T H E

```
        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
```

S O F T W A R E

L I B R A R I E S

International Astronomical Union

Division 1: Fundamental Astronomy

Commission 19: Rotation of the Earth

Standards Of Fundamental Astronomy Review Board

Release 6

2008 December 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 113 "astronomy" routines supported by 52 "vector/matrix" routines, available in both Fortran77 and C implementations.

THE SOFA INITIATIVE

SOFA is an IAU Service which operates under Division 1 (Fundamental Astronomy) and reports through Commission 19 (Rotation of the Earth).

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 SOFA 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 Review 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 Review 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 113 routines (plus one obsolete Fortran routine that now appears under a revised name). The areas addressed include calendars, time scales, ephemerides, precession-nutation, star space-motion, star catalog transformations and geodetic/geocentric transformations.

The "vector-matrix" library, comprising 52 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 will not be revised or updated and will remain 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 file sofa_pn.pdf describes the SOFA tools for precession-nutation and other aspects of Earth attitude and includes example code and (see the appendix) diagrams showing the interrelationships between the routines supporting the latest (IAU 2006/2000A) models.

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 Review 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 Review 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", P. Kenneth Seidelmann (ed.), University Science Books, 1992. Recent developments are documented in the 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.

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

Time scales

   DAT       Delta(AT) (=TAI-UTC) for a given UTC date
   DTDB      TDB-TT

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
   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 space motion

     PVSTAR    space motion pv-vector to star catalog data
     STARPV    star catalog data to space motion pv-vector

  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
     STARPM    proper motion between two epochs

  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


  Obsolete

     C2TCEO    former name of C2TCIO


CALLS: FORTRAN VERSION

   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_DAT    ( IY, IM, ID, FD, DELTAT, J )
 D =  iau_DTDB   ( DATE1, DATE2, UT, ELONG, U, V )
 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 )
 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_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_HFK5Z  ( RH, DH, DATE1, DATE2, R5, D5, DR5, DD5 )
 CALL iau_JD2CAL ( DJ1, DJ2, IY, IM, ID, FD, J )
 CALL iau_JDCALF ( NDP, DJ1, DJ2, IYMDF, J )
 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_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 ( EP01, EP02, EP11, EP12, ZETA, Z, THETA )
    CALL iau_PVSTAR ( PV, RA, DEC, PMR, PMD, PX, RV, J )
    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_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

```
        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 = iauDat    ( iy, im, id, fd, &deltat );
    d = iauDtdb   ( date1, date2, ut, elong, u, v );
    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 );
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 );
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 );
    iauHfk5z  ( rh, dh, date1, date2,
                &r5, &d5, &dr5, &dd5 );
i = iauJd2cal ( dj1, dj2, &iy, &im, &id, &fd );
i = iauJdcalf ( ndp, dj1, dj2, iymdf );
    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 );
    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 ( ep01, ep02, ep11, ep12, &zeta, &z, &theta );
i = iauPvstar ( pv, &ra, &dec, &pmr, &pmd, &px, &rv );
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 );
    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 Review 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 hms
A2AF      decompose radians into d ' "
D2TF      decompose days into hms
```

