseconds, apart from the free core nutation.
*
*  Called:
*     iau_NUT00A   nutation, IAU 2000A
*
*  Reference:
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_NUT80 ( DATE1, DATE2, DPSI, DEPS )
*+
*  - - - - - - - - - -
*   i a u _ N U T 8 0
*  - - - - - - - - - -
*
*  Nutation, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2     d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI            d      nutation in longitude (radians)
*     DEPS            d      nutation in obliquity (radians)
*
*  Notes:
*
*  1) The DATE DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components are with respect to the ecliptic of
*     date.
*
*  Called:
*     iau_ANPM     normalize angle into range +/- pi
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.222 (p111).
*
*-
```

```
      SUBROUTINE iau_NUTM80 ( DATE1, DATE2, RMATN )
*+
*  - - - - - - - - - - -
*   i a u _ N U T M 8 0
*  - - - - - - - - - - -
*
*  Form the matrix of nutation for a given date, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2      d       TDB date (Note 1)
*
*  Returned:
*     RMATN          d(3,3)    nutation matrix
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  Called:
*     iau_NUT80    nutation, IAU 1980
*     iau_OBL80    mean obliquity, IAU 1980
*     iau_NUMAT    form nutation matrix
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_OBL06 ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ O B L 0 6
*  - - - - - - - - - -
*
*  Mean obliquity of the ecliptic, IAU 2006 precession model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d     TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_OBL06      d     obliquity of the ecliptic (radians, Note 2)
*
*  Notes:
*
*  1) The date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DATE1         DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result is the angle between the ecliptic and mean equator of
*     date DATE1+DATE2.
*
*  Reference:
*
*     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_OBL80 ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ O B L 8 0
*  - - - - - - - - - -
*
*  Mean obliquity of the ecliptic, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2     d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_OBL80       d      obliquity of the ecliptic (radians, Note 2)
*
*  Notes:
*
*  1) The date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DATE1         DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result is the angle between the ecliptic and mean equator of
*     date DATE1+DATE2.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Expression 3.222-1 (p114).
*
*-
```

```
      SUBROUTINE iau_P06E ( DATE1, DATE2,
     :                      EPS0, PSIA, OMA, BPA, BQA, PIA, BPIA,
     :                      EPSA, CHIA, ZA, ZETAA, THETAA, PA,
     :                      GAM, PHI, PSI )
*+
*  - - - - - - - - -
*   i a u _ P 0 6 E
*  - - - - - - - - -
*
*  Precession angles, IAU 2006, equinox based.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical models.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned (see Note 2):
*     EPS0          d    epsilon_0
*     PSIA          d    psi_A
*     OMA           d    omega_A
*     BPA           d    P_A
*     BQA           d    Q_A
*     PIA           d    pi_A
*     BPIA          d    Pi_A
*     EPSA          d    obliquity epsilon_A
*     CHIA          d    chi_A
*     ZA            d    z_A
*     ZETAA         d    zeta_A
*     THETAA        d    theta_A
*     PA            d    p_A
*     GAM           d    F-W angle gamma_J2000
*     PHI           d    F-W angle phi_J2000
*     PSI           d    F-W angle psi_J2000
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) This routine returns the set of equinox based angles for the
*     Capitaine et al. "P03" precession theory, adopted by the IAU in
*     2006.  The angles are set out in Table 1 of Hilton et al. (2006):
*
*      EPS0   epsilon_0   obliquity at J2000.0
*      PSIA   psi_A       luni-solar precession
*      OMA    omega_A     inclination of equator wrt J2000.0 ecliptic
*      BPA    P_A         ecliptic pole x, J2000.0 ecliptic triad
*      BQA    Q_A         ecliptic pole -y, J2000.0 ecliptic triad
*      PIA    pi_A        angle between moving and J2000.0 ecliptics
*      BPIA   Pi_A        longitude of ascending node of the ecliptic
*      EPSA   epsilon_A   obliquity of the ecliptic
*      CHIA   chi_A       planetary precession
*      ZA     z_A         equatorial precession: -3rd 323 Euler angle
```

```
*      ZETAA  zeta_A      equatorial precession: −1st 323 Euler angle
*      THETAA theta_A     equatorial precession: 2nd 323 Euler angle
*      PA     p_A         general precession
*      GAM    gamma_J2000 J2000.0 RA difference of ecliptic poles
*      PHI    phi_J2000   J2000.0 codeclination of ecliptic pole
*      PSI    psi_J2000   longitude difference of equator poles, J2000.0
*
*     The returned values are all radians.
*
*  3) Hilton et al. (2006) Table 1 also contains angles that depend on
*     models distinct from the P03 precession theory itself, namely the
*     IAU 2000A frame bias and nutation.  The quoted polynomials are
*     used in other SOFA routines:
*
*     . iau_XY06 contains the polynomial parts of the X and Y series.
*
*     . iau_S06 contains the polynomial part of the s+XY/2 series.
*
*     . iau_PFW06 implements the series for the Fukushima−Williams
*       angles that are with respect to the GCRS pole (i.e. the variants
*       that include frame bias).
*
*  4) The IAU resolution stipulated that the choice of parameterization
*     was left to the user, and so an IAU compliant precession
*     implementation can be constructed using various combinations of
*     the angles returned by the present routine.
*
*  5) The parameterization used by SOFA is the version of the Fukushima−
*     Williams angles that refers directly to the GCRS pole.  These
*     angles may be calculated by calling the routine iau_PFW06.  SOFA
*     also supports the direct computation of the CIP GCRS X,Y by
*     series, available by calling iau_XY06.
*
*  6) The agreement between the different parameterizations is at the
*     1 microarcsecond level in the present era.
*
*  7) When constructing a precession formulation that refers to the GCRS
*     pole rather than the dynamical pole, it may (depending on the
*     choice of angles) be necessary to introduce the frame bias
*     explicitly.
*
*  Reference:
*
*     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
*
*  Called:
*     iau_OBL06    mean obliquity, IAU 2006
*
*−
```

```
      SUBROUTINE iau_P2PV ( P, PV )
*+
*  - - - - - - - - -
*   i a u _ P 2 P V
*  - - - - - - - - -
*
*  Extend a p-vector to a pv-vector by appending a zero velocity.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)      p-vector
*
*  Returned:
*     PV       d(3,2)    pv-vector
*
*  Called:
*     iau_CP       copy p-vector
*     iau_ZP       zero p-vector
*
*-
```

```
      SUBROUTINE iau_P2S ( P, THETA, PHI, R )
*+
*  - - - - - - - -
*   i a u _ P 2 S
*  - - - - - - - -
*
*  P-vector to spherical polar coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)      p-vector
*
*  Returned:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*     R        d         radial distance
*
*  Notes:
*
*  1) If P is null, zero THETA, PHI and R are returned.
*
*  2) At either pole, zero THETA is returned.
*
*  Called:
*     iau_C2S      p-vector to spherical
*     iau_PM       modulus of p-vector
*
*-
```

```
      SUBROUTINE iau_PAP ( A, B, THETA )
*+
*  - - - - - - - -
*   i a u _ P A P
*  - - - - - - - -
*
*  Position-angle from two p-vectors.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      direction of reference point
*     B        d(3)      direction of point whose PA is required
*
*  Returned:
*     THETA    d         position angle of B with respect to A (radians)
*
*  Notes:
*
*  1) The result is the position angle, in radians, of direction B with
*     respect to direction A.  It is in the range -pi to +pi.  The sense
*     is such that if B is a small distance "north" of A the position
*     angle is approximately zero, and if B is a small distance "east" of
*     A the position angle is approximately +pi/2.
*
*  2) A and B need not be unit vectors.
*
*  3) Zero is returned if the two directions are the same or if either
*     vector is null.
*
*  4) If A is at a pole, the result is ill-defined.
*
*  Called:
*     iau_PN       decompose p-vector into modulus and direction
*     iau_PM       modulus of p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PMP      p-vector minus p-vector
*     iau_PDP      scalar product of two p-vectors
*
*-
```

```
      SUBROUTINE iau_PAS ( AL, AP, BL, BP, THETA )
*+
*  - - - - - - - -
*   i a u _ P A S
*  - - - - - - - -
*
*  Position-angle from spherical coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     AL       d       longitude of point A (e.g. RA) in radians
*     AP       d       latitude of point A (e.g. Dec) in radians
*     BL       d       longitude of point B
*     BP       d       latitude of point B
*
*  Returned:
*     THETA    d       position angle of B with respect to A
*
*  Notes:
*
*  1) The result is the bearing (position angle), in radians, of point
*     B with respect to point A.  It is in the range -pi to +pi.  The
*     sense is such that if B is a small distance "east" of point A,
*     the bearing is approximately +pi/2.
*
*  2) Zero is returned if the two points are coincident.
*
*-
```

```
      SUBROUTINE iau_PB06 ( DATE1, DATE2, BZETA, BZ, BTHETA )
*+
*  - - - - - - - - -
*   i a u _ P B 0 6
*  - - - - - - - - -
*
*  This routine forms three Euler angles which implement general
*  precession from epoch J2000.0, using the IAU 2006 model.  Frame
*  bias (the offset between ICRS and mean J2000.0) is included.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     BZETA         d    1st rotation: radians clockwise around z
*     BZ            d    3rd rotation: radians clockwise around z
*     BTHETA        d    2nd rotation: radians counterclockwise around y
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the arguments DATE1 and DATE2.  For
*     example, JD(TT)=2450123.7 could be expressed in any of these
*     ways, among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The traditional accumulated precession angles zeta_A, z_A, theta_A
*     cannot be obtained in the usual way, namely through polynomial
*     expressions, because of the frame bias.  The latter means that two
*     of the angles undergo rapid changes near this date.  They are
*     instead the results of decomposing the precession-bias matrix
*     obtained by using the Fukushima-Williams method, which does not
*     suffer from the problem.  The decomposition returns values which
*     can be used in the conventional formulation and which include
*     frame bias.
*
*  3) The three angles are returned in the conventional order, which
*     is not the same as the order of the corresponding Euler rotations.
*     The precession-bias matrix is R_3(-z) x R_2(+theta) x R_3(-zeta).
*
*  4) Should zeta_A, z_A, theta_A angles be required that do not contain
*     frame bias, they are available by calling the SOFA routine
*     iau_P06E.
*
*  Called:
*     iau_PMAT06   PB matrix, IAU 2006
*     iau_RZ       rotate around Z-axis
*
*-
```

```
      SUBROUTINE iau_PDP ( A, B, ADB )
*+
*  - - - - - - - -
*   i a u _ P D P
*  - - - - - - - -
*
*  p-vector inner (=scalar=dot) product.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector
*     B        d(3)      second p-vector
*
*  Returned:
*     ADB      d         A . B
*
*-
```

```
      SUBROUTINE iau_PFW06 ( DATE1, DATE2, GAMB, PHIB, PSIB, EPSA )
*+
*  - - - - - - - - - -
*   i a u _ P F W 0 6
*  - - - - - - - - - -
*
*  Precession angles, IAU 2006 (Fukushima-Williams 4-angle formulation).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     GAMB          d    F-W angle gamma_bar (radians)
*     PHIB          d    F-W angle phi_bar (radians)
*     PSIB          d    F-W angle psi_bar (radians)
*     EPSA          d    F-W angle epsilon_A (radians)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) Naming the following points:
*
*           e = J2000.0 ecliptic pole,
*           p = GCRS pole,
*           E = mean ecliptic pole of date,
*     and   P = mean pole of date,
*
*     the four Fukushima-Williams angles are as follows:
*
*        GAMB = gamma_bar = epE
*        PHIB = phi_bar = pE
*        PSIB = psi_bar = pEP
*        EPSA = epsilon_A = EP
*
*  3) The matrix representing the combined effects of frame bias and
*     precession is:
*
*        PxB = R_1(-EPSA).R_3(-PSIB).R_1(PHIB).R_3(GAMB)
*
*  4) The matrix representing the combined effects of frame bias,
*     precession and nutation is simply:
*
*        NxPxB = R_1(-EPSA-dE).R_3(-PSIB-dP).R_1(PHIB).R_3(GAMB)
*
*     where dP and dE are the nutation components with respect to the
*     ecliptic of date.
*
*  Reference:
*
```

```
*     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
*
*  Called:
*     iau_OBL06    mean obliquity, IAU 2006
*
*-
```

```
      SUBROUTINE iau_PLAN94 ( DATE1, DATE2, NP, PV, J )
*+
*  - - - - - - - - - - -
*   i a u _ P L A N 9 4
*  - - - - - - - - - - -
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Approximate heliocentric position and velocity of a nominated major
*  planet:  Mercury, Venus, EMB, Mars, Jupiter, Saturn, Uranus or
*  Neptune (but not the Earth itself).
*
*  Given:
*     DATE1    d       TDB date part A (Note 1)
*     DATE2    d       TDB date part B (Note 1)
*     NP       i       planet (1=Mercury, 2=Venus, 3=EMB ... 8=Neptune)
*
*  Returned:
*     PV       d(3,2)  planet pos,vel (heliocentric, J2000.0, au, au/d)
*     J        i       status: -1 = illegal NP (outside 1-8)
*                               0 = OK
*                              +1 = warning: date outside 1000-3000 AD
*                              +2 = warning: solution failed to converge
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     The limited accuracy of the present algorithm is such that any
*     of the methods is satisfactory.
*
*  2) If an NP value outside the range 1-8 is supplied, an error
*     status (J = -1) is returned and the PV vector set to zeroes.
*
*  3) For NP=3 the result is for the Earth-Moon Barycenter.  To
*     obtain the heliocentric position and velocity of the Earth,
*     use instead the SOFA routine iau_EPV00.
*
*  4) On successful return, the array PV contains the following:
*
*        PV(1,1)  x       }
*        PV(2,1)  y       } heliocentric position, au
*        PV(3,1)  z       }
*
*        PV(1,2)  xdot    }
*        PV(2,2)  ydot    } heliocentric velocity, au/d
*        PV(3,2)  zdot    }
*
*     The reference frame is equatorial and is with respect to the
*     mean equator and equinox of epoch J2000.0.
*
*  5) The algorithm is due to J.L. Simon, P. Bretagnon, J. Chapront,
*     M. Chapront-Touze, G. Francou and J. Laskar (Bureau des
```

```
*     Longitudes, Paris, France).  From comparisons with JPL
*     ephemeris DE102, they quote the following maximum errors
*     over the interval 1800−2050:
*
*                     L (arcsec)    B (arcsec)      R (km)
*
*        Mercury          4             1             300
*        Venus            5             1             800
*        EMB              6             1            1000
*        Mars            17             1            7700
*        Jupiter         71             5           76000
*        Saturn          81            13          267000
*        Uranus          86             7          712000
*        Neptune         11             1          253000
*
*     Over the interval 1000−3000, they report that the accuracy is no
*     worse than 1.5 times that over 1800−2050.  Outside 1000−3000 the
*     accuracy declines.
*
*     Comparisons of the present routine with the JPL DE200 ephemeris
*     give the following RMS errors over the interval 1960−2025:
*
*                      position (km)     velocity (m/s)
*
*        Mercury            334               0.437
*        Venus             1060               0.855
*        EMB               2010               0.815
*        Mars              7690               1.98
*        Jupiter          71700               7.70
*        Saturn          199000              19.4
*        Uranus          564000              16.4
*        Neptune         158000              14.4
*
*     Comparisons against DE200 over the interval 1800−2100 gave the
*     following maximum absolute differences.  (The results using
*     DE406 were essentially the same.)
*
*                   L (arcsec)   B (arcsec)     R (km)   Rdot (m/s)
*
*        Mercury       7            1            500       0.7
*        Venus         7            1           1100       0.9
*        EMB           9            1           1300       1.0
*        Mars         26            1           9000       2.5
*        Jupiter      78            6          82000       8.2
*        Saturn       87           14         263000      24.6
*        Uranus       86            7         661000      27.4
*        Neptune      11            2         248000      21.4
*
*  6) The present SOFA re−implementation of the original Simon et al.
*     Fortran code differs from the original in the following respects:
*
*       *  The date is supplied in two parts.
*
*       *  The result is returned only in equatorial Cartesian form;
*          the ecliptic longitude, latitude and radius vector are not
*          returned.
*
*       *  The result is in the J2000.0 equatorial frame, not ecliptic.
*
*       *  More is done in−line:  there are fewer calls to other
*          routines.
*
*       *  Different error/warning status values are used.
*
*       *  A different Kepler’s−equation−solver is used (avoiding
*          use of COMPLEX*16).
*
*       *  Polynomials in T are nested to minimize rounding errors.
*
*       *  Explicit double−precision constants are used to avoid
*          mixed−mode expressions.
*
*       *  There are other, cosmetic, changes to comply with SOFA
```