CALLS: FORTRAN VERSION

```
CALL iau_A2AF  ( NDP, ANGLE, SIGN, IDMSF )
CALL iau_A2TF  ( NDP, ANGLE, SIGN, IHMSF )
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_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 );
   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 );
```

```
iauTr     ( r, rt );
iauTrxp   ( r, p, trp );
iauTrxpv  ( r, pv, trpv );
iauZp     ( p );
iauZpv    ( pv );
iauZr     ( r );
```

```
void iauA2af(int ndp, double angle, char *sign, int idmsf[4])
/*
**  - - - - - - - -
**   i a u A 2 a f
**  - - - - - - - -
**
**  Decompose radians into degrees, arcminutes, arcseconds, fraction.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     ndp     int     resolution (Note 1)
**     angle   double  angle in radians
**
**  Returned:
**     sign    char    '+' or '-'
**     idmsf   int[4]  degrees, arcminutes, arcseconds, fraction
**
**  Called:
**     iauD2tf      decompose days to hms
**
**  Notes:
**
**  1) The argument 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 doubles on the target platform, and
**     the risk of overflowing idmsf[3].  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 int, or ndp=4 if int is
**     only 16 bits.
**
**  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[0]=360 and setting idmsf[0-3] to zero.
**
*/
```

```
void iauA2tf(int ndp, double angle, char *sign, int ihmsf[4])
/*
**  - - - - - - - -
**   i a u A 2 t f
**  - - - - - - - -
**
**  Decompose radians into hours, minutes, seconds, fraction.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     ndp     int     resolution (Note 1)
**     angle   double  angle in radians
**
**  Returned:
**     sign    char    '+' or '-'
**     ihmsf   int[4]  hours, minutes, seconds, fraction
**
**  Called:
**     iauD2tf      decompose days to hms
**
**  Notes:
**
**  1) The argument 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 doubles on the target platform, and
**     the risk of overflowing ihmsf[3].  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 int, or ndp=4 if int is
**     only 16 bits.
**
**  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[0]=24 and setting ihmsf(0-3) to zero.
**
*/
```

```
double iauAnp(double a)
/*
**  - - - - - - -
**   i a u A n p
**  - - - - - - -
**
**  Normalize angle into the range 0 <= a < 2pi.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double     angle (radians)
**
**  Returned (function value):
**              double     angle in range 0-2pi
**
*/
```

```
double iauAnpm(double a)
/*
**  - - - - - - - - -
**   i a u A n p m
**  - - - - - - - - -
**
**  Normalize angle into the range -pi <= a < +pi.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double     angle (radians)
**
**  Returned (function value):
**              double     angle in range +/-pi
**
*/
```

```
void iauBi00(double *dpsibi, double *depsbi, double *dra)
/*
**  - - - - - - - - -
**   i a u B i 0 0
**  - - - - - - - - -
**
**  Frame bias components of IAU 2000 precession-nutation models (part
**  of MHB2000 with additions).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Returned:
**     dpsibi,depsbi  double  longitude and obliquity corrections
**     dra            double  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 mean J2000.0 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.
**
*/
```

```
void iauBp00(double date1, double date2,
             double rb[3][3], double rp[3][3], double rbp[3][3])
/*
**  - - - - - - - - 
**   i a u B p 0 0
**  - - - - - - - - 
**
**  Frame bias and precession, IAU 2000.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double         TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rb           double[3][3]   frame bias matrix (Note 2)
**     rp           double[3][3]   precession matrix (Note 3)
**     rbp          double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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.
**
**  5) It is permissible to re-use the same array in the returned
**     arguments.  The arrays are filled in the order given.
**
**  Called:
**     iauBi00      frame bias components, IAU 2000
**     iauPr00      IAU 2000 precession adjustments
**     iauIr        initialize r-matrix to identity
**     iauRx        rotate around X-axis
**     iauRy        rotate around Y-axis
**     iauRz        rotate around Z-axis
**     iauCr        copy r-matrix
**     iauRxr       product of two r-matrices
**
**  Reference:
**     "Expressions for the Celestial Intermediate Pole and Celestial
**     Ephemeris Origin consistent with the IAU 2000A precession-
**     nutation model", Astronomy & Astrophysics, 400, 1145-1154 (2003)
```

```
**
**     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
**          intermediate origin" (CIO) by IAU 2006 Resolution 2.
**
*/
```

```
void iauBp06(double date1, double date2,
             double rb[3][3], double rp[3][3], double rbp[3][3])
/*
**  - - - - - - - -
**   i a u B p 0 6
**  - - - - - - - -
**
**  Frame bias and precession, IAU 2006.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double         TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rb           double[3][3]   frame bias matrix (Note 2)
**     rp           double[3][3]   precession matrix (Note 3)
**     rbp          double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPfw06     bias-precession F-W angles, IAU 2006
**     iauFw2m      F-W angles to r-matrix
**     iauPmat06    PB matrix, IAU 2006
**     iauTr        transpose r-matrix
**     iauRxr       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
**
*/
```

```
void iauBpn2xy(double rbpn[3][3], double *x, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     rbpn      double[3][3]  celestial-to-true matrix (Note 1)
**
**  Returned:
**     x,y       double        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) The arguments x,y are components of the Celestial Intermediate
**     Pole unit vector in the Geocentric Celestial Reference System.
**
**  Reference:
**
**     "Expressions for the Celestial Intermediate Pole and Celestial
**     Ephemeris Origin consistent with the IAU 2000A precession-
**     nutation model", Astronomy & Astrophysics, 400, 1145-1154
**     (2003)
**
**     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
**          intermediate origin" (CIO) by IAU 2006 Resolution 2.
**
*/
```

```
void iauC2i00a(double date1, double date2, double rc2i[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rc2i        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauC2i00b function.