```
*          style conventions.
*
*     None of the above changes affects the result significantly.
*
*  7) The returned status, J, indicates the most serious condition
*     encountered during execution of the routine.  Illegal NP is
*     considered the most serious, overriding failure to converge,
*     which in turn takes precedence over the remote epoch warning.
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  Reference:  Simon, J.L, Bretagnon, P., Chapront, J.,
*              Chapront-Touze, M., Francou, G., and Laskar, J.,
*              Astron.Astrophys., 282, 663 (1994).
*
*-
```

```
      SUBROUTINE iau_PM ( P, R )
*+
*  - - - - - - -
*   i a u _ P M
*  - - - - - - -
*
*  Modulus of p-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)      p-vector
*
*  Returned:
*     R        d         modulus
*
*-
```

```
      SUBROUTINE iau_PMAT00 ( DATE1, DATE2, RBP )
*+
*  - - - - - - - - - - -
*   i a u _ P M A T 0 0
*  - - - - - - - - - - -
*
*  Precession matrix (including frame bias) from GCRS to a specified
*  date, IAU 2000 model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RBP         d(3,3)    bias-precession matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the arguments DATE1 and DATE2.  For
*     example, JD(TT)=2450123.7 could be expressed in any of these
*     ways, among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RBP * V(GCRS), where
*     the p-vector V(GCRS) is with respect to the Geocentric Celestial
*     Reference System (IAU, 2000) and the p-vector V(date) is with
*     respect to the mean equatorial triad of the given date.
*
*  Called:
*     iau_BP00     frame bias and precession matrices, IAU 2000
*
*  Reference:
*
*     IAU: Trans. International Astronomical Union, Vol. XXIVB;  Proc.
*     24th General Assembly, Manchester, UK.  Resolutions B1.3, B1.6.
*     (2000)
*
*-
```

```
      SUBROUTINE iau_PMAT06 ( DATE1, DATE2, RBP )
*+
*  - - - - - - - - - - -
*   i a u _ P M A T 0 6
*  - - - - - - - - - - -
*
*  Precession matrix (including frame bias) from GCRS to a specified
*  date, IAU 2006 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RBP          d(3,3)    bias-precession matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the arguments DATE1 and DATE2.  For
*     example, JD(TT)=2450123.7 could be expressed in any of these
*     ways, among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RBP * V(GCRS), where
*     the p-vector V(GCRS) is with respect to the Geocentric Celestial
*     Reference System (IAU, 2000) and the p-vector V(date) is with
*     respect to the mean equatorial triad of the given date.
*
*  Called:
*     iau_PFW06    bias-precession F-W angles, IAU 2006
*     iau_FW2M     F-W angles to r-matrix
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_PMAT76 ( DATE1, DATE2, RMATP )
*+
*  - - - - - - - - - - -
*   i a u _ P M A T 7 6
*  - - - - - - - - - - -
*
*  Precession matrix from J2000.0 to a specified date, IAU 1976 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       ending date, TT (Note 1)
*
*  Returned:
*     RMATP          d(3,3)  precession matrix, J2000.0 -> DATE1+DATE2
*
*  Notes:
*
*  1) The ending date DATE1+DATE2 is a Julian Date, apportioned
*     in any convenient way between the arguments DATE1 and DATE2.
*     For example, JD(TT)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RMATP * V(J2000),
*     where the p-vector V(J2000) is with respect to the mean
*     equatorial triad of epoch J2000.0 and the p-vector V(date)
*     is with respect to the mean equatorial triad of the given
*     date.
*
*  3) Though the matrix method itself is rigorous, the precession
*     angles are expressed through canonical polynomials which are
*     valid only for a limited time span.  In addition, the IAU 1976
*     precession rate is known to be imperfect.  The absolute accuracy
*     of the present formulation is better than 0.1 arcsec from
*     1960AD to 2040AD, better than 1 arcsec from 1640AD to 2360AD,
*     and remains below 3 arcsec for the whole of the period
*     500BC to 3000AD.  The errors exceed 10 arcsec outside the
*     range 1200BC to 3900AD, exceed 100 arcsec outside 4200BC to
*     5600AD and exceed 1000 arcsec outside 6800BC to 8200AD.
*
*  Called:
*     iau_PREC76   accumulated precession angles, IAU 1976
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RY       rotate around Y-axis
*     iau_CR       copy r-matrix
*
*  References:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*      equations (6) & (7), p283.
*
*     Kaplan, G.H., 1981, USNO circular no. 163, pA2.
*
*-
```

```
      SUBROUTINE iau_PMP ( A, B, AMB )
*+
*  - - - - - - - -
*   i a u _ P M P
*  - - - - - - - -
*
*  P-vector subtraction.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector
*     B        d(3)      second p-vector
*
*  Returned:
*     AMB      d(3)      A - B
*
*-
```

```
      SUBROUTINE iau_PMPX ( RC, DC, PR, PD, PX, RV, PMT, POB, PCO )
*+
*  - - - - - - - - -
*   i a u _ P M P X
*  - - - - - - - - -
*
*  Proper motion and parallax.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC,DC    d      ICRS RA,Dec at catalog epoch (radians)
*     PR       d      RA proper motion (radians/year; Note 1)
*     PD       d      Dec proper motion (radians/year)
*     PX       d      parallax (arcsec)
*     RV       d      radial velocity (km/s, +ve if receding)
*     PMT      d      proper motion time interval (SSB, Julian years)
*     POB      d(3)   SSB to observer vector (au)
*
*  Returned:
*     PCO      d(3)   coordinate direction (BCRS unit vector)
*
*  Notes:
*
*  1) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  2) The proper motion time interval is for when the starlight
*     reaches the solar system barycenter.
*
*  3) To avoid the need for iteration, the Roemer effect (i.e. the
*     small annual modulation of the proper motion coming from the
*     changing light time) is applied approximately, using the
*     direction of the star at the catalog epoch.
*
*  References:
*
*     1984 Astronomical Almanac, pp B39-B41.
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013), Section 7.2.
*
*  Called:
*     iau_PDP      scalar product of two p-vectors
*     iau_PN       decompose p-vector into modulus and direction
*
*-
```

```
      SUBROUTINE iau_PMSAFE ( RA1, DEC1, PMR1, PMD1, PX1, RV1,
     :                        EP1A, EP1B, EP2A, EP2B,
     :                        RA2, DEC2, PMR2, PMD2, PX2, RV2, J )
*+
*  - - - - - - - - - - -
*   i a u _ P M S A F E
*  - - - - - - - - - - -
*
*  Star proper motion:  update star catalog data for space motion, with
*  special handling to handle the zero parallax case.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RA1      d         right ascension (radians), before
*     DEC1     d         declination (radians), before
*     PMR1     d         RA proper motion (radians/year), before
*     PMD1     d         Dec proper motion (radians/year), before
*     PX1      d         parallax (arcseconds), before
*     RV1      d         radial velocity (km/s, +ve = receding), before
*     EP1A     d         "before" epoch, part A (Note 1)
*     EP1B     d         "before" epoch, part B (Note 1)
*     EP2A     d         "after" epoch, part A (Note 1)
*     EP2B     d         "after" epoch, part B (Note 1)
*
*  Returned:
*     RA2      d         right ascension (radians), after
*     DEC2     d         declination (radians), after
*     PMR2     d         RA proper motion (radians/year), after
*     PMD2     d         Dec proper motion (radians/year), after
*     PX2      d         parallax (arcseconds), after
*     RV2      d         radial velocity (km/s, +ve = receding), after
*     J        i         status:
*                          -1 = system error (should not occur)
*                           0 = no warnings or errors
*                           1 = distance overridden (Note 6)
*                           2 = excessive velocity (Note 7)
*                           4 = solution didn't converge (Note 8)
*                        else = binary logical OR of the above warnings
*
*  Notes:
*
*  1) The starting and ending TDB epochs EP1A+EP1B and EP2A+EP2B are
*     Julian Dates, apportioned in any convenient way between the two
*     parts (A and B).  For example, JD(TDB)=2450123.7 could be
*     expressed in any of these ways, among others:
*
*             EPnA          EPnB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     resolution.  The MJD method and the date & time methods are both
*     good compromises between resolution and convenience.
*
*  2) In accordance with normal star-catalog conventions, the object's
*     right ascension and declination are freed from the effects of
*     secular aberration.  The frame, which is aligned to the catalog
*     equator and equinox, is Lorentzian and centered on the SSB.
*
*     The proper motions are the rate of change of the right ascension
*     and declination at the catalog epoch and are in radians per TDB
*     Julian year.
```

```
*
*     The parallax and radial velocity are in the same frame.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds.
*
*  4) The RA proper motion is in terms of coordinate angle, not true
*     angle.  If the catalog uses arcseconds for both RA and Dec proper
*     motions, the RA proper motion will need to be divided by cos(Dec)
*     before use.
*
*  5) Straight-line motion at constant speed, in the inertial frame, is
*     assumed.
*
*  6) An extremely small (or zero or negative) parallax is overridden to
*     ensure that the object is at a finite but very large distance, but
*     not so large that the proper motion is equivalent to a large but
*     safe speed (about 0.1c using the chosen constant).  A warning
*     status of 1 is added to the status if this action has been taken.
*
*  7) If the space velocity is a significant fraction of c (see the
*     constant VMAX in the routine iau_STARPV), it is arbitrarily set to
*     zero.  When this action occurs, 2 is added to the status.
*
*  8) The relativistic adjustment carried out in the iau_STARPV routine
*     involves an iterative calculation.  If the process fails to
*     converge within a set number of iterations, 4 is added to the
*     status.
*
*  Called:
*     iau_SEPS     angle between two points
*     iau_STARPM   update star catalog data for space motion
*
*-
```

```
      SUBROUTINE iau_PN ( P, R, U )
*+
*  - - - - - - -
*   i a u _ P N
*  - - - - - - -
*
*  Convert a p-vector into modulus and unit vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     P        d(3)      p-vector
*
*  Returned:
*     R        d         modulus
*     U        d(3)      unit vector
*
*  Note:
*     If P is null, the result is null.  Otherwise the result is
*     a unit vector.
*
*  Called:
*     iau_PM       modulus of p-vector
*     iau_ZP       zero p-vector
*     iau_SXP      multiply p-vector by scalar
*
*-
```

```
      SUBROUTINE iau_PN00 ( DATE1, DATE2, DPSI, DEPS,
     :                      EPSA, RB, RP, RBP, RN, RBPN )
*+
*  - - - - - - - - -
*   i a u _ P N 0 0
*  - - - - - - - - -
*
*  Precession-nutation, IAU 2000 model:  a multi-purpose routine,
*  supporting classical (equinox-based) use directly and CIO-based
*  use indirectly.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*     DPSI,DEPS     d       nutation (Note 2)
*
*  Returned:
*     EPSA          d       mean obliquity (Note 3)
*     RB          d(3,3)    frame bias matrix (Note 4)
*     RP          d(3,3)    precession matrix (Note 5)
*     RBP         d(3,3)    bias-precession matrix (Note 6)
*     RN          d(3,3)    nutation matrix (Note 7)
*     RBPN        d(3,3)    GCRS-to-true matrix (Note 8)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The caller is responsible for providing the nutation components;
*     they are in longitude and obliquity, in radians and are with
*     respect to the equinox and ecliptic of date.  For high-accuracy
*     applications, free core nutation should be included as well as
*     any other relevant corrections to the position of the CIP.
*
*  3) The returned mean obliquity is consistent with the IAU 2000
*     precession-nutation models.
*
*  4) The matrix RB transforms vectors from GCRS to J2000.0 mean equator
*     and equinox by applying frame bias.
*
*  5) The matrix RP transforms vectors from J2000.0 mean equator and
*     equinox to mean equator and equinox of date by applying
*     precession.
*
*  6) The matrix RBP transforms vectors from GCRS to mean equator and
*     equinox of date by applying frame bias then precession.  It is the
*     product RP x RB.
*
*  7) The matrix RN transforms vectors from mean equator and equinox of
*     date to true equator and equinox of date by applying the nutation
*     (luni-solar + planetary).
```

```
*
*  8) The matrix RBPN transforms vectors from GCRS to true equator and
*     equinox of date.  It is the product RN x RBP, applying frame bias,
*     precession and nutation in that order.
*
*  Called:
*     iau_PR00     IAU 2000 precession adjustments
*     iau_OBL80    mean obliquity, IAU 1980
*     iau_BP00     frame bias and precession matrices, IAU 2000
*     iau_NUMAT    form nutation matrix
*     iau_RXR      product of two r-matrices
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*-
```

```
      SUBROUTINE iau_PN00A ( DATE1, DATE2,
     :                       DPSI, DEPS, EPSA, RB, RP, RBP, RN, RBPN )
*+
*  - - - - - - - - - -
*   i a u _ P N 0 0 A
*  - - - - - - - - - -
*
*  Precession-nutation, IAU 2000A model:  a multi-purpose routine,
*  supporting classical (equinox-based) use directly and CIO-based
*  use indirectly.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d       nutation (Note 2)
*     EPSA          d       mean obliquity (Note 3)
*     RB          d(3,3)    frame bias matrix (Note 4)
*     RP          d(3,3)    precession matrix (Note 5)
*     RBP         d(3,3)    bias-precession matrix (Note 6)
*     RN          d(3,3)    nutation matrix (Note 7)
*     RBPN        d(3,3)    GCRS-to-true matrix (Notes 8,9)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components (luni-solar + planetary, IAU 2000A) in
*     longitude and obliquity are in radians and with respect to the
*     equinox and ecliptic of date.  Free core nutation is omitted;  for
*     the utmost accuracy, use the iau_PN00 routine, where the nutation
*     components are caller-specified.  For faster but slightly less
*     accurate results, use the iau_PN00B routine.
*
*  3) The mean obliquity is consistent with the IAU 2000 precession.
*
*  4) The matrix RB transforms vectors from GCRS to J2000.0 mean equator
*     and equinox by applying frame bias.
*
*  5) The matrix RP transforms vectors from J2000.0 mean equator and
*     equinox to mean equator and equinox of date by applying
*     precession.
*
*  6) The matrix RBP transforms vectors from GCRS to mean equator and
*     equinox of date by applying frame bias then precession.  It is the
*     product RP x RB.
*
*  7) The matrix RN transforms vectors from mean equator and equinox of
*     date to true equator and equinox of date by applying the nutation
*     (luni-solar + planetary).
```

```
*
*  8) The matrix RBPN transforms vectors from GCRS to true equator and
*     equinox of date.  It is the product RN x RBP, applying frame bias,
*     precession and nutation in that order.
*
*  9) The X,Y,Z coordinates of the IAU 2000A Celestial Intermediate Pole
*     are elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_NUT00A   nutation, IAU 2000A
*     iau_PN00     bias/precession/nutation results, IAU 2000
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003).
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*-
```

```
      SUBROUTINE iau_PN00B ( DATE1, DATE2,
     :                       DPSI, DEPS, EPSA, RB, RP, RBP, RN, RBPN )
*+
*  - - - - - - - - - -
*   i a u _ P N 0 0 B
*  - - - - - - - - - -
*
*  Precession-nutation, IAU 2000B model:  a multi-purpose routine,
*  supporting classical (equinox-based) use directly and CIO-based
*  use indirectly.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d       nutation (Note 2)
*     EPSA          d       mean obliquity (Note 3)
*     RB          d(3,3)    frame bias matrix (Note 4)
*     RP          d(3,3)    precession matrix (Note 5)
*     RBP         d(3,3)    bias-precession matrix (Note 6)
*     RN          d(3,3)    nutation matrix (Note 7)
*     RBPN        d(3,3)    GCRS-to-true matrix (Notes 8,9)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*        2450123.7D0        0D0        (JD method)
*         2451545D0      -1421.3D0     (J2000 method)
*        2400000.5D0     50123.2D0     (MJD method)
*        2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components (luni-solar + planetary, IAU 2000B) in
*     longitude and obliquity are in radians and with respect to the
*     equinox and ecliptic of date.  For more accurate results, but
*     at the cost of increased computation, use the iau_PN00A routine.
*     For the utmost accuracy, use the iau_PN00 routine, where the
*     nutation components are caller-specified.
*
*  3) The mean obliquity is consistent with the IAU 2000 precession.
*
*  4) The matrix RB transforms vectors from GCRS to J2000.0 mean equator
*     and equinox by applying frame bias.
*
*  5) The matrix RP transforms vectors from J2000.0 mean equator and
*     equinox to mean equator and equinox of date by applying
*     precession.
*
*  6) The matrix RBP transforms vectors from GCRS to mean equator and
*     equinox of date by applying frame bias then precession.  It is the
*     product RP x RB.
*
*  7) The matrix RN transforms vectors from mean equator and equinox of
*     date to true equator and equinox of date by applying the nutation
*     (luni-solar + planetary).
```

```
*
*  8) The matrix RBPN transforms vectors from GCRS to true equator and
*     equinox of date.  It is the product RN x RBP, applying frame bias,
*     precession and nutation in that order.
*
*  9) The X,Y,Z coordinates of the IAU 2000B Celestial Intermediate Pole
*     are elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_NUT00B   nutation, IAU 2000B
*     iau_PN00     bias/precession/nutation results, IAU 2000
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003).
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*-
```

```
      SUBROUTINE iau_PN06 ( DATE1, DATE2, DPSI, DEPS,
     :                      EPSA, RB, RP, RBP, RN, RBPN )
*+
*  - - - - - - - - -
*   i a u _ P N 0 6
*  - - - - - - - - -
*
*  Precession-nutation, IAU 2006 model:  a multi-purpose routine,
*  supporting classical (equinox-based) use directly and CIO-based use
*  indirectly.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*     DPSI,DEPS     d       nutation (Note 2)
*
*  Returned:
*     EPSA          d       mean obliquity (Note 3)
*     RB          d(3,3)    frame bias matrix (Note 4)
*     RP          d(3,3)    precession matrix (Note 5)
*     RBP         d(3,3)    bias-precession matrix (Note 6)
*     RN          d(3,3)    nutation matrix (Note 7)
*     RBPN        d(3,3)    GCRS-to-true matrix (Note 8)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The caller is responsible for providing the nutation components;
*     they are in longitude and obliquity, in radians and are with
*     respect to the equinox and ecliptic of date.  For high-accuracy
*     applications, free core nutation should be included as well as
*     any other relevant corrections to the position of the CIP.
*
*  3) The returned mean obliquity is consistent with the IAU 2006
*     precession.
*
*  4) The matrix RB transforms vectors from GCRS to mean J2000.0 by
*     applying frame bias.
*
*  5) The matrix RP transforms vectors from mean J2000.0 to mean of date
*     by applying precession.
*
*  6) The matrix RBP transforms vectors from GCRS to mean of date by
*     applying frame bias then precession.  It is the product RP x RB.
*
*  7) The matrix RN transforms vectors from mean of date to true of date
*     by applying the nutation (luni-solar + planetary).
*
*  8) The matrix RBPN transforms vectors from GCRS to true of date
*     (CIP/equinox).  It is the product RN x RBP, applying frame bias,
```

```
*     precession and nutation in that order.
*
*  9) The X,Y,Z coordinates of the Celestial Intermediate Pole are
*     elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_PFW06    bias-precession F-W angles, IAU 2006
*     iau_FW2M     F-W angles to r-matrix
*     iau_TR       transpose r-matrix
*     iau_RXR      product of two r-matrices
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_PN06A ( DATE1, DATE2,
     :                       DPSI, DEPS, EPSA, RB, RP, RBP, RN, RBPN )
*+
*  - - - - - - - - - - -
*   i a u _ P N 0 6 A
*  - - - - - - - - - - -
*
*  Precession-nutation, IAU 2006/2000A models:  a multi-purpose routine,
*  supporting classical (equinox-based) use directly and CIO-based use
*  indirectly.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d       nutation (Note 2)
*     EPSA          d       mean obliquity (Note 3)
*     RB          d(3,3)    frame bias matrix (Note 4)
*     RP          d(3,3)    precession matrix (Note 5)
*     RBP         d(3,3)    bias-precession matrix (Note 6)
*     RN          d(3,3)    nutation matrix (Note 7)
*     RBPN        d(3,3)    GCRS-to-true matrix (Notes 8,9)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components (luni-solar + planetary, IAU 2000A) in
*     longitude and obliquity are in radians and with respect to the
*     equinox and ecliptic of date.  Free core nutation is omitted;  for
*     the utmost accuracy, use the iau_PN06 routine, where the nutation
*     components are caller-specified.
*
*  3) The mean obliquity is consistent with the IAU 2006 precession.
*
*  4) The matrix RB transforms vectors from GCRS to mean J2000.0 by
*     applying frame bias.
*
*  5) The matrix RP transforms vectors from mean J2000.0 to mean of date
*     by applying precession.
*
*  6) The matrix RBP transforms vectors from GCRS to mean of date by
*     applying frame bias then precession.  It is the product RP x RB.
*
*  7) The matrix RN transforms vectors from mean of date to true of date
*     by applying the nutation (luni-solar + planetary).
*
*  8) The matrix RBPN transforms vectors from GCRS to true of date
*     (CIP/equinox).  It is the product RN x RBP, applying frame bias,
*     precession and nutation in that order.
```

```
*
*  9) The X,Y,Z coordinates of the IAU 2006/2000A Celestial Intermediate
*     Pole are elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_NUT06A   nutation, IAU 2006/2000A
*     iau_PN06     bias/precession/nutation results, IAU 2006
*
*  Reference:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*-
```

```
      SUBROUTINE iau_PNM00A ( DATE1, DATE2, RBPN )
*+
*  - - - - - - - - - - -
*   i a u _ P N M 0 0 A
*  - - - - - - - - - - -
*
*  Form the matrix of precession-nutation for a given date (including
*  frame bias), equinox-based, IAU 2000A model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RBPN         d(3,3)    classical NPB matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RBPN * V(GCRS), where
*     the p-vector V(date) is with respect to the true equatorial triad
*     of date DATE1+DATE2 and the p-vector V(GCRS) is with respect to
*     the Geocentric Celestial Reference System (IAU, 2000).
*
*  3) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_PNM00B routine.
*
*  Called:
*     iau_PN00A    bias/precession/nutation, IAU 2000A
*
*  Reference:
*
*     IAU: Trans. International Astronomical Union, Vol. XXIVB;  Proc.
*     24th General Assembly, Manchester, UK.  Resolutions B1.3, B1.6.
*     (2000)
*
*-
```

```
      SUBROUTINE iau_PNM00B ( DATE1, DATE2, RBPN )
*+
*  - - - - - - - - - - -
*   i a u _ P N M 0 0 B
*  - - - - - - - - - - -
*
*  Form the matrix of precession-nutation for a given date (including
*  frame bias), equinox-based, IAU 2000B model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RBPN         d(3,3)    bias-precession-nutation matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RBPN * V(GCRS), where
*     the p-vector V(date) is with respect to the true equatorial triad
*     of date DATE1+DATE2 and the p-vector V(GCRS) is with respect to
*     the Geocentric Celestial Reference System (IAU, 2000).
*
*  3) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_PNM00A routine.
*
*  Called:
*     iau_PN00B    bias/precession/nutation, IAU 2000B
*
*  Reference:
*
*     IAU: Trans. International Astronomical Union, Vol. XXIVB;  Proc.
*     24th General Assembly, Manchester, UK.  Resolutions B1.3, B1.6.
*     (2000)
*
*-
```

```
      SUBROUTINE iau_PNM06A ( DATE1, DATE2, RNPB )
*+
*  - - - - - - - - - - -
*   i a u _ P N M 0 6 A
*  - - - - - - - - - - -
*
*  Form the matrix of precession-nutation for a given date (including
*  frame bias), IAU 2006 precession and IAU 2000A nutation models.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RNPB         d(3,3)    bias-precession-nutation matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RNPB * V(GCRS), where
*     the p-vector V(date) is with respect to the true equatorial triad
*     of date DATE1+DATE2 and the p-vector V(GCRS) is with respect to
*     the Geocentric Celestial Reference System (IAU, 2000).
*
*  Called:
*     iau_PFW06    bias-precession F-W angles, IAU 2006
*     iau_NUT06A   nutation, IAU 2006/2000A
*     iau_FW2M     F-W angles to r-matrix
*
*  Reference:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855.
*
*-
```

```
      SUBROUTINE iau_PNM80 ( DATE1, DATE2, RMATPN )
*+
*  - - - - - - - - - -
*   i a u _ P N M 8 0
*  - - - - - - - - - -
*
*  Form the matrix of precession/nutation for a given date, IAU 1976
*  precession model, IAU 1980 nutation model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2     d         TDB date (Note 1)
*
*  Returned:
*     RMATPN          d(3,3)    combined precession/nutation matrix
*
*  Notes:
*
*  1) The TDB date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TDB)=2450123.7 could be expressed in any of these ways, among
*     others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RMATPN * V(J2000),
*     where the p-vector V(date) is with respect to the true
*     equatorial triad of date DATE1+DATE2 and the p-vector
*     V(J2000) is with respect to the mean equatorial triad of
*     epoch J2000.0.
*
*  Called:
*     iau_PMAT76   precession matrix, IAU 1976
*     iau_NUTM80   nutation matrix, IAU 1980
*     iau_RXR      product of two r-matrices
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.3 (p145).
*
*-
```

```
      SUBROUTINE iau_POM00 ( XP, YP, SP, RPOM )
*+
*  - - - - - - - - - - -
*   i a u _ P O M 0 0
*  - - - - - - - - - - -
*
*  Form the matrix of polar motion for a given date, IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XP,YP      d      coordinates of the pole (radians, Note 1)
*     SP         d      the TIO locator s' (radians, Note 2)
*
*  Returned:
*     RPOM     d(3,3)   polar-motion matrix (Note 3)
*
*  Notes:
*
*  1) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians to 0 and 90 deg west respectively.
*
*  2) SP is the TIO locator s', in radians, which positions the
*     Terrestrial Intermediate Origin on the equator.  It is obtained
*     from polar motion observations by numerical integration, and so is
*     in essence unpredictable.  However, it is dominated by a secular
*     drift of about 47 microarcseconds per century, and so can be taken
*     into account by using s' = -47*t, where t is centuries since
*     J2000.0.  The routine iau_SP00 implements this approximation.
*
*  3) The matrix operates in the sense V(TRS) = RPOM * V(CIP), meaning
*     that it is the final rotation when computing the pointing
*     direction to a celestial source.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RY       rotate around Y-axis
*     iau_RX       rotate around X-axis
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_PPP ( A, B, APB )
*+
*  - - - - - - - -
*   i a u _ P P P
*  - - - - - - - -
*
*  P-vector addition.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector
*     B        d(3)      second p-vector
*
*  Returned:
*     APB      d(3)      A + B
*
*-
```

```
      SUBROUTINE iau_PPSP ( A, S, B, APSB )
*+
*  - - - - - - - - -
*   i a u _ P P S P
*  - - - - - - - - -
*
*  P-vector plus scaled p-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector
*     S        d         scalar (multiplier for B)
*     B        d(3)      second p-vector
*
*  Returned:
*     APSB     d(3)      A + S*B
*
*-
```

```
      SUBROUTINE iau_PR00 ( DATE1, DATE2, DPSIPR, DEPSPR )
*+
*  - - - - - - - - - -
*   i a u _ P R 0 0
*  - - - - - - - - - -
*
*  Precession-rate part of the IAU 2000 precession-nutation models
*  (part of MHB2000).
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d   TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSIPR,DEPSPR  d   precession corrections (Notes 2,3)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The precession adjustments are expressed as "nutation components",
*     corrections in longitude and obliquity with respect to the J2000.0
*     equinox and ecliptic.
*
*  3) Although the precession adjustments are stated to be with respect
*     to Lieske et al. (1977), the MHB2000 model does not specify which
*     set of Euler angles are to be used and how the adjustments are to
*     be applied.  The most literal and straightforward procedure is to
*     adopt the 4-rotation epsilon_0, psi_A, omega_A, xi_A option, and
*     to add DPSIPR to psi_A and DEPSPR to both omega_A and eps_A.
*
*  4) This is an implementation of one aspect of the IAU 2000A nutation
*     model, formally adopted by the IAU General Assembly in 2000,
*     namely MHB2000 (Mathews et al. 2002).
*
*  References:
*
*     Lieske, J.H., Lederle, T., Fricke, W. & Morando, B., "Expressions
*     for the precession quantities based upon the IAU (1976) System of
*     Astronomical Constants", Astron.Astrophys., 58, 1-16 (1977)
*
*     Mathews, P.M., Herring, T.A., Buffet, B.A., "Modeling of nutation
*     and precession   New nutation series for nonrigid Earth and
*     insights into the Earth’s interior", J.Geophys.Res., 107, B4,
*     2002.  The MHB2000 code itself was obtained on 9th September 2002
*     from ftp://maia.usno.navy.mil/conv2000/chapter5/IAU2000A.
*
*     Wallace, P.T., "Software for Implementing the IAU 2000
*     Resolutions", in IERS Workshop 5.1 (2002).
*
*-
```

```
      SUBROUTINE iau_PREC76 ( DATE01, DATE02, DATE11, DATE12,
     :                        ZETA, Z, THETA )
*+
*  - - - - - - - - - - -
*   i a u _ P R E C 7 6
*  - - - - - - - - - - -
*
*  IAU 1976 precession model.
*
*  This routine forms the three Euler angles which implement general
*  precession between two dates, using the IAU 1976 model (as for
*  the FK5 catalog).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE01,DATE02  d   TDB starting date (Note 1)
*     DATE11,DATE12  d   TDB ending date (Note 1)
*
*  Returned:
*     ZETA           d   1st rotation: radians clockwise around z
*     Z              d   3rd rotation: radians clockwise around z
*     THETA          d   2nd rotation: radians counterclockwise around y
*
*  Notes:
*
*  1) The dates DATE01+DATE02 and DATE11+DATE12 are Julian Dates,
*     apportioned in any convenient way between the arguments DATEn1 and
*     DATEn2.  For example, JD(TDB)=2450123.7 could be expressed in any
*     of these ways, among others:
*
*           DATEn1         DATEn2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     The two dates may be expressed using different methods, but at
*     the risk of losing some resolution.
*
*  2) The accumulated precession angles zeta, z, theta are expressed
*     through canonical polynomials which are valid only for a limited
*     time span.  In addition, the IAU 1976 precession rate is known to
*     be imperfect.  The absolute accuracy of the present formulation is
*     better than 0.1 arcsec from 1960AD to 2040AD, better than 1 arcsec
*     from 1640AD to 2360AD, and remains below 3 arcsec for the whole of
*     the period 500BC to 3000AD.  The errors exceed 10 arcsec outside
*     the range 1200BC to 3900AD, exceed 100 arcsec outside 4200BC to
*     5600AD and exceed 1000 arcsec outside 6800BC to 8200AD.
*
*  3) The three angles are returned in the conventional order, which
*     is not the same as the order of the corresponding Euler rotations.
*     The precession matrix is R_3(-z) x R_2(+theta) x R_3(-zeta).
*
*  Reference:
*
*     Lieske, J.H., 1979, Astron.Astrophys. 73, 282.
*      equations (6) & (7), p283.
*
*-
```

```
      SUBROUTINE iau_PV2P ( PV, P )
*+
*  - - - - - - - - -
*   i a u _ P V 2 P
*  - - - - - - - - -
*
*  Discard velocity component of a pv-vector.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PV       d(3,2)      pv-vector
*
*  Returned:
*     P        d(3)        p-vector
*
*  Called:
*     iau_CP       copy p-vector
*
*-
```

```
      SUBROUTINE iau_PV2S ( PV, THETA, PHI, R, TD, PD, RD )
*+
*  - - - - - - - - -
*   i a u _ P V 2 S
*  - - - - - - - - -
*
*  Convert position/velocity from Cartesian to spherical coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*     R        d         radial distance
*     TD       d         rate of change of THETA
*     PD       d         rate of change of PHI
*     RD       d         rate of change of R
*
*  Notes:
*
*  1) If the position part of PV is null, THETA, PHI, TD and PD
*     are indeterminate.  This is handled by extrapolating the
*     position through unit time by using the velocity part of
*     PV.  This moves the origin without changing the direction
*     of the velocity component.  If the position and velocity
*     components of PV are both null, zeroes are returned for all
*     six results.
*
*  2) If the position is a pole, THETA, TD and PD are indeterminate.
*     In such cases zeroes are returned for all three.
*
*-
```

```
      SUBROUTINE iau_PVDPV ( A, B, ADB )
*+
*  - - - - - - - - - - -
*   i a u _ P V D P V
*  - - - - - - - - - - -
*
*  Inner (=scalar=dot) product of two pv-vectors.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3,2)      first pv-vector
*     B        d(3,2)      second pv-vector
*
*  Returned:
*     ADB      d(2)        A . B (see note)
*
*  Note:
*
*     If the position and velocity components of the two pv-vectors are
*     ( Ap, Av ) and ( Bp, Bv ), the result, A . B, is the pair of
*     numbers ( Ap . Bp , Ap . Bv + Av . Bp ).  The two numbers are the
*     dot-product of the two p-vectors and its derivative.
*
*  Called:
*     iau_PDP      scalar product of two p-vectors
*
*-
```