**
**  Called:
**     iauPnm00a    classical NPB matrix, IAU 2000A
**     iauC2ibpn    celestial-to-intermediate matrix, given NPB matrix
**
**  References:
**
**     "Expressions for the Celestial Intermediate Pole and Celestial
**     Ephemeris Origin consistent with the IAU 2000A precession-
**     nutation model", Astronomy & Astrophysics, 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)
**
*/
```

```
void iauC2i00b(double date1, double date2, double rc2i[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rc2i        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is faster, but slightly less accurate (about
**     1 mas), than the iauC2i00a function.
**
**  Called:
**     iauPnm00b    classical NPB matrix, IAU 2000B
**     iauC2ibpn    celestial-to-intermediate matrix, given NPB matrix
**
**  References:
**
**     "Expressions for the Celestial Intermediate Pole and Celestial
**     Ephemeris Origin consistent with the IAU 2000A precession-
**     nutation model", Astronomy & Astrophysics, 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)
**
*/
```

```
void iauC2i06a(double date1, double date2, double rc2i[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rc2i        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPnm06a    classical NPB matrix, IAU 2006/2000A
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS06       the CIO locator s, Given X,Y, IAU 2006
**     iauC2ixys    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
**
*/
```

```
void iauC2ibpn(double date1, double date2, double rbpn[3][3],
               double rc2i[3][3])
/*
**  - - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**     rbpn        double[3][3] celestial-to-true matrix (Note 2)
**
**  Returned:
**     rc2i        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is in fact
**     specific to the IAU 2000 models.
**
**  Called:
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauC2ixy     celestial-to-intermediate matrix, given X,Y
**
**  References:
**     "Expressions for the Celestial Intermediate Pole and Celestial
**     Ephemeris Origin consistent with the IAU 2000A precession-
**     nutation model", Astronomy & Astrophysics, 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)
**
*/
```

```
void iauC2ixy(double date1, double date2, double x, double y,
              double rc2i[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**     x,y         double       Celestial Intermediate Pole (Note 2)
**
**  Returned:
**     rc2i        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is in fact
**     specific to the IAU 2000 models.
**
**  Called:
**     iauC2ixys    celestial-to-intermediate matrix, given X,Y and s
**     iauS00       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)
**
*/
```

```
void iauC2ixys(double x, double y, double s, double rc2i[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     x,y      double         Celestial Intermediate Pole (Note 1)
**     s        double         the CIO locator s (Note 2)
**
**  Returned:
**     rc2i     double[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:
**     iauIr        initialize r-matrix to identity
**     iauRz        rotate around Z-axis
**     iauRy        rotate around Y-axis
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauC2s(double p[3], double *theta, double *phi)
/*
**  - - - - - - - -
**   i a u C 2 s
**  - - - - - - - -
**
**  P-vector to spherical coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p      double[3]    p-vector
**
**  Returned:
**     theta  double       longitude angle (radians)
**     phi    double       latitude angle (radians)
**
**  Notes:
**
**  1) The vector 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.
**
*/
```

```
void iauC2t00a(double tta, double ttb, double uta, double utb,
               double xp, double yp, double rc2t[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     tta,ttb  double         TT as a 2-part Julian Date (Note 1)
**     uta,utb  double         UT1 as a 2-part Julian Date (Note 1)
**     xp,yp    double         coordinates of the pole (radians, Note 2)
**
**  Returned:
**     rc2t     double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 precision 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 arguments 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 iauC2t00b function.
**
**  Called:
**     iauC2i00a    celestial-to-intermediate matrix, IAU 2000A
**     iauEra00     Earth rotation angle, IAU 2000
**     iauSp00      the TIO locator s', IERS 2000
```

```
**     iauPom00     polar motion matrix
**     iauC2tcio    form CIO-based celestial-to-terrestrial matrix
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauC2t00b(double tta, double ttb, double uta, double utb,
               double xp, double yp, double rc2t[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     tta,ttb  double         TT as a 2-part Julian Date (Note 1)
**     uta,utb  double         UT1 as a 2-part Julian Date (Note 1)
**     xp,yp    double         coordinates of the pole (radians, Note 2)
**
**  Returned:
**     rc2t     double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 precision 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 arguments 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 function is faster, but slightly less accurate (about
**     1 mas), than the iauC2t00a function.
**
**  Called:
**     iauC2i00b    celestial-to-intermediate matrix, IAU 2000B
**     iauEra00     Earth rotation angle, IAU 2000
**     iauPom00     polar motion matrix
```

```
**     iauC2tcio    form CIO-based celestial-to-terrestrial matrix
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauC2t06a(double tta, double ttb, double uta, double utb,
               double xp, double yp, double rc2t[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     tta,ttb  double         TT as a 2-part Julian Date (Note 1)
**     uta,utb  double         UT1 as a 2-part Julian Date (Note 1)
**     xp,yp    double         coordinates of the pole (radians, Note 2)
**
**  Returned:
**     rc2t     double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 precision 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 arguments 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:
**     iauC2i06a    celestial-to-intermediate matrix, IAU 2006/2000A
**     iauEra00     Earth rotation angle, IAU 2000
**     iauSp00      the TIO locator s', IERS 2000
**     iauPom00     polar motion matrix
**     iauC2tcio    form CIO-based celestial-to-terrestrial matrix
```