```
      SUBROUTINE iau_PVM ( PV, R, S )
*+
*  - - - - - - - -
*   i a u _ P V M
*  - - - - - - - -
*
*  Modulus of pv-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     R        d         modulus of position component
*     S        d         modulus of velocity component
*
*  Called:
*     iau_PM       modulus of p-vector
*
*-
```

```
      SUBROUTINE iau_PVMPV ( A, B, AMB )
*+
*  - - - - - - - - - -
*   i a u _ P V M P V
*  - - - - - - - - - -
*
*  Subtract one pv-vector from another.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3,2)      first pv-vector
*     B        d(3,2)      second pv-vector
*
*  Returned:
*     AMB      d(3,2)      A - B
*
*  Called:
*     iau_PMP      p-vector minus p-vector
*
*-
```

```
      SUBROUTINE iau_PVPPV ( A, B, APB )
*+
*  - - - - - - - - - -
*   i a u _ P V P P V
*  - - - - - - - - - -
*
*  Add one pv-vector to another.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3,2)      first pv-vector
*     B        d(3,2)      second pv-vector
*
*  Returned:
*     APB      d(3,2)      A + B
*
*  Called:
*     iau_PPP      p-vector plus p-vector
*
*-
```

```
      SUBROUTINE iau_PVSTAR ( PV, RA, DEC, PMR, PMD, PX, RV, J )
*+
*  - - - - - - - - - - -
*   i a u _ P V S T A R
*  - - - - - - - - - - -
*
*  Convert star position+velocity vector to catalog coordinates.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given (Note 1):
*     PV       d(3,2)    pv-vector (au, au/day)
*
*  Returned (Note 2):
*     RA       d         right ascension (radians)
*     DEC      d         declination (radians)
*     PMR      d         RA proper motion (radians/year)
*     PMD      d         Dec proper motion (radians/year)
*     PX       d         parallax (arcsec)
*     RV       d         radial velocity (km/s, positive = receding)
*     J        i         status:
*                           0 = OK
*                          -1 = superluminal speed (Note 5)
*                          -2 = null position vector
*
*  Notes:
*
*  1) The specified pv-vector is the coordinate direction (and its rate
*     of change) for the epoch at which the light leaving the star
*     reached the solar-system barycenter.
*
*  2) The star data returned by this routine are "observables" for an
*     imaginary observer at the solar-system barycenter.  Proper motion
*     and radial velocity are, strictly, in terms of barycentric
*     coordinate time, TCB.  For most practical applications, it is
*     permissible to neglect the distinction between TCB and ordinary
*     "proper" time on Earth (TT/TAI).  The result will, as a rule, be
*     limited by the intrinsic accuracy of the proper-motion and radial-
*     velocity data;  moreover, the supplied pv-vector is likely to be
*     merely an intermediate result (for example generated by the
*     routine iau_STARPV), so that a change of time unit will cancel
*     out overall.
*
*     In accordance with normal star-catalog conventions, the object’s
*     right ascension and declination are freed from the effects of
*     secular aberration.  The frame, which is aligned to the catalog
*     equator and equinox, is Lorentzian and centered on the SSB.
*
*     Summarizing, the specified pv-vector is for most stars almost
*     identical to the result of applying the standard geometrical
*     "space motion" transformation to the catalog data.  The
*     differences, which are the subject of the Stumpff paper cited
*     below, are:
*
*     (i) In stars with significant radial velocity and proper motion,
*     the constantly changing light-time distorts the apparent proper
*     motion.  Note that this is a classical, not a relativistic,
*     effect.
*
*     (ii) The transformation complies with special relativity.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds; the radial velocity is in km/s, but
*     the pv-vector result is in au and au/day.
*
*  4) The proper motions are the rate of change of the right ascension
*     and declination at the catalog epoch and are in radians per Julian
*     year.  The RA proper motion is in terms of coordinate angle, not
```

```
*     true angle, and will thus be numerically larger at high
*     declinations.
*
*  5) Straight-line motion at constant speed in the inertial frame is
*     assumed.  If the speed is greater than or equal to the speed of
*     light, the routine aborts with an error status.
*
*  6) The inverse transformation is performed by the routine iau_STARPV.
*
*  Called:
*     iau_PN       decompose p-vector into modulus and direction
*     iau_PDP      scalar product of two p-vectors
*     iau_SXP      multiply p-vector by scalar
*     iau_PMP      p-vector minus p-vector
*     iau_PM       modulus of p-vector
*     iau_PPP      p-vector plus p-vector
*     iau_PV2S     pv-vector to spherical
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  Reference:
*
*     Stumpff, P., Astron.Astrophys. 144, 232-240 (1985).
*
*-
```

```
      SUBROUTINE iau_PVTOB ( ELONG, PHI, HM, XP, YP, SP, THETA, PV )
*+
*  - - - - - - - - - -
*   i a u _ P V T O B
*  - - - - - - - - - -
*
*  Position and velocity of a terrestrial observing station.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     ELONG      d       longitude (radians, east +ve, Note 1)
*     PHI        d       latitude (geodetic, radians, Note 1)
*     HM         d       height above reference ellipsoid (geodetic, m)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*     SP         d       the TIO locator s’ (radians, Note 2)
*     THETA      d       Earth rotation angle (radians, Note 3)
*
*  Returned:
*     PV         d(3,2)  position/velocity vector (m, m/s, CIRS)
*
*  Notes:
*
*  1) The terrestrial coordinates are with respect to the WGS84
*     reference ellipsoid.
*
*  2) XP and YP are the coordinates (in radians) of the Celestial
*     Intermediate Pole with respect to the International Terrestrial
*     Reference System (see IERS Conventions 2003), measured along the
*     meridians 0 and 90 deg west respectively.  SP is the TIO locator
*     s’, in radians, which positions the Terrestrial Intermediate
*     Origin on the equator.  For many applications, XP, YP and
*     (especially) SP can be set to zero.
*
*  3) If THETA is Greenwich apparent sidereal time instead of Earth
*     rotation angle, the result is with respect to the true equator
*     and equinox of date, i.e. with the x-axis at the equinox rather
*     than the celestial intermediate origin.
*
*  4) The velocity units are meters per UT1 second, not per SI second.
*     This is unlikely to have any practical consequences in the modern
*     era.
*
*  5) No validation is performed on the arguments.  Error cases that
*     could lead to arithmetic exceptions are trapped by the iau_GD2GC
*     routine, and the result set to zeros.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Urban, S. & Seidelmann, P. K. (eds), Explanatory Supplement to
*     the Astronomical Almanac, 3rd ed., University Science Books
*     (2013), Section 7.4.3.3.
*
*  Called:
*     iau_GD2GC    geodetic to geocentric transformation
*     iau_POM00    polar motion matrix
*     iau_TRXP     product of transpose of r-matrix and p-vector
*
*-
```

```
      SUBROUTINE iau_PVU ( DT, PV, UPV )
*+
*  - - - - - - - -
*   i a u _ P V U
*  - - - - - - - -
*
*  Update a pv-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     DT       d           time interval
*     PV       d(3,2)      pv-vector
*
*  Returned:
*     UPV      d(3,2)      p updated, v unchanged
*
*  Notes:
*
*  1) "Update" means "refer the position component of the vector
*     to a new epoch DT time units from the existing epoch".
*
*  2) The time units of DT must match those of the velocity.
*
*  Called:
*     iau_PPSP     p-vector plus scaled p-vector
*     iau_CP       copy p-vector
*
*-
```

```
      SUBROUTINE iau_PVUP ( DT, PV, P )
*+
*  - - - - - - - - -
*   i a u _ P V U P
*  - - - - - - - - -
*
*  Update a pv-vector, discarding the velocity component.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     DT       d           time interval
*     PV       d(3,2)      pv-vector
*
*  Returned:
*     P        d(3)        p-vector
*
*  Notes:
*
*  1) "Update" means "refer the position component of the vector to a
*     new date DT time units from the existing date".
*
*  2) The time units of DT must match those of the velocity.
*
*-
```

```
      SUBROUTINE iau_PVXPV ( A, B, AXB )
*+
*  - - - - - - - - - -
*   i a u _ P V X P V
*  - - - - - - - - - -
*
*  Outer (=vector=cross) product of two pv-vectors.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3,2)      first pv-vector
*     B        d(3,2)      second pv-vector
*
*  Returned:
*     AXB      d(3,2)      A x B
*
*  Note:
*
*     If the position and velocity components of the two pv-vectors are
*     ( Ap, Av ) and ( Bp, Bv ), the result, A x B, is the pair of
*     vectors ( Ap x Bp, Ap x Bv + Av x Bp ).  The two vectors are the
*     cross-product of the two p-vectors and its derivative.
*
*  Called:
*     iau_CPV      copy pv-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PPP      p-vector plus p-vector
*
*-
```

```
      SUBROUTINE iau_PXP ( A, B, AXB )
*+
*  - - - - - - - -
*   i a u _ P X P
*  - - - - - - - -
*
*  p-vector outer (=vector=cross) product.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector
*     B        d(3)      second p-vector
*
*  Returned:
*     AXB      d(3)      A x B
*
*-
```

```
      SUBROUTINE iau_REFCO ( PHPA, TC, RH, WL, REFA, REFB )
*+
*  - - - - - - - - - -
*   i a u _ R E F C O
*  - - - - - - - - - -
*
*  Determine the constants A and B in the atmospheric refraction model
*  dZ = A tan Z + B tan^3 Z.
*
*  Z is the "observed" zenith distance (i.e. affected by refraction)
*  and dZ is what to add to Z to give the "topocentric" (i.e. in vacuo)
*  zenith distance.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     PHPA     d      pressure at the observer (hPa = millibar)
*     TC       d      ambient temperature at the observer (deg C)
*     RH       d      relative humidity at the observer (range 0-1)
*     WL       d      wavelength (micrometers)
*
*  Returned:
*     REFA     d      tan Z coefficient (radians)
*     REFB     d      tan^3 Z coefficient (radians)
*
*  Notes:
*
*  1) The model balances speed and accuracy to give good results in
*     applications where performance at low altitudes is not paramount.
*     Performance is maintained across a range of conditions, and
*     applies to both optical/IR and radio.
*
*  2) The model omits the effects of (i) height above sea level (apart
*     from the reduced pressure itself), (ii) latitude (i.e. the
*     flattening of the Earth), (iii) variations in tropospheric lapse
*     rate and (iv) dispersive effects in the radio.
*
*     The model was tested using the following range of conditions:
*
*       lapse rates 0.0055, 0.0065, 0.0075 deg/meter
*       latitudes 0, 25, 50, 75 degrees
*       heights 0, 2500, 5000 meters ASL
*       pressures mean for height -10% to +5% in steps of 5%
*       temperatures -10 deg to +20 deg with respect to 280 deg at SL
*       relative humidity 0, 0.5, 1
*       wavelengths 0.4, 0.6, ... 2 micron, + radio
*       zenith distances 15, 45, 75 degrees
*
*     The accuracy with respect to raytracing through a model
*     atmosphere was as follows:
*
*                            worst         RMS
*
*       optical/IR           62 mas       8 mas
*       radio               319 mas      49 mas
*
*     For this particular set of conditions:
*
*       lapse rate 0.0065 K/meter
*       latitude 50 degrees
*       sea level
*       pressure 1005 mb
*       temperature 280.15 K
*       humidity 80%
*       wavelength 5740 Angstroms
*
*     the results were as follows:
*
*       ZD       raytrace     iau_REFCO  Saastamoinen
```

```
*
*       10         10.27        10.27        10.27
*       20         21.19        21.20        21.19
*       30         33.61        33.61        33.60
*       40         48.82        48.83        48.81
*       45         58.16        58.18        58.16
*       50         69.28        69.30        69.27
*       55         82.97        82.99        82.95
*       60        100.51       100.54       100.50
*       65        124.23       124.26       124.20
*       70        158.63       158.68       158.61
*       72        177.32       177.37       177.31
*       74        200.35       200.38       200.32
*       76        229.45       229.43       229.42
*       78        267.44       267.29       267.41
*       80        319.13       318.55       319.10
*
*      deg        arcsec       arcsec       arcsec
*
*     The values for Saastamoinen’s formula (which includes terms
*     up to tan^5) are taken from Hohenkerk and Sinclair (1985).
*
*  3) A WL value in the range 0−100 selects the optical/IR case and is
*     wavelength in micrometers.  Any value outside this range selects
*     the radio case.
*
*  4) Outlandish input parameters are silently limited to mathematically
*     safe values.  Zero pressure is permissible, and causes zeroes to
*     be returned.
*
*  5) The algorithm draws on several sources, as follows:
*
*     a) The formula for the saturation vapour pressure of water as
*        a function of temperature and temperature is taken from
*        Equations (A4.5−A4.7) of Gill (1982).
*
*     b) The formula for the water vapour pressure, given the
*        saturation pressure and the relative humidity, is from
*        Crane (1976), Equation (2.5.5).
*
*     c) The refractivity of air is a function of temperature,
*        total pressure, water−vapour pressure and, in the case
*        of optical/IR, wavelength.  The formulae for the two cases are
*        developed from Hohenkerk & Sinclair (1985) and Rueger (2002).
*
*     d) The formula for beta, the ratio of the scale height of the
*        atmosphere to the geocentric distance of the observer, is
*        an adaption of Equation (9) from Stone (1996).  The
*        adaptations, arrived at empirically, consist of (i) a small
*        adjustment to the coefficient and (ii) a humidity term for the
*        radio case only.
*
*     e) The formulae for the refraction constants as a function of
*        n−1 and beta are from Green (1987), Equation (4.31).
*
*  References:
*
*     Crane, R.K., Meeks, M.L. (ed), "Refraction Effects in the Neutral
*     Atmosphere", Methods of Experimental Physics: Astrophysics 12B,
*     Academic Press, 1976.
*
*     Gill, Adrian E., "Atmosphere−Ocean Dynamics", Academic Press,
*     1982.
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987.
*
*     Hohenkerk, C.Y., & Sinclair, A.T., NAO Technical Note No. 63,
*     1985.
*
*     Rueger, J.M., "Refractive Index Formulae for Electronic Distance
*     Measurement with Radio and Millimetre Waves", in Unisurv Report
*     S−68, School of Surveying and Spatial Information Systems,
```

```
*     University of New South Wales, Sydney, Australia, 2002.
*
*     Stone, Ronald C., P.A.S.P. 108, 1051-1058, 1996.
*
*-
```

```
      SUBROUTINE iau_RM2V ( R, W )
*+
*  - - - - - - - - -
*   i a u _ R M 2 V
*  - - - - - - - - -
*
*  Express an r-matrix as an r-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    rotation matrix
*
*  Returned:
*     W        d(3)      rotation vector (Note 1)
*
*  Notes:
*
*  1) A rotation matrix describes a rotation through some angle about
*     some arbitrary axis called the Euler axis.  The "rotation vector"
*     returned by this routine has the same direction as the Euler axis,
*     and its magnitude is the angle in radians.  (The magnitude and
*     direction can be separated by means of the routine iau_PN.)
*
*  2) If R is null, so is the result.  If R is not a rotation matrix
*     the result is undefined.  R must be proper (i.e. have a positive
*     determinant) and real orthogonal (inverse = transpose).
*
*  3) The reference frame rotates clockwise as seen looking along
*     the rotation vector from the origin.
*
*-
```

```
      SUBROUTINE iau_RV2M ( W, R )
*+
*  - - - - - - - - -
*   i a u _ R V 2 M
*  - - - - - - - - -
*
*  Form the r-matrix corresponding to a given r-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     W        d(3)      rotation vector (Note 1)
*
*  Returned:
*     R        d(3,3)    rotation matrix
*
*  Notes:
*
*  1) A rotation matrix describes a rotation through some angle about
*     some arbitrary axis called the Euler axis.  The "rotation vector"
*     supplied to this routine has the same direction as the Euler axis,
*     and its magnitude is the angle in radians.
*
*  2) If W is null, the unit matrix is returned.
*
*  3) The reference frame rotates clockwise as seen looking along the
*     rotation vector from the origin.
*
*-
```

```
      SUBROUTINE iau_RX ( PHI, R )
*+
*  - - - - - - -
*   i a u _ R X
*  - - - - - - -
*
*  Rotate an r-matrix about the x-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PHI      d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix, rotated
*
*  Notes:
*
*  1) Calling this routine with positive PHI incorporates in the
*     supplied r-matrix R an additional rotation, about the x-axis,
*     anticlockwise as seen looking towards the origin from positive x.
*
*  2) The additional rotation can be represented by this matrix:
*
*         (  1        0            0      )
*         (                               )
*         (  0   + cos(PHI)   + sin(PHI)  )
*         (                               )
*         (  0   - sin(PHI)   + cos(PHI)  )
*
*-
```