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

```
void iauC2tcio(double rc2i[3][3], double era, double rpom[3][3],
               double rc2t[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     rc2i     double[3][3]    celestial-to-intermediate matrix
**     era      double          Earth rotation angle
**     rpom     double[3][3]    polar-motion matrix
**
**  Returned:
**     rc2t     double[3][3]    celestial-to-terrestrial matrix
**
**  Notes:
**
**  1) This function 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 function 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:
**     iauCr        copy r-matrix
**     iauRz        rotate around Z-axis
**     iauRxr       product of two r-matrices
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG
**
*/
```

```
void iauC2teqx(double rbpn[3][3], double gst, double rpom[3][3],
               double rc2t[3][3])
/*
**  - - - - - - - - - - 
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     rbpn     double[3][3]    celestial-to-true matrix
**     gst      double          Greenwich (apparent) Sidereal Time
**     rpom     double[3][3]    polar-motion matrix
**
**  Returned:
**     rc2t     double[3][3]    celestial-to-terrestrial matrix (Note 2)
**
**  Notes:
**
**  1) This function 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 function
**     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:
**     iauCr        copy r-matrix
**     iauRz        rotate around Z-axis
**     iauRxr       product of two r-matrices
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauC2tpe(double tta, double ttb, double uta, double utb,
              double dpsi, double deps, double xp, double yp,
              double rc2t[3][3])
/*
**  - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     tta,ttb    double        TT as a 2-part Julian Date (Note 1)
**     uta,utb    double        UT1 as a 2-part Julian Date (Note 1)
**     dpsi,deps  double        nutation (Note 2)
**     xp,yp      double        coordinates of the pole (radians, Note 3)
**
**  Returned:
**     rc2t       double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 precision 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) The arguments 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 function is in fact
**     specific to the IAU 2000 models.
**
**  Called:
**     iauPn00      bias/precession/nutation results, IAU 2000
**     iauGmst00    Greenwich mean sidereal time, IAU 2000
**     iauSp00      the TIO locator s', IERS 2000
**     iauEe00      equation of the equinoxes, IAU 2000
**     iauPom00     polar motion matrix
**     iauC2teqx    form equinox-based celestial-to-terrestrial matrix
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauC2txy(double tta, double ttb, double uta, double utb,
              double x, double y, double xp, double yp,
              double rc2t[3][3])
/*
**  - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     tta,ttb  double         TT as a 2-part Julian Date (Note 1)
**     uta,utb  double         UT1 as a 2-part Julian Date (Note 1)
**     x,y      double         Celestial Intermediate Pole (Note 2)
**     xp,yp    double         coordinates of the pole (radians, Note 3)
**
**  Returned:
**     rc2t     double[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 o
**     these ways, among others:
**
**             uta            utb
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 precision 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) The arguments 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 function is in fact
**     specific to the IAU 2000 models.
```

```
**
**  Called:
**     iauC2ixy     celestial-to-intermediate matrix, given X,Y
**     iauEra00     Earth rotation angle, IAU 2000
**     iauSp00      the TIO locator s', IERS 2000
**     iauPom00     polar motion matrix
**     iauC2tcio    form CIO-based celestial-to-terrestrial matrix
**
** Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
int iauCal2jd(int iy, int im, int id, double *djm0, double *djm)
/*
**  - - - - - - - - - -
**   i a u C a l 2 j d
**  - - - - - - - - - -
**
**  Gregorian Calendar to Julian Date.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     iy,im,id  int     year, month, day in Gregorian calendar (Note 1)
**
**  Returned:
**     djm0      double  MJD zero-point: always 2400000.5
**     djm       double  Modified Julian Date for 0 hrs
**
**  Returned (function value):
**               int     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).
**
*/
```

```
void iauCp(double p[3], double c[3])
/*
**  - - - - - -
**   i a u C p
**  - - - - - -
**
**  Copy a p-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p        double[3]     p-vector to be copied
**
**  Returned:
**     c        double[3]     copy
**
*/
```

```
void iauCpv(double pv[2][3], double c[2][3])
/*
**  - - - - - - -
**   i a u C p v
**  - - - - - - -
**
**  Copy a position/velocity vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     pv     double[2][3]    position/velocity vector to be copied
**
**  Returned:
**     c      double[2][3]    copy
**
**  Called:
**     iauCp        copy p-vector
**
*/
```

```
void iauCr(double r[3][3], double c[3][3])
/*
**  - - - - - -
**   i a u C r
**  - - - - - -
**
**  Copy an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    r-matrix to be copied
**
**  Returned:
**   char[]     double[3][3]    copy
**
**  Called:
**     iauCp        copy p-vector
**
*/
```

```
void iauD2tf(int ndp, double days, char *sign, int ihmsf[4])
/*
**  - - - - - - - -
**   i a u D 2 t f
**  - - - - - - - -
**
**  Decompose days to hours, minutes, seconds, fraction.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     ndp     int     resolution (Note 1)
**     days    double  interval in days
**
**  Returned:
**     sign    char    '+' or '-'
**     ihmsf   int[4]  hours, minutes, seconds, fraction
**
**  Notes:
**
**  1) The argument 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 on the target platform, and
**     the risk of overflowing ihmsf[3].  On a typical platform, for
**     days up to 1.0, 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 int, or ndp=4 if int is
**     only 16 bits.
**
**  3) The absolute value of days may exceed 1.0.  In cases where it
**     does not, it is up to the caller to test for and handle the
**     case where days is very nearly 1.0 and rounds up to 24 hours,
**     by testing for ihms[0]=24 and setting ihmsf[0-3] to zero.
**
*/
```

```
int iauDat(int iy, int im, int id, double fd, double *deltat )
/*
**  - - - - - - - -
**   i a u D a t
**  - - - - - - - -
**
**  For a given UTC date, calculate delta(AT) = TAI-UTC.
**
**     :------------------------------------------:
**     :                                          :
**     :                 IMPORTANT                :
**     :                                          :
**     :  A new version of this function must be  :
**     :  produced whenever a new leap second is  :
**     :  announced.  There are four items to     :
**     :  change on each such occasion:           :
**     :                                          :
**     :  1) A new line must be added to the set  :
**     :     of statements that initialize the    :
**     :     array "changes".                     :
**     :                                          :
**     :  2) The parameter IYV must be set to     :
**     :     the current year.                    :
**     :                                          :
**     :  3) The "Latest leap second" comment     :
**     :     below must be set to the new leap    :
**     :     second date.                         :
**     :                                          :
**     :  4) The "This revision" comment, later,  :
**     :     must be set to the current date.     :
**     :                                          :
**     :  Change (2) must also be carried out     :
**     :  whenever the function is re-issued,     :
**     :  even if no leap seconds have been       :
**     :  added.                                  :
**     :                                          :
**     :  Latest leap second:  2008 December 31   :
**     :                                          :
**     :__________________________________________:
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     iy     int      UTC:  year (Notes 1 and 2)
**     im     int            month (Note 2)
**     id     int            day (Notes 2 and 3)
**     fd     double         fraction of day (Note 4)
**
**  Returned:
**     deltat double   TAI minus UTC, seconds
**
**  Returned (function value):
**            int      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)
**
**  Notes:
**
**  1) UTC began at 1960 January 1.0 (JD 2436934.5) and it is improper
**     to call the function 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 function (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 UTC-TAI 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 UTC-TAI 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
**      function, 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 (zero to less than 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.
**
**   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:
**      iauCal2jd    Gregorian calendar to Julian Day number
**
*/
```

```
double iauDtdb(double date1, double date2,
               double ut, double elong, double u, double 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      <-  iauDtdb  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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double  date, TDB (Notes 1-3)
**     ut            double  universal time (UT1, fraction of one day)
**     elong         double  longitude (east positive, radians)
**     u             double  distance from Earth spin axis (km)
**     v             double  distance north of equatorial plane (km)
**
**  Returned (function value):
**                   double  TDB-TT (seconds)
**
**  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function,
**     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 function 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 C function 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 1e-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 function 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 function 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).
**
*/
```

```
double iauEe00(double date1, double date2, double epsa, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**     epsa         double    mean obliquity (Note 2)
**     dpsi         double    nutation in longitude (Note 3)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauEect00    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 iauEe00a(double date1, double date2)
/*
**  - - - - - - - - - -
**   i a u E e 0 0 a
**  - - - - - - - - - -
**
**  Equation of the equinoxes, compatible with IAU 2000 resolutions.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPr00      IAU 2000 precession adjustments
**     iauObl80     mean obliquity, IAU 1980
**     iauNut00a    nutation, IAU 2000A
**     iauEe00      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 iauEe00b(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPr00      IAU 2000 precession adjustments
**     iauObl80     mean obliquity, IAU 1980
**     iauNut00b    nutation, IAU 2000B
**     iauEe00      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 iauEe06a(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauAnpm      normalize angle into range +/- pi
**     iauGst06a    Greenwich apparent sidereal time, IAU 2006/2000A
**     iauGmst06    Greenwich mean sidereal time, IAU 2006
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), 2004, IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG
**
*/
```

```
double iauEect00(double date1, double date2)
/*
**  - - - - - - - - - - -
**   i a u E e c t 0 0
**  - - - - - - - - - - -
**
**  Equation of the equinoxes complementary terms, consistent with
**  IAU 2000 resolutions.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function computes CT in the above expression,
**     compatible with IAU 2000 resolutions (Capitaine et al., 2002, and
**     IERS Conventions 2003).
```

```
**
**  Called:
**     iauFal03     mean anomaly of the Moon
**     iauFalp03    mean anomaly of the Sun
**     iauFaf03     mean argument of the latitude of the Moon
**     iauFad03     mean elongation of the Moon from the Sun
**     iauFaom03    mean longitude of the Moon's ascending node
**     iauFave03    mean longitude of Venus
**     iauFae03     mean longitude of Earth
**     iauFapa03    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", Astronomy &
**     Astrophysics, 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)
**
*/
```

```
int iauEform ( int n, double *a, double *f )
/*
**  - - - - - - - - -
**   i a u E f o r m
**  - - - - - - - - -
**
**  Earth reference ellipsoids.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards of Fundamental Astronomy) software collection.
**
**  Status:  canonical.
**
**  Given:
**     n       int      ellipsoid identifier (Note 1)
**
**  Returned:
**     a       double   equatorial radius (meters, Note 2)
**     f       double   flattening (Note 2)
**
**  Returned (function value):
**            int      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
**
**     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).
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     p220.
**
*/
```

```
double iauEo06a(double date1, double date2)
/*
**  - - - - - - - - -
**   i a u E o 0 6 a
**  - - - - - - - - -
**
**  Equation of the origins, IAU 2006 precession and IAU 2000A nutation.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPnm06a    classical NPB matrix, IAU 2006/2000A
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS06       the CIO locator s, given X,Y, IAU 2006
**     iauEors      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 iauEors(double rnpb[3][3], double s)
/*
**  - - - - - - - -
**   i a u E o r s
**  - - - - - - - -
**
**  Equation of the origins, given the classical NPB matrix and the
**  quantity s.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     rnpb  double[3][3]  classical nutation x precession x bias matrix
**     s     double        the quantity s (the CIO locator)
**
**  Returned (function value):
**           double        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 iauEpb(double dj1, double dj2)
/*
**  - - - - - - -
**   i a u E p b
**  - - - - - - -
**
**  Julian Date to Besselian Epoch.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     dj1,dj2    double     Julian Date (see note)
**
**  Returned (function value):
**                double     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.
**
*/
```

```
void iauEpb2jd(double epb, double *djm0, double *djm)
/*
**  - - - - - - - - - -
**   i a u E p b 2 j d
**  - - - - - - - - - -
**
**  Besselian Epoch to Julian Date.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     epb      double    Besselian Epoch (e.g. 1957.3D0)
**
**  Returned:
**     djm0     double    MJD zero-point: always 2400000.5
**     djm      double    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 iauEpj(double dj1, double dj2)
/*
**  - - - - - - -
**   i a u E p j
**  - - - - - - -
**
**  Julian Date to Julian Epoch.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     dj1,dj2    double     Julian Date (see note)
**
**  Returned (function value):
**                double     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.
**
*/
```

```
void iauEpj2jd(double epj, double *djm0, double *djm)
/*
**  - - - - - - - - - -
**   i a u E p j 2 j d
**  - - - - - - - - - -
**
**  Julian Epoch to Julian Date.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     epj      double    Julian Epoch (e.g. 1996.8D0)
**
**  Returned:
**     djm0     double    MJD zero-point: always 2400000.5
**     djm      double    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.
**
*/
```

```
int iauEpv00(double date1, double date2,
             double pvh[2][3], double pvb[2][3])
/*