```
      SUBROUTINE iau_RXP ( R, P, RP )
*+
*  - - - - - - - -
*   i a u _ R X P
*  - - - - - - - -
*
*  Multiply a p-vector by an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix
*     P        d(3)      p-vector
*
*  Returned:
*     RP       d(3)      R * P
*
*  Called:
*     iau_CP       copy p-vector
*
*-
```

```
      SUBROUTINE iau_RXPV ( R, PV, RPV )
*+
*  - - - - - - - - -
*   i a u _ R X P V
*  - - - - - - - - -
*
*  Multiply a pv-vector by an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     RPV      d(3,2)    R * PV
*
*  Called:
*     iau_RXP      product of r-matrix and p-vector
*
*-
```

```
      SUBROUTINE iau_RXR ( A, B, ATB )
*+
*  - - - - - - - -
*   i a u _ R X R
*  - - - - - - - -
*
*  Multiply two r-matrices.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3,3)    first r-matrix
*     B        d(3,3)    second r-matrix
*
*  Returned:
*     ATB      d(3,3)    A * B
*
*  Called:
*     iau_CR       copy r-matrix
*
*-
```

```
      SUBROUTINE iau_RY ( THETA, R )
*+
*  - - - - - - - -
*   i a u _ R Y
*  - - - - - - - -
*
*  Rotate an r-matrix about the y-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     THETA    d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix, rotated
*
*  Notes:
*
*  1) Calling this routine with positive THETA incorporates in the
*     supplied r-matrix R an additional rotation, about the y-axis,
*     anticlockwise as seen looking towards the origin from positive y.
*
*  2) The additional rotation can be represented by this matrix:
*
*         (  + cos(THETA)     0      - sin(THETA)  )
*         (                                        )
*         (       0           1           0        )
*         (                                        )
*         (  + sin(THETA)     0      + cos(THETA)  )
*
*-
```

```
      SUBROUTINE iau_RZ ( PSI, R )
*+
*  - - - - - - -
*   i a u _ R Z
*  - - - - - - -
*
*  Rotate an r-matrix about the z-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PSI      d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix, rotated
*
*  Notes:
*
*  1) Calling this routine with positive PSI incorporates in the
*     supplied r-matrix R an additional rotation, about the z-axis,
*     anticlockwise as seen looking towards the origin from positive z.
*
*  2) The additional rotation can be represented by this matrix:
*
*         (  + cos(PSI)   + sin(PSI)     0  )
*         (                                 )
*         (  - sin(PSI)   + cos(PSI)     0  )
*         (                                 )
*         (       0            0         1  )
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00 ( DATE1, DATE2, X, Y )
*+
*  - - - - - - - -
*   i a u _ S 0 0
*  - - - - - - - -
*
*  The CIO locator s, positioning the Celestial Intermediate Origin on
*  the equator of the Celestial Intermediate Pole, given the CIP's X,Y
*  coordinates.  Compatible with IAU 2000A precession-nutation.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     X,Y            d      CIP coordinates (Note 3)
*
*  Returned:
*     iau_S00        d      the CIO locator s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The CIO locator s is the difference between the right ascensions
*     of the same point in two systems:  the two systems are the GCRS
*     and the CIP,CIO, and the point is the ascending node of the
*     CIP equator.  The quantity s remains below 0.1 arcsecond
*     throughout 1900-2100.
*
*  3) The series used to compute s is in fact for s+XY/2, where X and Y
*     are the x and y components of the CIP unit vector;  this series is
*     more compact than a direct series for s would be.  This routine
*     requires X,Y to be supplied by the caller, who is responsible for
*     providing values that are consistent with the supplied date.
*
*  4) The model is consistent with the IAU 2000A precession-nutation.
*
*  Called:
*     iau_FAL03    mean anomaly of the Moon
*     iau_FALP03   mean anomaly of the Sun
*     iau_FAF03    mean argument of the latitude of the Moon
*     iau_FAD03    mean elongation of the Moon from the Sun
*     iau_FAOM03   mean longitude of the Moon's ascending node
*     iau_FAVE03   mean longitude of Venus
*     iau_FAE03    mean longitude of Earth
*     iau_FAPA03   general accumulated precession in longitude
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
```

```
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00A ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S 0 0 A
*  - - - - - - - - -
*
*  The CIO locator s, positioning the Celestial Intermediate Origin on
*  the equator of the Celestial Intermediate Pole, using the IAU 2000A
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_S00A       d      the CIO locator s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The CIO locator s is the difference between the right ascensions
*     of the same point in two systems.  The two systems are the GCRS
*     and the CIP,CIO, and the point is the ascending node of the
*     CIP equator.  The CIO locator s remains a small fraction of
*     1 arcsecond throughout 1900-2100.
*
*  3) The series used to compute s is in fact for s+XY/2, where X and Y
*     are the x and y components of the CIP unit vector;  this series is
*     more compact than a direct series for s would be.  The present
*     routine uses the full IAU 2000A nutation model when predicting the
*     CIP position.  Faster results, with no significant loss of
*     accuracy, can be obtained via the routine iau_S00B, which uses
*     instead the IAU 2000B truncated model.
*
*  Called:
*     iau_PNM00A   classical NPB matrix, IAU 2000A
*     iau_BNP2XY    extract CIP X,Y from the BPN matrix
*     iau_S00      the CIO locator s, given X,Y, IAU 2000A
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
```

```
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00B ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S 0 0 B
*  - - - - - - - - -
*
*  The CIO locator s, positioning the Celestial Intermediate Origin on
*  the equator of the Celestial Intermediate Pole, using the IAU 2000B
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_S00B       d      the CIO locator s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The CIO locator s is the difference between the right ascensions
*     of the same point in two systems.  The two systems are the GCRS
*     and the CIP,CIO, and the point is the ascending node of the
*     CIP equator.  The CIO locator s remains a small fraction of
*     1 arcsecond throughout 1900-2100.
*
*  3) The series used to compute s is in fact for s+XY/2, where X and Y
*     are the x and y components of the CIP unit vector;  this series is
*     more compact than a direct series for s would be.  The present
*     routine uses the IAU 2000B truncated nutation model when
*     predicting the CIP position.  The routine iau_S00A uses instead
*     the full IAU 2000A model, but with no significant increase in
*     accuracy and at some cost in speed.
*
*  Called:
*     iau_PNM00B   classical NPB matrix, IAU 2000B
*     iau_BNP2XY    extract CIP X,Y from the BPN matrix
*     iau_S00      the CIO locator s, given X,Y, IAU 2000A
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
```

```
*      IERS Technical Note No. 32, BKG (2004)
*
*_
```

```
      DOUBLE PRECISION FUNCTION iau_S06 ( DATE1, DATE2, X, Y )
*+
*  − − − − − − − − −
*   i a u _ S 0 6
*  − − − − − − − − −
*
*  The CIO locator s, positioning the Celestial Intermediate Origin on
*  the equator of the Celestial Intermediate Pole, given the CIP’s X,Y
*  coordinates.  Compatible with IAU 2006/2000A precession−nutation.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2−part Julian Date (Note 1)
*     X,Y            d      CIP coordinates (Note 3)
*
*  Returned:
*     iau_S06        d      the CIO locator s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      −1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The CIO locator s is the difference between the right ascensions
*     of the same point in two systems:  the two systems are the GCRS
*     and the CIP,CIO, and the point is the ascending node of the
*     CIP equator.  The quantity s remains below 0.1 arcsecond
*     throughout 1900−2100.
*
*  3) The series used to compute s is in fact for s+XY/2, where X and Y
*     are the x and y components of the CIP unit vector;  this series is
*     more compact than a direct series for s would be.  This routine
*     requires X,Y to be supplied by the caller, who is responsible for
*     providing values that are consistent with the supplied date.
*
*  4) The model is consistent with the "P03" precession (Capitaine et
*     al. 2003), adopted by IAU 2006 Resolution 1, 2006, and the
*     IAU 2000A nutation (with P03 adjustments).
*
*  Called:
*     iau_FAL03    mean anomaly of the Moon
*     iau_FALP03   mean anomaly of the Sun
*     iau_FAF03    mean argument of the latitude of the Moon
*     iau_FAD03    mean elongation of the Moon from the Sun
*     iau_FAOM03   mean longitude of the Moon’s ascending node
*     iau_FAVE03   mean longitude of Venus
*     iau_FAE03    mean longitude of Earth
*     iau_FAPA03   general accumulated precession in longitude
*
*  References:
*
*     Capitaine, N., Wallace, P.T. & Chapront, J., 2003, Astron.
```

```
*     Astrophys. 432, 355
*
*     McCarthy, D.D., Petit, G. (eds.) 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S06A ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S 0 6 A
*  - - - - - - - - -
*
*  The CIO locator s, positioning the Celestial Intermediate Origin on
*  the equator of the Celestial Intermediate Pole, using the IAU 2006
*  precession and IAU 2000A nutation models.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_S06A       d      the CIO locator s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The CIO locator s is the difference between the right ascensions
*     of the same point in two systems.  The two systems are the GCRS
*     and the CIP,CIO, and the point is the ascending node of the
*     CIP equator.  The CIO locator s remains a small fraction of
*     1 arcsecond throughout 1900-2100.
*
*  3) The series used to compute s is in fact for s+XY/2, where X and Y
*     are the x and y components of the CIP unit vector;  this series is
*     more compact than a direct series for s would be.  The present
*     routine uses the full IAU 2000A nutation model when predicting the
*     CIP position.
*
*  Called:
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", Astron.Astrophys. 400, 1145-1154 (2003)
*
*     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
*          intermediate origin" (CIO) by IAU 2006 Resolution 2.
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
```

```
*     IERS Technical Note No. 32, BKG
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_S2C ( THETA, PHI, C )
*+
*  - - - - - - - - 
*   i a u _ S 2 C
*  - - - - - - - - 
*
*  Convert spherical coordinates to Cartesian.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*
*  Returned:
*     C        d(3)      direction cosines
*
*-
```

```
      SUBROUTINE iau_S2P ( THETA, PHI, R, P )
*+
*  - - - - - - - -
*   i a u _ S 2 P
*  - - - - - - - -
*
*  Convert spherical polar coordinates to p-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*     R        d         radial distance
*
*  Returned:
*     P        d(3)      Cartesian coordinates
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_SXP      multiply p-vector by scalar
*
*-
```

```
      SUBROUTINE iau_S2PV ( THETA, PHI, R, TD, PD, RD, PV )
*+
*  - - - - - - - - -
*   i a u _ S 2 P V
*  - - - - - - - - -
*
*  Convert position/velocity from spherical to Cartesian coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     THETA    d         longitude angle (radians)
*     PHI      d         latitude angle (radians)
*     R        d         radial distance
*     TD       d         rate of change of THETA
*     PD       d         rate of change of PHI
*     RD       d         rate of change of R
*
*  Returned:
*     PV       d(3,2)    pv-vector
*
*-
```

```
      SUBROUTINE iau_S2XPV ( S1, S2, PV, SPV )
*+
*  - - - - - - - - - -
*   i a u _ S 2 X P V
*  - - - - - - - - - -
*
*  Multiply a pv-vector by two scalars.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     S1       d         scalar to multiply position component by
*     S2       d         scalar to multiply velocity component by
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     SPV      d(3,2)    pv-vector: p scaled by S1, v scaled by S2
*
*  Called:
*     iau_SXP      multiply p-vector by scalar
*
*-
```

```
      SUBROUTINE iau_SEPP ( A, B, S )
*+
*  - - - - - - - - -
*   i a u _ S E P P
*  - - - - - - - - -
*
*  Angular separation between two p-vectors.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     A        d(3)      first p-vector (not necessarily unit length)
*     B        d(3)      second p-vector (not necessarily unit length)
*
*  Returned:
*     S        d         angular separation (radians, always positive)
*
*  Notes:
*
*  1) If either vector is null, a zero result is returned.
*
*  2) The angular separation is most simply formulated in terms of
*     scalar product.  However, this gives poor accuracy for angles
*     near zero and pi.  The present algorithm uses both cross product
*     and dot product, to deliver full accuracy whatever the size of
*     the angle.
*
*  Called:
*     iau_PXP      vector product of two p-vectors
*     iau_PM       modulus of p-vector
*     iau_PDP      scalar product of two p-vectors
*
*-
```

```
      SUBROUTINE iau_SEPS ( AL, AP, BL, BP, S )
*+
*  - - - - - - - - -
*   i a u _ S E P S
*  - - - - - - - - -
*
*  Angular separation between two sets of spherical coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     AL       d         first longitude (radians)
*     AP       d         first latitude (radians)
*     BL       d         second longitude (radians)
*     BP       d         second latitude (radians)
*
*  Returned:
*     S        d         angular separation (radians)
*
*  Called:
*     iau_S2C      spherical coordinates to unit vector
*     iau_SEPP     angular separation between two p-vectors
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_SP00 ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S P 0 0
*  - - - - - - - - -
*
*  The TIO locator s', positioning the Terrestrial Intermediate Origin
*  on the equator of the Celestial Intermediate Pole.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_SP00       d      the TIO locator s' in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The TIO locator s' is obtained from polar motion observations by
*     numerical integration, and so is in essence unpredictable.
*     However, it is dominated by a secular drift of about
*     47 microarcseconds per century, which is the approximation
*     evaluated by the present routine.
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_STARPM ( RA1, DEC1, PMR1, PMD1, PX1, RV1,
     :                        EP1A, EP1B, EP2A, EP2B,
     :                        RA2, DEC2, PMR2, PMD2, PX2, RV2, J )
*+
*  - - - - - - - - - - -
*   i a u _ S T A R P M
*  - - - - - - - - - - -
*
*  Star proper motion:  update star catalog data for space motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RA1       d         right ascension (radians), before
*     DEC1      d         declination (radians), before
*     PMR1      d         RA proper motion (radians/year), before
*     PMD1      d         Dec proper motion (radians/year), before
*     PX1       d         parallax (arcseconds), before
*     RV1       d         radial velocity (km/s, +ve = receding), before
*     EP1A      d         "before" epoch, part A (Note 1)
*     EP1B      d         "before" epoch, part B (Note 1)
*     EP2A      d         "after" epoch, part A (Note 1)
*     EP2B      d         "after" epoch, part B (Note 1)
*
*  Returned:
*     RA2       d         right ascension (radians), after
*     DEC2      d         declination (radians), after
*     PMR2      d         RA proper motion (radians/year), after
*     PMD2      d         Dec proper motion (radians/year), after
*     PX2       d         parallax (arcseconds), after
*     RV2       d         radial velocity (km/s, +ve = receding), after
*     J         i         status:
*                           -1 = system error (should not occur)
*                            0 = no warnings or errors
*                            1 = distance overridden (Note 6)
*                            2 = excessive velocity (Note 7)
*                            4 = solution didn't converge (Note 8)
*                         else = binary logical OR of the above warnings
*
*  Notes:
*
*  1) The starting and ending TDB epochs EP1A+EP1B and EP2A+EP2B are
*     Julian Dates, apportioned in any convenient way between the two
*     parts (A and B).  For example, JD(TDB)=2450123.7 could be
*     expressed in any of these ways, among others:
*
*             EPnA          EPnB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) In accordance with normal star-catalog conventions, the object's
*     right ascension and declination are freed from the effects of
*     secular aberration.  The frame, which is aligned to the catalog
*     equator and equinox, is Lorentzian and centered on the SSB.
*
*     The proper motions are the rate of change of the right ascension
*     and declination at the catalog epoch and are in radians per TDB
*     Julian year.
*
```

```
*     The parallax and radial velocity are in the same frame.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds.
*
*  4) The RA proper motion is in terms of coordinate angle, not true
*     angle.  If the catalog uses arcseconds for both RA and Dec proper
*     motions, the RA proper motion will need to be divided by cos(Dec)
*     before use.
*
*  5) Straight-line motion at constant speed, in the inertial frame,
*     is assumed.
*
*  6) An extremely small (or zero or negative) parallax is interpreted
*     to mean that the object is on the "celestial sphere", the radius
*     of which is an arbitrary (large) value (see the iau_STARPV routine
*     for the value used).  When the distance is overridden in this way,
*     the status, initially zero, has 1 added to it.
*
*  7) If the space velocity is a significant fraction of c (see the
*     constant VMAX in the routine iau_STARPV), it is arbitrarily set to
*     zero.  When this action occurs, 2 is added to the status.
*
*  8) The relativistic adjustment carried out in the iau_STARPV routine
*     involves an iterative calculation.  If the process fails to
*     converge within a set number of iterations, 4 is added to the
*     status.
*
*  Called:
*     iau_STARPV   star catalog data to space motion pv-vector
*     iau_PVU      update a pv-vector
*     iau_PDP      scalar product of two p-vectors
*     iau_PVSTAR   space motion pv-vector to star catalog data
*
*-
```