**  - - - - - - - - -
**   i a u E p v 0 0
**  - - - - - - - - -
**
**  Earth position and velocity, heliocentric and barycentric, with
**  respect to the Barycentric Celestial Reference System.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double        TDB date (Note 1)
**
**  Returned:
**     pvh          double[2][3]  heliocentric Earth position/velocity
**     pvb          double[2][3]  barycentric Earth position/velocity
**
**  Returned (function value):
**                  int           status: 0 = OK
**                                       +1 = warning: date outside
**                                            the range 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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 date has been expressed.
**
**     n.b. TT can be used instead of TDB in most applications.
**
**  2) On return, the arrays pvh and pvb contain the following:
**
**        pvh[0][0]  x       }
**        pvh[0][1]  y       } heliocentric position, AU
**        pvh[0][2]  z       }
**
**        pvh[1][0]  xdot    }
**        pvh[1][1]  ydot    } heliocentric velocity, AU/d
**        pvh[1][2]  zdot    }
**
**        pvb[0][0]  x       }
**        pvb[0][1]  y       } barycentric position, AU
**        pvb[0][2]  z       }
**
**        pvb[1][0]  xdot    }
**        pvb[1][1]  ydot    } barycentric velocity, AU/d
**        pvb[1][2]  zdot    }
**
**     The vectors are with respect to the Barycentric Celestial
```

```
**     Reference System.  The time unit is one day in TDB.
**
**  3) The function 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.
**
**  5) It is permissible to use the same array for pvh and pvb, which
**     will receive the barycentric values.
**
*/
```

```
double iauEqeq94(double date1, double date2)
/*
**  - - - - - - - - - -
**   i a u E q e q 9 4
**  - - - - - - - - - -
**
**  Equation of the equinoxes, IAU 1994 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double     TDB date (Note 1)
**
**  Returned (function value):
**                   double     equation of the equinoxes (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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauNut80     nutation, IAU 1980
**     iauObl80     mean obliquity, IAU 1980
**
**  References:
**
**     IAU Resolution C7, Recommendation 3 (1994).
**
**     Capitaine, N. & Gontier, A.-M., 1993, Astron. Astrophys., 275,
**     645-650.
**
*/
```

```
double iauEra00(double dj1, double dj2)
/*
**  - - - - - - - - -
**   i a u E r a 0 0
**  - - - - - - - - -
**
**  Earth rotation angle (IAU 2000 model).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     dj1,dj2   double    UT1 as a 2-part Julian Date (see note)
**
**  Returned (function value):
**               double    Earth rotation angle (radians), range 0-2pi
**
**  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 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:
**     iauAnp       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 iauFad03(double t)
/*
**  - - - - - - - - - -
**   i a u F a d 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean elongation of the Moon from the Sun.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFae03(double t)
/*
**  - - - - - - - - -
**   i a u F a e 0 3
**  - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Earth.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFaf03(double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFaju03(double t)
/*
**  - - - - - - - - - -
**   i a u F a j u 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Jupiter.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFal03(double t)
/*
**  - - - - - - - - -
**   i a u F a l 0 3
**  - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean anomaly of the Moon.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFalp03(double t)
/*
**  - - - - - - - - - -
**   i a u F a l p 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean anomaly of the Sun.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFama03(double t)
/*
**  - - - - - - - - - -
**   i a u F a m a 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Mars.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFame03(double t)
/*
**  - - - - - - - - - - -
**   i a u F a m e 0 3
**  - - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Mercury.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFane03(double t)
/*
**  - - - - - - - - - -
**   i a u F a n e 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Neptune.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFaom03(double t)
/*
**  - - - - - - - - - -
**   i a u F a o m 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of the Moon's ascending node.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFapa03(double t)
/*
**  - - - - - - - - - -
**   i a u F a p a 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  general accumulated precession in longitude.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFasa03(double t)
/*
**  - - - - - - - - - -
**   i a u F a s a 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Saturn.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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 iauFaur03(double t)
/*
**  - - - - - - - - - -
**   i a u F a u r 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Uranus.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned  (function value):
**           double    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 iauFave03(double t)
/*
**  - - - - - - - - - -
**   i a u F a v e 0 3
**  - - - - - - - - - -
**
**  Fundamental argument, IERS Conventions (2003):
**  mean longitude of Venus.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     t     double    TDB, Julian centuries since J2000.0 (Note 1)
**
**  Returned (function value):
**           double    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
**
*/
```

```
void iauFk52h(double r5, double d5,
              double dr5, double dd5, double px5, double rv5,
              double *rh, double *dh,
              double *drh, double *ddh, double *pxh, double *rvh)
/*
**  - - - - - - - - -
**   i a u F k 5 2 h
**  - - - - - - - - -
**
**  Transform FK5 (J2000.0) star data into the Hipparcos system.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given (all FK5, equinox J2000.0, epoch J2000.0):
**     r5      double    RA (radians)
**     d5      double    Dec (radians)
**     dr5     double    proper motion in RA (dRA/dt, rad/Jyear)
**     dd5     double    proper motion in Dec (dDec/dt, rad/Jyear)
**     px5     double    parallax (arcsec)
**     rv5     double    radial velocity (km/s, positive = receding)
**
**  Returned (all Hipparcos, epoch J2000.0):
**     rh      double    RA (radians)
**     dh      double    Dec (radians)
**     drh     double    proper motion in RA (dRA/dt, rad/Jyear)
**     ddh     double    proper motion in Dec (dDec/dt, rad/Jyear)
**     pxh     double    parallax (arcsec)
**     rvh     double    radial velocity (km/s, positive = receding)
**
**  Notes:
**
**  1) This function 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 iauH2fk5, iauFk5hz, iauHfk5z.
**
**  Called:
**     iauStarpv    star catalog data to space motion pv-vector
**     iauFk5hip    FK5 to Hipparcos rotation and spin
**     iauRxp       product of r-matrix and p-vector
**     iauPxp       vector product of two p-vectors
**     iauPpp       p-vector plus p-vector
**     iauPvstar    space motion pv-vector to star catalog data
**
**  Reference:
**
**     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
**
*/
```

```
void iauFk5hip(double r5h[3][3], double s5h[3])
/*
**  - - - - - - - - - -
**   i a u F k 5 h i p
**  - - - - - - - - - -
**
**  FK5 to Hipparcos rotation and spin.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Returned:
**     r5h   double[3][3]  r-matrix: FK5 rotation wrt Hipparcos (Note 2)
**     s5h   double[3]     r-vector: FK5 spin wrt Hipparcos (Note 3)
**
**  Notes:
**
**  1) This function 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:
**     iauRv2m      r-vector to r-matrix
**
**  Reference:
**
**     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
**
*/
```

```
void iauFk5hz(double r5, double d5, double date1, double date2,
              double *rh, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     r5           double   FK5 RA (radians), equinox J2000.0, at date
**     d5           double   FK5 Dec (radians), equinox J2000.0, at date
**     date1,date2  double   TDB date (Notes 1,2)
**
**  Returned:
**     rh           double   Hipparcos RA (radians)
**     dh           double   Hipparcos Dec (radians)
**
**  Notes:
**
**  1) This function 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 function requires the date
**     at which the position in the FK5 system was determined.
**
**  2) 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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is in the Hipparcos
**     reference system but at date date1+date2.
**
**  5) See also iauFk52h, iauH2fk5, iauHfk5z.
**
**  Called:
**     iauS2c       spherical coordinates to unit vector
**     iauFk5hip    FK5 to Hipparcos rotation and spin
**     iauSxp       multiply p-vector by scalar
**     iauRv2m      r-vector to r-matrix
**     iauTrxp      product of transpose of r-matrix and p-vector
**     iauPxp       vector product of two p-vectors
**     iauC2s       p-vector to spherical
**     iauAnp       normalize angle into range 0 to 2pi
**
**  Reference:
```