```
      SUBROUTINE iau_STARPV ( RA, DEC, PMR, PMD, PX, RV, PV, J )
*+
*  - - - - - - - - - - -
*   i a u _ S T A R P V
*  - - - - - - - - - - -
*
*  Convert star catalog coordinates to position+velocity vector.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given (Note 1):
*     RA       d         right ascension (radians)
*     DEC      d         declination (radians)
*     PMR      d         RA proper motion (radians/year)
*     PMD      d         Dec proper motion (radians/year)
*     PX       d         parallax (arcseconds)
*     RV       d         radial velocity (km/s, positive = receding)
*
*  Returned (Note 2):
*     PV       d(3,2)    pv-vector (au, au/day)
*     J        i         status:
*                           0 = no warnings
*                           1 = distance overridden (Note 6)
*                           2 = excessive velocity (Note 7)
*                           4 = solution didn’t converge (Note 8)
*                        else = binary logical OR of the above
*
*  Notes:
*
*  1) The star data accepted by this routine are "observables" for an
*     imaginary observer at the solar-system barycenter.  Proper motion
*     and radial velocity are, strictly, in terms of barycentric
*     coordinate time, TCB.  For most practical applications, it is
*     permissible to neglect the distinction between TCB and ordinary
*     "proper" time on Earth (TT/TAI).  The result will, as a rule, be
*     limited by the intrinsic accuracy of the proper-motion and radial-
*     velocity data;  moreover, the pv-vector is likely to be merely an
*     intermediate result, so that a change of time unit would cancel
*     out overall.
*
*     In accordance with normal star-catalog conventions, the object’s
*     right ascension and declination are freed from the effects of
*     secular aberration.  The frame, which is aligned to the catalog
*     equator and equinox, is Lorentzian and centered on the SSB.
*
*  2) The resulting position and velocity pv-vector is with respect to
*     the same frame and, like the catalog coordinates, is freed from
*     the effects of secular aberration.  Should the "coordinate
*     direction", where the object was located at the catalog epoch, be
*     required, it may be obtained by calculating the magnitude of the
*     position vector PV(1-3,1) dividing by the speed of light in au/day
*     to give the light-time, and then multiplying the space velocity
*     PV(1-3,2) by this light-time and adding the result to PV(1-3,1).
*
*     Summarizing, the pv-vector returned is for most stars almost
*     identical to the result of applying the standard geometrical
*     "space motion" transformation.  The differences, which are the
*     subject of the Stumpff paper referenced below, are:
*
*     (i) In stars with significant radial velocity and proper motion,
*     the constantly changing light-time distorts the apparent proper
*     motion.  Note that this is a classical, not a relativistic,
*     effect.
*
*     (ii) The transformation complies with special relativity.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds; the radial velocity is in km/s, but
```

```
*     the pv-vector result is in au and au/day.
*
*  4) The RA proper motion is in terms of coordinate angle, not true
*     angle.  If the catalog uses arcseconds for both RA and Dec proper
*     motions, the RA proper motion will need to be divided by cos(Dec)
*     before use.
*
*  5) Straight-line motion at constant speed, in the inertial frame,
*     is assumed.
*
*  6) An extremely small (or zero or negative) parallax is interpreted
*     to mean that the object is on the "celestial sphere", the radius
*     of which is an arbitrary (large) value (see the constant PXMIN).
*     When the distance is overridden in this way, the status, initially
*     zero, has 1 added to it.
*
*  7) If the space velocity is a significant fraction of c (see the
*     constant VMAX), it is arbitrarily set to zero.  When this action
*     occurs, 2 is added to the status.
*
*  8) The relativistic adjustment involves an iterative calculation.
*     If the process fails to converge within a set number (IMAX) of
*     iterations, 4 is added to the status.
*
*  9) The inverse transformation is performed by the routine iau_PVSTAR.
*
*  Called:
*     iau_S2PV     spherical coordinates to pv-vector
*     iau_PM       modulus of p-vector
*     iau_ZP       zero p-vector
*     iau_PN       decompose p-vector into modulus and direction
*     iau_PDP      scalar product of two p-vectors
*     iau_SXP      multiply p-vector by scalar
*     iau_PMP      p-vector minus p-vector
*     iau_PPP      p-vector plus p-vector
*
*  Reference:
*
*     Stumpff, P., Astron.Astrophys. 144, 232-240 (1985).
*
*-
```

```
      SUBROUTINE iau_SXP ( S, P, SP )
*+
*  - - - - - - - -
*   i a u _ S X P
*  - - - - - - - -
*
*  Multiply a p-vector by a scalar.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     S        d         scalar
*     P        d(3)      p-vector
*
*  Returned:
*     SP       d(3)      S * P
*
*-
```

```
      SUBROUTINE iau_SXPV ( S, PV, SPV )
*+
*  - - - - - - - - -
*   i a u _ S X P V
*  - - - - - - - - -
*
*  Multiply a pv-vector by a scalar.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     S        d         scalar
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     SPV      d(3,2)    S * PV
*
*  Called:
*     iau_S2XPV    multiply pv-vector by two scalars
*
*-
```

```
      SUBROUTINE iau_TAITT ( TAI1, TAI2, TT1, TT2, J )
*+
*  - - - - - - - - - -
*   i a u _ T A I T T
*  - - - - - - - - - -
*
*  Time scale transformation:  International Atomic Time, TAI, to
*  Terrestrial Time, TT.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date
*
*  Returned:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Note:
*
*     TAI1+TAI2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TAI1 is the Julian
*     Day Number and TAI2 is the fraction of a day.  The returned
*     TT1,TT2 follow suit.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_TAIUT1 ( TAI1, TAI2, DTA, UT11, UT12, J )
*+
*  - - - - - - - - - - -
*   i a u _ T A I U T 1
*  - - - - - - - - - - -
*
*  Time scale transformation:  International Atomic Time, TAI, to
*  Universal Time, UT1.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date
*     DTA          d      UT1-TAI in seconds
*
*  Returned:
*     UT11,UT12    d      UT1 as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TAI1+TAI2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TAI1 is the Julian
*     Day Number and TAI2 is the fraction of a day.  The returned
*     UT11,UT12 follow suit.
*
*  2) The argument DTA, i.e. UT1-TAI, is an observed quantity, and is
*     available from IERS tabulations.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_TAIUTC ( TAI1, TAI2, UTC1, UTC2, J )
*+
*  - - - - - - - - - - -
*   i a u _ T A I U T C
*  - - - - - - - - - - -
*
*  Time scale transformation:  International Atomic Time, TAI, to
*  Coordinated Universal Time, UTC.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date (Note 1)
*
*  Returned:
*     UTC1,UTC2    d      UTC as a 2-part quasi Julian Date (Notes 1-3)
*     J            i      status: +1 = dubious year (Note 4)
*                                  0 = OK
*                                 -1 = unacceptable date
*
*  Notes:
*
*  1) TAI1+TAI2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TAI1 is the Julian
*     Day Number and TAI2 is the fraction of a day.  The returned UTC1
*     and UTC2 form an analogous pair, except that a special convention
*     is used, to deal with the problem of leap seconds - see the next
*     note.
*
*  2) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The convention in the present routine
*     is that the JD day represents UTC days whether the length is
*     86399, 86400 or 86401 SI seconds.  In the 1960-1972 era there were
*     smaller jumps (in either direction) each time the linear UTC(TAI)
*     expression was changed, and these "mini-leaps" are also included
*     in the SOFA convention.
*
*  3) The routine iau_D2DTF can be used to transform the UTC quasi-JD
*     into calendar date and clock time, including UTC leap second
*     handling.
*
*  4) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  Called:
*     iau_UTCTAI   UTC to TAI
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_TCBTDB ( TCB1, TCB2, TDB1, TDB2, J )
*+
*  - - - - - - - - - - -
*   i a u _ T C B T D B
*  - - - - - - - - - - -
*
*  Time scale transformation:  Barycentric Coordinate Time, TCB, to
*  Barycentric Dynamical Time, TDB.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TCB1,TCB2    d      TCB as a 2-part Julian Date
*
*  Returned:
*     TDB1,TDB2    d      TDB as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TCB1+TCB2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TCB1 is the Julian
*     Day Number and TCB2 is the fraction of a day.  The returned
*     TDB1,TDB2 follow suit.
*
*  2) The 2006 IAU General Assembly introduced a conventional linear
*     transformation between TDB and TCB.  This transformation
*     compensates for the drift between TCB and terrestrial time TT,
*     and keeps TDB approximately centered on TT.  Because the
*     relationship between TT and TCB depends on the adopted solar
*     system ephemeris, the degree of alignment between TDB and TT over
*     long intervals will vary according to which ephemeris is used.
*     Former definitions of TDB attempted to avoid this problem by
*     stipulating that TDB and TT should differ only by periodic
*     effects.  This is a good description of the nature of the
*     relationship but eluded precise mathematical formulation.  The
*     conventional linear relationship adopted in 2006 sidestepped
*     these difficulties whilst delivering a TDB that in practice was
*     consistent with values before that date.
*
*  3) TDB is essentially the same as Teph, the time argument for the
*     JPL solar system ephemerides.
*
*  Reference:
*
*     IAU 2006 Resolution B3
*
*-
```

```
      SUBROUTINE iau_TCGTT ( TCG1, TCG2, TT1, TT2, J )
*+
*  - - - - - - - - - -
*   i a u _ T C G T T
*  - - - - - - - - - -
*
*  Time scale transformation:  Geocentric Coordinate Time, TCG, to
*  Terrestrial Time, TT.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TCG1,TCG2    d      TCG as a 2-part Julian Date
*
*  Returned:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Note:
*
*     TCG1+TCG2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TCG1 is the Julian
*     Day Number and TCG2 is the fraction of a day.  The returned
*     TT1,TT2 follow suit.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),.
*     IERS Technical Note No. 32, BKG (2004)
*
*     IAU 2000 Resolution B1.9
*
*-
```

```
      SUBROUTINE iau_TDBTCB ( TDB1, TDB2, TCB1, TCB2, J )
*+
*  - - - - - - - - - - -
*   i a u _ T D B T C B
*  - - - - - - - - - - -
*
*  Time scale transformation:  Barycentric Dynamical Time, TDB, to
*  Barycentric Coordinate Time, TCB.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TDB1,TDB2    d      TDB as a 2-part Julian Date
*
*  Returned:
*     TCB1,TCB2    d      TCB as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TDB1+TDB2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TDB1 is the Julian
*     Day Number and TDB2 is the fraction of a day.  The returned
*     TCB1,TCB2 follow suit.
*
*  2) The 2006 IAU General Assembly introduced a conventional linear
*     transformation between TDB and TCB.  This transformation
*     compensates for the drift between TCB and terrestrial time TT,
*     and keeps TDB approximately centered on TT.  Because the
*     relationship between TT and TCB depends on the adopted solar
*     system ephemeris, the degree of alignment between TDB and TT over
*     long intervals will vary according to which ephemeris is used.
*     Former definitions of TDB attempted to avoid this problem by
*     stipulating that TDB and TT should differ only by periodic
*     effects.  This is a good description of the nature of the
*     relationship but eluded precise mathematical formulation.  The
*     conventional linear relationship adopted in 2006 sidestepped
*     these difficulties whilst delivering a TDB that in practice was
*     consistent with values before that date.
*
*  3) TDB is essentially the same as Teph, the time argument for the
*     JPL solar system ephemerides.
*
*  Reference:
*
*     IAU 2006 Resolution B3
*
*-
```

```
      SUBROUTINE iau_TDBTT ( TDB1, TDB2, DTR, TT1, TT2, J )
*+
*  - - - - - - - - - -
*   i a u _ T D B T T
*  - - - - - - - - - -
*
*  Time scale transformation:  Barycentric Dynamical Time, TDB, to
*  Terrestrial Time, TT.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TDB1,TDB2    d      TDB as a 2-part Julian Date
*     DTR          d      TDB-TT in seconds
*
*  Returned:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TDB1+TDB2 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where TDB1 is the Julian
*     Day Number and TDB2 is the fraction of a day.  The returned
*     TT1,TT2 follow suit.
*
*  2) The argument DTR represents the quasi-periodic component of the
*     GR transformation between TT and TCB.  It is dependent upon the
*     adopted solar-system ephemeris, and can be obtained by numerical
*     integration, by interrogating a precomputed time ephemeris or by
*     evaluating a model such as that implemented in the SOFA routine
*     iau_DTDB.  The quantity is dominated by an annual term of 1.7 ms
*     amplitude.
*
*  3) TDB is essentially the same as Teph, the time argument for the
*     JPL solar system ephemerides.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     IAU 2006 Resolution 3
*
*-
```

```
      SUBROUTINE iau_TF2A ( S, IHOUR, IMIN, SEC, RAD, J )
*+
*  - - - - - - - - -
*   i a u _ T F 2 A
*  - - - - - - - - -
*
*  Convert hours, minutes, seconds to radians.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     S          c      sign:  '-' = negative, otherwise positive
*     IHOUR      i      hours
*     IMIN       i      minutes
*     SEC        d      seconds
*
*  Returned:
*     RAD        d      angle in radians
*     J          i      status:  0 = OK
*                                1 = IHOUR outside range 0-23
*                                2 = IMIN outside range 0-59
*                                3 = SEC outside range 0-59.999...
*
*  Notes:
*
*  1)  If the s argument is a string, only the leftmost character is
*      used and no warning status is provided.
*
*  2)  The result is computed even if any of the range checks fail.
*
*  3)  Negative IHOUR, IMIN and/or SEC produce a warning status, but the
*      absolute value is used in the conversion.
*
*  4)  If there are multiple errors, the status value reflects only the
*      first, the smallest taking precedence.
*
*-
```

```
      SUBROUTINE iau_TF2D ( S, IHOUR, IMIN, SEC, DAYS, J )
*+
*  - - - - - - - - -
*   i a u _ T F 2 D
*  - - - - - - - - -
*
*  Convert hours, minutes, seconds to days.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     S          c      sign:  '-' = negative, otherwise positive
*     IHOUR      i      hours
*     IMIN       i      minutes
*     SEC        d      seconds
*
*  Returned:
*     DAYS       d      interval in days
*     J          i      status:  0 = OK
*                                1 = IHOUR outside range 0-23
*                                2 = IMIN outside range 0-59
*                                3 = SEC outside range 0-59.999...
*
*  Notes:
*
*  1)  If the s argument is a string, only the leftmost character is
*      used and no warning status is provided.
*
*  2)  The result is computed even if any of the range checks fail.
*
*  3)  Negative IHOUR, IMIN and/or SEC produce a warning status, but the
*      absolute value is used in the conversion.
*
*  4)  If there are multiple errors, the status value reflects only the
*      first, the smallest taking precedence.
*
*-
```

```
      SUBROUTINE iau_TPORS ( XI, ETA, A, B, A01, B01, A02, B02, N )
*
*  - - - - - - - - - -
*   i a u _ T P O R S
*  - - - - - - - - - -
*
*  In the tangent plane projection, given the rectangular coordinates
*  of a star and its spherical coordinates, determine the spherical
*  coordinates of the tangent point.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XI,ETA    d       rectangular coordinates of star image (Note 2)
*     A,B       d       star's spherical coordinates (Note 3)
*
*  Returned:
*     A01,B01   d       tangent point's spherical coordinates, Soln. 1
*     A02,B02   d       tangent point's spherical coordinates, Soln. 2
*     N         i       number of solutions:
*                         0 = no solutions returned (Note 5)
*                         1 = only the first solution is useful (Note 6)
*                         2 = both solutions are useful (Note 6)
*
*  Notes:
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the spherical coordinates are observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  If the spherical coordinates are with respect to a
*     right-handed triad, (xi,eta) are also right-handed.  The units of
*     (xi,eta) are, effectively, radians at the tangent point.
*
*  3) All angular arguments are in radians.
*
*  4) The angles A01 and A02 are returned in the range 0-2pi.  The
*     angles B01 and B02 are returned in the range +/-pi, but in the
*     usual, non-pole-crossing, case, the range is +/-pi/2.
*
*  5) Cases where there is no solution can arise only near the poles.
*     For example, it is clearly impossible for a star at the pole
*     itself to have a non-zero xi value, and hence it is meaningless
*     to ask where the tangent point would have to be to bring about
*     this combination of xi and dec.
*
*  6) Also near the poles, cases can arise where there are two useful
*     solutions.  The returned value N indicates whether the second of
*     the two solutions returned is useful;  N=1 indicates only one
*     useful solution, the usual case.
*
*  7) The basis of the algorithm is to solve the spherical triangle PSC,
*     where P is the north celestial pole, S is the star and C is the
*     tangent point.  The spherical coordinates of the tangent point are
*     [a0,b0];  writing rho^2 = (xi^2+eta^2) and r^2 = (1+rho^2), side c
*     is then (pi/2-b), side p is sqrt(xi^2+eta^2) and side s (to be
*     found) is (pi/2-b0).  Angle C is given by sin(C) = xi/rho and
*     cos(C) = eta/rho.  Angle P (to be found) is the longitude
*     difference between star and tangent point (a-a0).
*
*  8) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*         iau_TPXES      iau_TPXEV      xi,eta
*         iau_TPSTS      iau_TPSTV       star
*       > iau_TPORS <    iau_TPORV      origin
```

```
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TPORV ( XI, ETA, V, V01, V02, N )
*
*  - - - - - - - - - -
*   i a u _ T P O R V
*  - - - - - - - - - -
*
*  In the tangent plane projection, given the rectangular coordinates
*  of a star and its direction cosines, determine the direction
*  cosines of the tangent point.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XI,ETA    d       rectangular coordinates of star image (Note 2)
*     V         d(3)    star’s direction cosines (Note 3)
*
*  Returned:
*     V01       d(3)    tangent point’s direction cosines, Solution 1
*     V02       d(3)    tangent point’s direction cosines, Solution 2
*     N         i       number of solutions:
*                         0 = no solutions returned (Note 4)
*                         1 = only the first solution is useful (Note 5)
*                         2 = both solutions are useful (Note 5)
*
*  Notes:
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the direction cosines represent observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  If the direction cosines are with respect to a
*     right-handed triad, (xi,eta) are also right-handed.  The units of
*     (xi,eta) are, effectively, radians at the tangent point.
*
*  3) The vector V must be of unit length or the result will be wrong.
*
*  4) Cases where there is no solution can arise only near the poles.
*     For example, it is clearly impossible for a star at the pole
*     itself to have a non-zero xi value, and hence it is meaningless
*     to ask where the tangent point would have to be.
*
*  5) Also near the poles, cases can arise where there are two useful
*     solutions.  The returned value N indicates whether the second of
*     the two solutions returned is useful;  N=1 indicates only one
*     useful solution, the usual case.
*
*  6) The basis of the algorithm is to solve the spherical triangle PSC,
*     where P is the north celestial pole, S is the star and C is the
*     tangent point.  Calling the celestial spherical coordinates of the
*     star and tangent point (a,b) and (a0,b0) respectively, and writing
*     rho^2 = (xi^2+eta^2) and r^2 = (1+rho^2), and transforming the
*     vector V into (a,b) in the normal way, side c is then (pi/2-b),
*     side p is sqrt(xi^2+eta^2) and side s (to be found) is (pi/2-b0),
*     while angle C is given by sin(C) = xi/rho and cos(C) = eta/rho;
*     angle P (to be found) is (a-a0).  After solving the spherical
*     triangle, the result (a0,b0) can be expressed in vector form as
*     V0.
*
*  7) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*         iau_TPXES      iau_TPXEV      xi,eta
*         iau_TPSTS      iau_TPSTV       star
*         iau_TPORS    > iau_TPORV <    origin
*
*  References:
```

```
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TPSTS ( XI, ETA, A0, B0, A, B )
*+
*  - - - - - - - - - -
*   i a u _ T P S T S
*  - - - - - - - - - -
*
*  In the tangent plane projection, given the star's rectangular
*  coordinates and the spherical coordinates of the tangent point,
*  solve for the spherical coordinates of the star.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XI,ETA    d       rectangular coordinates of star image (Note 2)
*     A0,B0     d       tangent point's spherical coordinates
*
*  Returned:
*     A,B       d       star's spherical coordinates
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the spherical coordinates are observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  If the direction cosines are with respect to a
*     right-handed triad, (xi,eta) are also right-handed.  The units of
*     (xi,eta) are, effectively, radians at the tangent point.
*
*  3) All angular arguments are in radians.
*
*  4) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*         iau_TPXES      iau_TPXEV      xi,eta
*       > iau_TPSTS <    iau_TPSTV       star
*         iau_TPORS      iau_TPORV      origin
*
*  Called:
*     iau_ANP      normalize angle into range 0 to 2pi
*
*  References:
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TPSTV ( XI, ETA, V0, V )
*
*  - - - - - - - - - -
*   i a u _ T P S T V
*  - - - - - - - - - -
*
*  In the tangent plane projection, given the star's rectangular
*  coordinates and the direction cosines of the tangent point, solve
*  for the direction cosines of the star.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XI,ETA    d       rectangular coordinates of star image (Note 2)
*     V0        d(3)    tangent point's direction cosines (Note 4)
*
*  Returned:
*     V         d(3)    star's direction cosines
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the direction cosines represent observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  If the direction cosines are with respect to a
*     right-handed triad, (xi,eta) are also right-handed.  The units of
*     (xi,eta) are, effectively, radians at the tangent point.
*
*  3) The method used is to complete the star vector in the (xi,eta)
*     based triad and normalize it, then rotate the triad to put the
*     tangent point at the pole with the x-axis aligned to zero
*     longitude.  Writing (a0,b0) for the celestial spherical
*     coordinates of the tangent point, the sequence of rotations is
*     (b0-pi/2) around the x-axis followed by (-a0-pi/2) around the
*     z-axis.
*
*  4) If vector V0 is not of unit length, the returned vector V will
*     be wrong.
*
*  5) If vector V0 points at a pole, the returned vector V will be
*     based on the arbitrary assumption that the longitude coordinate
*     of the tangent point is zero.
*
*  6) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*         iau_TPXES      iau_TPXEV      xi,eta
*         iau_TPSTS    > iau_TPSTV <     star
*         iau_TPORS      iau_TPORV      origin
*
*  References:
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TPXES ( A, B, A0, B0, XI, ETA, J )
*+
*  - - - - - - - - - -
*   i a u _ T P X E S
*  - - - - - - - - - -
*
*  In the tangent plane projection, given celestial spherical
*  coordinates for a star and the tangent point, solve for the star's
*  rectangular coordinates in the tangent plane.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     A,B       d       star's spherical coordinates
*     A0,B0     d       tangent point's spherical coordinates
*
*  Returned:
*     XI,ETA    d       rectangular coordinates of star image (Note 2)
*     J         i       status:  0 = OK
*                                1 = star too far from axis
*                                2 = antistar on tangent plane
*                                3 = antistar too far from axis
*
*  Notes:
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the spherical coordinates are observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  For right-handed spherical coordinates, (xi,eta)
*     are also right-handed.  The units of (xi,eta) are, effectively,
*     radians at the tangent point.
*
*  3) All angular arguments are in radians.
*
*  4) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*       > iau_TPXES <    iau_TPXEV       xi,eta
*         iau_TPSTS      iau_TPSTV        star
*         iau_TPORS      iau_TPORV       origin
*
*  References:
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TPXEV ( V, V0, XI, ETA, J )
*+
*  - - - - - - - - - -
*   i a u _ T P X E V
*  - - - - - - - - - -
*
*  In the tangent plane projection, given celestial direction cosines
*  for a star and the tangent point, solve for the star's rectangular
*  coordinates in the tangent plane.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     V         d(3)       direction cosines of star (Note 4)
*     V0        d(3)       direction cosines of tangent point (Note 4)
*
*  Returned:
*     XI,ETA    d          tangent plane coordinates of star
*     J         i          status: 0 = OK
*                                  1 = star too far from axis
*                                  2 = antistar on tangent plane
*                                  3 = antistar too far from axis
*
*  Notes:
*
*  1) The tangent plane projection is also called the "gnomonic
*     projection" and the "central projection".
*
*  2) The eta axis points due north in the adopted coordinate system.
*     If the direction cosines represent observed (RA,Dec), the tangent
*     plane coordinates (xi,eta) are conventionally called the "standard
*     coordinates".  If the direction cosines are with respect to a
*     right-handed triad, (xi,eta) are also right-handed.  The units of
*     (xi,eta) are, effectively, radians at the tangent point.
*
*  3) The method used is to extend the star vector to the tangent
*     plane and then rotate the triad so that (x,y) becomes (xi,eta).
*     Writing (a,b) for the celestial spherical coordinates of the
*     star, the sequence of rotations is (a+pi/2) around the z-axis
*     followed by (pi/2-b) around the x-axis.
*
*  4) If vector V0 is not of unit length, or if vector V is of zero
*     length, the results will be wrong.
*
*  5) If V0 points at a pole, the returned (XI,ETA) will be based on the
*     arbitrary assumption that the longitude coordinate of the tangent
*     point is zero.
*
*  6) This routine is a member of the following set:
*
*         spherical       vector       solve for
*
*         iau_TPXES    > iau_TPXEV <    xi,eta
*         iau_TPSTS      iau_TPSTV       star
*         iau_TPORS      iau_TPORV      origin
*
*  References:
*
*     Calabretta M.R. & Greisen, E.W., 2002, "Representations of
*     celestial coordinates in FITS", Astron.Astrophys. 395, 1077
*
*     Green, R.M., "Spherical Astronomy", Cambridge University Press,
*     1987, Chapter 13.
*
*-
```

```
      SUBROUTINE iau_TR ( R, RT )
*+
*  - - - - - - -
*   i a u _ T R
*  - - - - - - -
*
*  Transpose an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix
*
*  Returned:
*     RT       d(3,3)    transpose
*
*  Called:
*     iau_CR       copy r-matrix
*
*-
```

```
      SUBROUTINE iau_TRXP ( R, P, TRP )
*+
*  - - - - - - - - -
*   i a u _ T R X P
*  - - - - - - - - -
*
*  Multiply a p-vector by the transpose of an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix
*     P        d(3)      p-vector
*
*  Returned:
*     TRP      d(3)      R * P
*
*  Called:
*     iau_TR       transpose r-matrix
*     iau_RXP      product of r-matrix and p-vector
*
*-
```

```
      SUBROUTINE iau_TRXPV ( R, PV, TRPV )
*+
*  - - - - - - - - - - -
*   i a u _ T R X P V
*  - - - - - - - - - - -
*
*  Multiply a pv-vector by the transpose of an r-matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     R        d(3,3)    r-matrix
*     PV       d(3,2)    pv-vector
*
*  Returned:
*     TRPV     d(3,2)    R * PV
*
*  Called:
*     iau_TR       transpose r-matrix
*     iau_RXPV     product of r-matrix and pv-vector
*
*-
```

```
      SUBROUTINE iau_TTTAI ( TT1, TT2, TAI1, TAI2, J )
*+
*  - - - - - - - - - -
*   i a u _ T T T A I
*  - - - - - - - - - -
*
*  Time scale transformation:  Terrestrial Time, TT, to International
*  Atomic Time, TAI.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TT1,TT2      d      TT as a 2-part Julian Date
*
*  Returned:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Note:
*
*     TT1+TT2 is Julian Date, apportioned in any convenient way between
*     the two arguments, for example where TT1 is the Julian Day Number
*     and TT2 is the fraction of a day.  The returned TAI1,TAI2 follow
*     suit.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_TTTCG ( TT1, TT2, TCG1, TCG2, J )
*+
*  - - - - - - - - - - -
*   i a u _ T T T C G
*  - - - - - - - - - - -
*
*  Time scale transformation:  Terrestrial Time, TT, to Geocentric
*  Coordinate Time, TCG.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TT1,TT2      d      TT as a 2-part Julian Date
*
*  Returned:
*     TCG1,TCG2    d      TCG as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Note:
*
*     TT1+TT2 is Julian Date, apportioned in any convenient way between
*     the two arguments, for example where TT1 is the Julian Day Number
*     and TT2 is the fraction of a day.  The returned TCG1,TCG2 follow
*     suit.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     IAU 2000 Resolution B1.9
*
*-
```

```
      SUBROUTINE iau_TTTDB ( TT1, TT2, DTR, TDB1, TDB2, J )
*+
*  - - - - - - - - - -
*   i a u _ T T T D B
*  - - - - - - - - - -
*
*  Time scale transformation:  Terrestrial Time, TT, to Barycentric
*  Dynamical Time, TDB.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     DTR          d      TDB-TT in seconds
*
*  Returned:
*     TDB1,TDB2    d      TDB as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TT1+TT2 is Julian Date, apportioned in any convenient way between
*     the two arguments, for example where TT1 is the Julian Day Number
*     and TT2 is the fraction of a day.  The returned TDB1,TDB2 follow
*     suit.
*
*  2) The argument DTR represents the quasi-periodic component of the
*     GR transformation between TT and TCB.  It is dependent upon the
*     adopted solar-system ephemeris, and can be obtained by numerical
*     integration, by interrogating a precomputed time ephemeris or by
*     evaluating a model such as that implemented in the SOFA routine
*     iau_DTDB.  The quantity is dominated by an annual term of 1.7 ms
*     amplitude.
*
*  3) TDB is essentially the same as Teph, the time argument for the JPL
*     solar system ephemerides.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     IAU 2006 Resolution 3
*
*-
```

```
      SUBROUTINE iau_TTUT1 ( TT1, TT2, DT, UT11, UT12, J )
*+
*  - - - - - - - - - -
*   i a u _ T T U T 1
*  - - - - - - - - - -
*
*  Time scale transformation:  Terrestrial Time, TT, to Universal Time,
*  UT1.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     DT           d      TT-UT1 in seconds
*
*  Returned:
*     UT11,UT12    d      UT1 as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) TT1+TT2 is Julian Date, apportioned in any convenient way between
*     the two arguments, for example where TT1 is the Julian Day Number
*     and TT2 is the fraction of a day.  The returned UT11,UT12 follow
*     suit.
*
*  2) The argument DT is classical Delta T.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_UT1TAI ( UT11, UT12, DTA, TAI1, TAI2, J )
*+
*  - - - - - - - - - - -
*   i a u _ U T 1 T A I
*  - - - - - - - - - - -
*
*  Time scale transformation:  Universal Time, UT1, to International
*  Atomic Time, TAI.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     UT11,UT12    d      UT1 as a 2-part Julian Date
*     DTA          d      UT1-TAI in seconds
*
*  Returned:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) UT11+UT12 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where UT11 is the Julian
*     Day Number and UT12 is the fraction of a day.  The returned
*     TAI1,TAI2 follow suit.
*
*  2) The argument DTA, i.e. UT1-TAI, is an observed quantity, and is
*     available from IERS tabulations.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_UT1TT ( UT11, UT12, DT, TT1, TT2, J )
*+
*  - - - - - - - - - -
*   i a u _ U T 1 T T
*  - - - - - - - - - -
*
*  Time scale transformation:  Universal Time, UT1, to Terrestrial Time,
*  TT.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     UT11,UT12    d      UT1 as a 2-part Julian Date
*     DT           d      TT-UT1 in seconds
*
*  Returned:
*     TT1,TT2      d      TT as a 2-part Julian Date
*     J            i      status:  0 = OK
*
*  Notes:
*
*  1) UT11+UT12 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where UT11 is the Julian
*     Day Number and UT12 is the fraction of a day.  The returned
*     TT1,TT2 follow suit.
*
*  2) The argument DT is classical Delta T.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_UT1UTC ( UT11, UT12, DUT1, UTC1, UTC2, J )
*+
*  - - - - - - - - - - -
*   i a u _ U T 1 U T C
*  - - - - - - - - - - -
*
*  Time scale transformation:  Universal Time, UT1, to Coordinated
*  Universal Time, UTC.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     UT11,UT12    d      UT1 as a 2-part Julian Date (Note 1)
*     DUT1         d      Delta UT1: UT1-UTC in seconds (Note 2)
*
*  Returned:
*     UTC1,UTC2    d      UTC as a 2-part quasi Julian Date (Notes 3,4)
*     J            i      status: +1 = dubious year (Note 5)
*                                  0 = OK
*                                 -1 = unacceptable date
*
*  Notes:
*
*  1) UT11+UT12 is Julian Date, apportioned in any convenient way
*     between the two arguments, for example where UT11 is the Julian
*     Day Number and UT12 is the fraction of a day.  The returned UTC1
*     and UTC2 form an analogous pair, except that a special convention
*     is used, to deal with the problem of leap seconds - see Note 3.
*
*  2) Delta UT1 can be obtained from tabulations provided by the
*     International Earth Rotation and Reference Systems Service.  The
*     value changes abruptly by 1s at a leap second;  however, close to
*     a leap second the algorithm used here is tolerant of the "wrong"
*     choice of value being made.
*
*  3) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The convention in the present routine
*     is that the returned quasi JD day UTC1+UTC2 represents UTC days
*     whether the length is 86399, 86400 or 86401 SI seconds.
*
*  4) The routine iau_D2DTF can be used to transform the UTC quasi-JD
*     into calendar date and clock time, including UTC leap second
*     handling.
*
*  5) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  Called:
*     iau_JD2CAL   JD to Gregorian calendar
*     iau_DAT      delta(AT) = TAI-UTC
*     iau_CAL2JD   Gregorian calendar to JD
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_UTCTAI ( UTC1, UTC2, TAI1, TAI2, J )
*+
*  - - - - - - - - - - -
*   i a u _ U T C T A I
*  - - - - - - - - - - -
*
*  Time scale transformation:  Coordinated Universal Time, UTC, to
*  International Atomic Time, TAI.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     UTC1,UTC2    d      UTC as a 2-part quasi Julian Date (Notes 1-4)
*
*  Returned:
*     TAI1,TAI2    d      TAI as a 2-part Julian Date (Note 5)
*     J            i      status: +1 = dubious year (Note 3)
*                                  0 = OK
*                                 -1 = unacceptable date
*
*  Notes:
*
*  1) UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*     convenient way between the two arguments, for example where UTC1
*     is the Julian Day Number and UTC2 is the fraction of a day.
*
*  2) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The convention in the present routine
*     is that the JD day represents UTC days whether the length is
*     86399, 86400 or 86401 SI seconds.  In the 1960-1972 era there were
*     smaller jumps (in either direction) each time the linear UTC(TAI)
*     expression was changed, and these "mini-leaps" are also included
*     in the SOFA convention.
*
*  3) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  4) The routine iau_DTF2D converts from calendar date and time of day
*     into 2-part Julian Date, and in the case of UTC implements the
*     leap-second-ambiguity convention described above.
*
*  5) The returned TAI1,TAI2 are such that their sum is the TAI Julian
*     Date.
*
*  Called:
*     iau_JD2CAL   JD to Gregorian calendar
*     iau_DAT      delta(AT) = TAI-UTC
*     iau_CAL2JD   Gregorian calendar to JD
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*-
```