```
**
**      F.Mignard & M.Froeschle, 2000, Astron.Astrophys. 354, 732-739.
**
*/
```

```
void iauFw2m(double gamb, double phib, double psi, double eps,
             double r[3][3])
/*
**  - - - - - - - - -
**   i a u F w 2 m
**  - - - - - - - - -
**
**  Form rotation matrix given the Fukushima-Williams angles.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     gamb     double         F-W angle gamma_bar (radians)
**     phib     double         F-W angle phi_bar (radians)
**     psi      double         F-W angle psi (radians)
**     eps      double         F-W angle epsilon (radians)
**
**  Returned:
**     r        double[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 function.
**
**     o  To obtain the precession x frame bias matrix, generate the
**        four precession angles and call the present function.
**
**     o  To obtain the frame bias matrix, generate the four precession
**        angles for date J2000.0 and call the present function.
**
**     The nutation-only and precession-only matrices can if necessary
**     be obtained by combining these three appropriately.
**
**  Called:
**     iauIr        initialize r-matrix to identity
**     iauRz        rotate around Z-axis
**     iauRx        rotate around X-axis
**
**  Reference:
**
**     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
**
*/
```

```
void iauFw2xy(double gamb, double phib, double psi, double eps,
              double *x, double *y)
/*
**  - - - - - - - - -
**   i a u F w 2 x y
**  - - - - - - - - -
**
**  CIP X,Y given Fukushima-Williams bias-precession-nutation angles.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     gamb     double    F-W angle gamma_bar (radians)
**     phib     double    F-W angle phi_bar (radians)
**     psi      double    F-W angle psi (radians)
**     eps      double    F-W angle epsilon (radians)
**
**  Returned:
**     x,y      double    CIP X,Y ("radians")
**
**  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)
**
**     X,Y are elements (3,1) and (3,2) of the matrix.
**
**  Called:
**     iauFw2m      F-W angles to r-matrix
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**
**  Reference:
**
**     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
**
*/
```

```
int iauGc2gd ( int n, double xyz[3],
               double *elong, double *phi, double *height )
/*
**  - - - - - - - - -
**   i a u G c 2 g d
**  - - - - - - - - -
**
**  Transform geocentric coordinates to geodetic using the specified
**  reference ellipsoid.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards of Fundamental Astronomy) software collection.
**
**  Status:  canonical transformation.
**
**  Given:
**     n       int        ellipsoid identifier (Note 1)
**     xyz     double[3]  geocentric vector (Note 2)
**
**  Returned:
**     elong   double     longitude (radians, east +ve)
**     phi     double     latitude (geodetic, radians, Note 3)
**     height  double     height above ellipsoid (geodetic, Notes 2,3)
**
**  Returned (function value):
**            int       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
**
**     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 -1 means that the identifier n is illegal.  An
**     error status -2 is theoretically impossible.  In all error cases,
**     phi and height are both set to -1e9.
**
**  4) The inverse transformation is performed in the function iauGd2c.
**
**  Called:
**     iauEform     Earth reference ellipsoids
**     iauGc2gde    geocentric to geodetic transformation, general
**
*/
```

```
int iauGc2gde ( double a, double f, double xyz[3],
                double *elong, double *phi, double *height )
/*
**  - - - - - - - - - -
**   i a u G c 2 g d e
**  - - - - - - - - - -
**
**  Transform geocentric coordinates to geodetic for a reference
**  ellipsoid of specified form.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     a       double     equatorial radius (Notes 2,4)
**     f       double     flattening (Note 3)
**     xyz     double[3]  geocentric vector (Note 4)
**
**  Returned:
**     elong   double     longitude (radians, east +ve)
**     phi     double     latitude (geodetic, radians)
**     height  double     height above ellipsoid (geodetic, Note 4)
**
**  Returned (function value):
**            int       status:
**                          0 = OK
**                         -1 = illegal a
**                         -2 = illegal f
**
**  Notes:
**
**  1) This function is based on the GCONV2H Fortran 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 (status < 0), elong, phi and height are
**     unchanged.
**
**  6) The inverse transformation is performed in the function
**     iauGd2gce.
**
**  7) The transformation for a standard ellipsoid (such as WGS84) can
**     more conveniently be performed by calling iauGc2gd, 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
**
*/
```

```
int iauGd2gc ( int n, double elong, double phi, double height,
               double xyz[3] )
/*
**  - - - - - - - - -
**   i a u G d 2 g c
**  - - - - - - - - -
**
**  Transform geodetic coordinates to geocentric using the specified
**  reference ellipsoid.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards of Fundamental Astronomy) software collection.
**
**  Status:  canonical transformation.
**
**  Given:
**     n       int        ellipsoid identifier (Note 1)
**     elong   double     longitude (radians, east +ve)
**     phi     double     latitude (geodetic, radians, Note 3)
**     height  double     height above ellipsoid (geodetic, Notes 2,3)
**
**  Returned:
**     xyz     double[3]  geocentric vector (Note 2)
**
**  Returned (function value):
**            int       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
**
**     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 -1 means that the identifier n is
**     illegal.  An error status -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 function
**     iauGc2gde.
**
**  Called:
**     iauEform     Earth reference ellipsoids
**     iauGd2gce    geodetic to geocentric transformation, general
**
*/
```

```
int iauGd2gce ( double a, double f, double elong, double phi,
                double height, double xyz[3] )
/*
**  - - - - - - - - - -
**   i a u G d 2 g c e
**  - - - - - - - - - -
**
**  Transform geodetic coordinates to geocentric for a reference
**  ellipsoid of specified form.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     a       double     equatorial radius (Notes 1,4)
**     f       double     flattening (Notes 2,4)
**     elong   double     longitude (radians, east +ve)
**     phi     double     latitude (geodetic, radians, Note 4)
**     height  double     height above ellipsoid (geodetic, Notes 3,4)
**
**  Returned:
**     xyz     double[3]  geocentric vector (Note 3)
**
**  Returned (function value):
**            int         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 -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 function
**     iauGc2gde.
**
**  6) The transformation for a standard ellipsoid (such as WGS84) can
**     more conveniently be performed by calling iauGd2gc,  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 iauGmst00(double uta, double utb, double tta, double ttb)
/*
**  - - - - - - - - - -
**   i a u G m s t 0 0
**  - - - - - - - - - -
**
**  Greenwich mean sidereal time (model consistent with IAU 2000
**  resolutions).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**     tta,ttb    double    TT as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauEra00     Earth rotation angle, IAU 2000
**     iauAnp       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 iauGmst06(double uta, double utb, double tta, double ttb)
/*
**  - - - - - - - - - -
**   i a u G m s t 0 6
**  - - - - - - - - - -
**
**  Greenwich mean sidereal time (consistent with IAU 2006 precession).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**     tta,ttb    double    TT as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauEra00     Earth rotation angle, IAU 2000
**     iauAnp       normalize angle into range 0 to 2pi
**
**  Reference:
**
**     Capitaine, N., Wallace, P.T. & Chapront, J., 2005,
**     Astron.Astrophys. 432, 355
**
*/