```
      SUBROUTINE iau_UTCUT1 ( UTC1, UTC2, DUT1, UT11, UT12, J )
*+
*  - - - - - - - - - - -
*   i a u _ U T C U T 1
*  - - - - - - - - - - -
*
*  Time scale transformation:  Coordinated Universal Time, UTC, to
*  Universal Time, UT1.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical.
*
*  Given:
*     UTC1,UTC2    d      UTC as a 2-part quasi Julian Date (Notes 1-4)
*     DUT1         d      Delta UT1 = UT1-UTC in seconds (Note 5)
*
*  Returned:
*     UT11,UT12    d      UT1 as a 2-part Julian Date (Note 6)
*     J            i      status: +1 = dubious year (Note 3)
*                                  0 = OK
*                                 -1 = unacceptable date
*
*  Notes:
*
*  1) UTC1+UTC2 is quasi Julian Date (see Note 2), apportioned in any
*     convenient way between the two arguments, for example where UTC1
*     is the Julian Day Number and UTC2 is the fraction of a day.
*
*  2) JD cannot unambiguously represent UTC during a leap second unless
*     special measures are taken.  The convention in the present routine
*     is that the JD day represents UTC days whether the length is
*     86399, 86400 or 86401 SI seconds.
*
*  3) The warning status "dubious year" flags UTCs that predate the
*     introduction of the time scale or that are too far in the future
*     to be trusted.  See iau_DAT for further details.
*
*  4) The routine iau_DTF2D converts from calendar date and time of day
*     into 2-part Julian Date, and in the case of UTC implements the
*     leap-second-ambiguity convention described above.
*
*  5) Delta UT1 can be obtained from tabulations provided by the
*     International Earth Rotation and Reference Systems Service.
*     It is the caller's responsibility to supply a DUT1 argument
*     containing the UT1-UTC value that matches the given UTC.
*
*  6) The returned UT11,UT12 are such that their sum is the UT1 Julian
*     Date.
*
*  References:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*  Called:
*     iau_JD2CAL   JD to Gregorian calendar
*     iau_DAT      delta(AT) = TAI-UTC
*     iau_UTCTAI   UTC to TAI
*     iau_TAIUT1   TAI to UT1
*
*-
```

```
      SUBROUTINE iau_XY06 ( DATE1, DATE2, X, Y )
*+
*  - - - - - - - - -
*   i a u _ X Y 0 6
*  - - - - - - - - -
*
*  X,Y coordinates of celestial intermediate pole from series based
*  on IAU 2006 precession and IAU 2000A nutation.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y            d       CIP X,Y coordinates (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The X,Y coordinates are those of the unit vector towards the
*     celestial intermediate pole.  They represent the combined effects
*     of frame bias, precession and nutation.
*
*  3) The fundamental arguments used are as adopted in IERS Conventions
*     (2003) and are from Simon et al. (1994) and Souchay et al. (1999).
*
*  4) This is an alternative to the angles-based method, via the SOFA
*     routine iau_FW2XY and as used in iau_XYS06A for example.  The
*     two methods agree at the 1 microarcsecond level (at present),
*     a negligible amount compared with the intrinsic accuracy of the
*     models.  However, it would be unwise to mix the two methods
*     (angles-based and series-based) in a single application.
*
*  Called:
*     iau_FAL03    mean anomaly of the Moon
*     iau_FALP03   mean anomaly of the Sun
*     iau_FAF03    mean argument of the latitude of the Moon
*     iau_FAD03    mean elongation of the Moon from the Sun
*     iau_FAOM03   mean longitude of the Moon’s ascending node
*     iau_FAME03   mean longitude of Mercury
*     iau_FAVE03   mean longitude of Venus
*     iau_FAE03    mean longitude of Earth
*     iau_FAMA03   mean longitude of Mars
*     iau_FAJU03   mean longitude of Jupiter
*     iau_FASA03   mean longitude of Saturn
*     iau_FAUR03   mean longitude of Uranus
*     iau_FANE03   mean longitude of Neptune
*     iau_FAPA03   general accumulated precession in longitude
*
*  References:
```

```
*
*     Capitaine, N., Wallace, P.T. & Chapront, J., 2003,
*     Astron.Astrophys., 412, 567
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG
*
*     Simon, J.L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G. & Laskar, J., Astron.Astrophys., 1994, 282, 663
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M., 1999,
*     Astron.Astrophys.Supp.Ser. 135, 111
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_XYS00A ( DATE1, DATE2, X, Y, S )
*+
*  - - - - - - - - - - -
*   i a u _ X Y S 0 0 A
*  - - - - - - - - - - -
*
*  For a given TT date, compute the X,Y coordinates of the Celestial
*  Intermediate Pole and the CIO locator s, using the IAU 2000A
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union’s
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y           d    Celestial Intermediate Pole (Note 2)
*     S             d    the CIO locator s (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The CIO locator s (in radians) positions the Celestial
*     Intermediate Origin on the equator of the CIP.
*
*  4) A faster, but slightly less accurate result (about 1 mas for X,Y),
*     can be obtained by using instead the iau_XYS00B routine.
*
*  Called:
*     iau_PNM00A   classical NPB matrix, IAU 2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S00      the CIO locator s, given X,Y, IAU 2000A
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_XYS00B ( DATE1, DATE2, X, Y, S )
*+
*  - - - - - - - - - - -
*   i a u _ X Y S 0 0 B
*  - - - - - - - - - - -
*
*  For a given TT date, compute the X,Y coordinates of the Celestial
*  Intermediate Pole and the CIO locator s, using the IAU 2000B
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y           d    Celestial Intermediate Pole (Note 2)
*     S             d    the CIO locator s (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The CIO locator s (in radians) positions the Celestial
*     Intermediate Origin on the equator of the CIP.
*
*  4) The present routine is faster, but slightly less accurate (about
*     1 mas in X,Y), than the iau_XYS00A routine.
*
*  Called:
*     iau_PNM00B   classical NPB matrix, IAU 2000B
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S00      the CIO locator s, given X,Y, IAU 2000A
*
*  Reference:
*
*     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
*     IERS Technical Note No. 32, BKG (2004)
*
*-
```

```
      SUBROUTINE iau_XYS06A ( DATE1, DATE2, X, Y, S )
*+
*  - - - - - - - - - - -
*   i a u _ X Y S 0 6 A
*  - - - - - - - - - - -
*
*  For a given TT date, compute the X,Y coordinates of the Celestial
*  Intermediate Pole and the CIO locator s, using the IAU 2006
*  precession and IAU 2000A nutation models.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y           d    Celestial Intermediate Pole (Note 2)
*     S             d    the CIO locator s (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The CIO locator s (in radians) positions the Celestial
*     Intermediate Origin on the equator of the CIP.
*
*  4) Series-based solutions for generating X and Y are also available:
*     see Capitaine & Wallace (2006) and iau_XY06.
*
*  Called:
*     iau_PNM06A   classical NPB matrix, IAU 2006/2000A
*     iau_BPN2XY   extract CIP X,Y coordinates from NPB matrix
*     iau_S06      the CIO locator s, given X,Y, IAU 2006
*
*  References:
*
*     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
*
*     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
*
*-
```

```
      SUBROUTINE iau_ZP ( P )
*+
*  - - - - - - - -
*   i a u _ Z P
*  - - - - - - - -
*
*  Zero a p-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Returned:
*     P        d(3)      p-vector
*
*-
```

```
      SUBROUTINE iau_ZPV ( PV )
*+
*  - - - - - - - -
*   i a u _ Z P V
*  - - - - - - - -
*
*  Zero a pv-vector.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Returned:
*     PV       d(3,2)      pv-vector
*
*  Called:
*     iau_ZP       zero p-vector
*
*-
```

```
      SUBROUTINE iau_ZR ( R )
*+
*  - - - - - - -
*   i a u _ Z R
*  - - - - - - -
*
*  Initialize an r-matrix to the null matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Returned:
*     R       d(3,3)    r-matrix
*
*-
```

copyr.lis 2018 January 26

COPYRIGHT NOTICE

Text equivalent to that below appears at the end of every SOFA routine (with one exception). There are small formatting differences between the Fortran and C versions.

The one exception is the "leap second" routine DAT. This uniquely is classified as "user replaceable", and has a mitigated license statement that permits the distribution of local variants under the same name. This measure allows other SOFA routines to call the local variant, which may be file or network based, or otherwise equipped to pick up IERS leap second updates with no need to download new SOFA code.

```
*+----------------------------------------------------------------------
*
*  Copyright (C) 2018
*  Standards Of Fundamental Astronomy Board
*  of the International Astronomical Union.
*
*  =====================
*  SOFA Software License
*  =====================
*
*  NOTICE TO USER:
*
*  BY USING THIS SOFTWARE YOU ACCEPT THE FOLLOWING SIX TERMS AND
*  CONDITIONS WHICH APPLY TO ITS USE.
*
*  1. The Software is owned by the IAU SOFA Board ("SOFA").
*
*  2. Permission is granted to anyone to use the SOFA software for any
*     purpose, including commercial applications, free of charge and
*     without payment of royalties, subject to the conditions and
*     restrictions listed below.
*
*  3. You (the user) may copy and distribute SOFA source code to others,
*     and use and adapt its code and algorithms in your own software,
*     on a world-wide, royalty-free basis.  That portion of your
*     distribution that does not consist of intact and unchanged copies
*     of SOFA source code files is a "derived work" that must comply
*     with the following requirements:
*
*     a) Your work shall be marked or carry a statement that it
*        (i) uses routines and computations derived by you from
*        software provided by SOFA under license to you; and
*        (ii) does not itself constitute software provided by and/or
*        endorsed by SOFA.
*
*     b) The source code of your derived work must contain descriptions
*        of how the derived work is based upon, contains and/or differs
*        from the original SOFA software.
*
*     c) The names of all routines in your derived work shall not
*        include the prefix "iau" or "sofa" or trivial modifications
*        thereof such as changes of case.
*
*     d) The origin of the SOFA components of your derived work must
*        not be misrepresented;  you must not claim that you wrote the
*        original software, nor file a patent application for SOFA
*        software or algorithms embedded in the SOFA software.
*
*     e) These requirements must be reproduced intact in any source
*        distribution and shall apply to anyone to whom you have
*        granted a further right to modify the source code of your
*        derived work.
*
*     Note that, as originally distributed, the SOFA software is
*     intended to be a definitive implementation of the IAU standards,
*     and consequently third-party modifications are discouraged.  All
*     variations, no matter how minor, must be explicitly marked as
```

```
*     such, as explained above.
*
*  4. You shall not cause the SOFA software to be brought into
*     disrepute, either by misuse, or use for inappropriate tasks, or
*     by inappropriate modification.
*
*  5. The SOFA software is provided "as is" and SOFA makes no warranty
*     as to its use or performance.   SOFA does not and cannot warrant
*     the performance or results which the user may obtain by using the
*     SOFA software.  SOFA makes no warranties, express or implied, as
*     to non-infringement of third party rights, merchantability, or
*     fitness for any particular purpose.  In no event will SOFA be
*     liable to the user for any consequential, incidental, or special
*     damages, including any lost profits or lost savings, even if a
*     SOFA representative has been advised of such damages, or for any
*     claim by any third party.
*
*  6. The provision of any version of the SOFA software under the terms
*     and conditions specified herein does not imply that future
*     versions will also be made available under the same terms and
*     conditions.
*
*  In any published work or commercial product which uses the SOFA
*  software directly, acknowledgement (see www.iausofa.org) is
*  appreciated.
*
*  Correspondence concerning SOFA software should be addressed as
*  follows:
*
*      By email:  sofa@ukho.gov.uk
*      By post:   IAU SOFA Center
*                 HM Nautical Almanac Office
*                 UK Hydrographic Office
*                 Admiralty Way, Taunton
*                 Somerset, TA1 2DN
*                 United Kingdom
*
*-----------------------------------------------------------------------
```

consts.lis 2008 September 30

SOFA Fortran constants

These must be used exactly as presented below.

```
*  Pi
      DOUBLE PRECISION DPI
      PARAMETER ( DPI = 3.141592653589793238462643D0 )

*  2Pi
      DOUBLE PRECISION D2PI
      PARAMETER ( D2PI = 6.283185307179586476925287D0 )

*  Radians to hours
      DOUBLE PRECISION DR2H
      PARAMETER ( DR2H = 3.819718634205488058453210D0 )

*  Radians to seconds
      DOUBLE PRECISION DR2S
      PARAMETER ( DR2S = 13750.98708313975701043156D0 )

*  Radians to degrees
      DOUBLE PRECISION DR2D
      PARAMETER ( DR2D = 57.29577951308232087679815D0 )

*  Radians to arc seconds
      DOUBLE PRECISION DR2AS
      PARAMETER ( DR2AS = 206264.8062470963551564734D0 )

*  Hours to radians
      DOUBLE PRECISION DH2R
      PARAMETER ( DH2R = 0.2617993877991494365385536D0 )

*  Seconds to radians
      DOUBLE PRECISION DS2R
      PARAMETER ( DS2R = 7.272205216643039903848712D-5 )

*  Degrees to radians
      DOUBLE PRECISION DD2R
      PARAMETER ( DD2R = 1.745329251994329576923691D-2 )

*  Arc seconds to radians
      DOUBLE PRECISION DAS2R
      PARAMETER ( DAS2R = 4.848136811095359935899141D-6 )
```

SOFA C constants

The constants used by the C version of SOFA are defined in the header file sofam.h.

IAU STANDARDS OF FUNDAMENTAL ASTRONOMY BOARD

Current Membership

John Bangert − United States Naval Observatory, retired
Steven Bell − Her Majesty’s Nautical Almanac Office (HMNAO)
Nicole Capitaine − Paris Observatory
William Folkner − Jet Propulsion Laboratory
Mickaël Gastineau − Paris Observatory, IMCCE
Catherine Hohenkerk − HMNAO (Chair, retired)
Li Jinling − Shanghai Astronomical Observatory
Brian Luzum − United States Naval Observatory (IERS)
Zinovy Malkin − Pulkovo Observatory, St Petersburg
Jeffrey Percival − University of Wisconsin
Wendy Puatua − United States Naval Observatory
Scott Ransom − National Radio Astronomy Observatory
Patrick Wallace − RAL Space, retired

Past Members

Wim Brouw	University of Groningen
Mark Calabretta	Australia Telescope National Facility
Anne−Marie Gontier	Paris Observatory
George Hobbs	Australia Telescope National Facility
George Kaplan	United States Naval Observatory
Dennis McCarthy	United States Naval Observatory
Skip Newhall	Jet Propulsion Laboratory
Jin Wen−Jing	Shanghai Astronomical Observatory

The e−mail for the Board chair is Catherine.Hohenkerk@ukho.gov.uk