```

```
double iauGmst82(double dj1, double dj2)
/*
**  - - - - - - - - - -
**   i a u G m s t 8 2
**  - - - - - - - - - -
**
**  Universal Time to Greenwich mean sidereal time (IAU 1982 model).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     dj1,dj2    double    UT1 Julian Date (see note)
**
**  Returned (function value):
**                double    Greenwich mean sidereal time (radians)
**
**  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 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 function, 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.
**
**  Called:
**     iauAnp       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 iauGst00a(double uta, double utb, double tta, double ttb)
/*
**  - - - - - - - - - -
**   i a u G s t 0 0 a
**  - - - - - - - - - -
**
**  Greenwich apparent sidereal time (consistent with IAU 2000
**  resolutions).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**     tta,ttb    double    TT as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauGmst00    Greenwich mean sidereal time, IAU 2000
**     iauEe00a     equation of the equinoxes, IAU 2000A
**     iauAnp       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 iauGst00b(double uta, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauGmst00    Greenwich mean sidereal time, IAU 2000
**     iauEe00b     equation of the equinoxes, IAU 2000B
**     iauAnp       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 iauGst06(double uta, double utb, double tta, double ttb,
                double rnpb[3][3])
/*
**  - - - - - - - - -
**   i a u G s t 0 6
**  - - - - - - - - -
**
**  Greenwich apparent sidereal time, IAU 2006, given the NPB matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     uta,utb  double        UT1 as a 2-part Julian Date (Notes 1,2)
**     tta,ttb  double        TT as a 2-part Julian Date (Notes 1,2)
**     rnpb     double[3][3]  nutation x precession x bias matrix
**
**  Returned (function value):
**              double        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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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 function 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:
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS06       the CIO locator s, given X,Y, IAU 2006
**     iauAnp       normalize angle into range 0 to 2pi
**     iauEra00     Earth rotation angle, IAU 2000
**     iauEors      equation of the origins, given NPB matrix and s
**
**  Reference:
**
**     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
**
*/
```

```
double iauGst06a(double uta, double utb, double tta, double ttb)
/*
**  - - - - - - - - - -
**   i a u G s t 0 6 a
**  - - - - - - - - - -
**
**  Greenwich apparent sidereal time (consistent with IAU 2000 and 2006
**  resolutions).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**     tta,ttb    double    TT as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauPnm06a    classical NPB matrix, IAU 2006/2000A
**     iauGst06     Greenwich apparent ST, IAU 2006, given NPB matrix
**
**  Reference:
**
**     Wallace, P.T. & Capitaine, N., 2006, Astron.Astrophys. 459, 981
**
*/
```

```
double iauGst94(double uta, double utb)
/*
**  - - - - - - - - -
**   i a u G s t 9 4
**  - - - - - - - - -
**
**  Greenwich apparent sidereal time (consistent with IAU 1982/94
**  resolutions).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     uta,utb    double    UT1 as a 2-part Julian Date (Notes 1,2)
**
**  Returned (function value):
**                double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function, called internally:  maximum precision 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:
**     iauGmst82    Greenwich mean sidereal time, IAU 1982
**     iauEqeq94    equation of the equinoxes, IAU 1994
**     iauAnp       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)
**
*/
```

```
void iauH2fk5(double rh, double dh,
              double drh, double ddh, double pxh, double rvh,
              double *r5, double *d5,
              double *dr5, double *dd5, double *px5, double *rv5)
/*
**  - - - - - - - - -
**   i a u H 2 f k 5
**  - - - - - - - - -
**
**  Transform Hipparcos star data into the FK5 (J2000.0) system.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given (all Hipparcos, epoch J2000.0):
**     rh      double    RA (radians)
**     dh      double    Dec (radians)
**     drh     double    proper motion in RA (dRA/dt, rad/Jyear)
**     ddh     double    proper motion in Dec (dDec/dt, rad/Jyear)
**     pxh     double    parallax (arcsec)
**     rvh     double    radial velocity (km/s, positive = receding)
**
**  Returned (all FK5, equinox J2000.0, epoch J2000.0):
**     r5      double    RA (radians)
**     d5      double    Dec (radians)
**     dr5     double    proper motion in RA (dRA/dt, rad/Jyear)
**     dd5     double    proper motion in Dec (dDec/dt, rad/Jyear)
**     px5     double    parallax (arcsec)
**     rv5     double    radial velocity (km/s, positive = receding)
**
**  Notes:
**
**  1) This function 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 iauFk52h, iauFk5hz, iauHfk5z.
**
**  Called:
**     iauStarpv    star catalog data to space motion pv-vector
**     iauFk5hip    FK5 to Hipparcos rotation and spin
**     iauRv2m      r-vector to r-matrix
**     iauRxp       product of r-matrix and p-vector
**     iauTrxp      product of transpose of r-matrix and p-vector
**     iauPxp       vector product of two p-vectors
**     iauPmp       p-vector minus p-vector
**     iauPvstar    space motion pv-vector to star catalog data
**
**  Reference:
**
**     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
**
*/
```

```
void iauHfk5z(double rh, double dh, double date1, double date2,
              double *r5, double *d5, double *dr5, double *dd5)
/*
**  - - - - - - - - -
**   i a u H f k 5 z
**  - - - - - - - - -
**
**  Transform a Hipparcos star position into FK5 J2000.0, assuming
**  zero Hipparcos proper motion.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     rh            double    Hipparcos RA (radians)
**     dh            double    Hipparcos Dec (radians)
**     date1,date2   double    TDB date (Note 1)
**
**  Returned (all FK5, equinox J2000.0, date date1+date2):
**     r5            double    RA (radians)
**     d5            double    Dec (radians)
**     dr5           double    FK5 RA proper motion (rad/year, Note 4)
**     dd5           double    Dec proper motion (rad/year, 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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function returns those fictitious
**     proper motions.
**
**  5) The position returned by this function is in the FK5 J2000.0
**     reference system but at date date1+date2.
**
**  6) See also iauFk52h, iauH2fk5, iauFk5zhz.
**
**  Called:
**     iauS2c       spherical coordinates to unit vector
**     iauFk5hip    FK5 to Hipparcos rotation and spin
**     iauRxp       product of r-matrix and p-vector
**     iauSxp       multiply p-vector by scalar
**     iauRxr       product of two r-matrices
**     iauTrxp      product of transpose of r-matrix and p-vector
```

```
**     iauPxp       vector product of two p-vectors
**     iauPv2s      pv-vector to spherical
**     iauAnp       normalize angle into range 0 to 2pi
**
**  Reference:
**
**     F.Mignard & M.Froeschle, 2000, Astron.Astrophys. 354, 732-739.
**
*/
```

```
void iauIr(double r[3][3])
/*
**  - - - - - -
**   i a u I r
**  - - - - - -
**
**  Initialize an r-matrix to the identity matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Returned:
**     r       double[3][3]    r-matrix
**
**  Called:
**     iauZr        zero r-matrix
**
*/
```

```
int iauJd2cal(double dj1, double dj2,
              int *iy, int *im, int *id, double *fd)
/*
**  - - - - - - - - - -
**   i a u J d 2 c a l
**  - - - - - - - - - -
**
**  Julian Date to Gregorian year, month, day, and fraction of a day.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     dj1,dj2   double   Julian Date (Notes 1, 2)
**
**  Returned (arguments):
**     iy        int      year
**     im        int      month
**     id        int      day
**     fd        double   fraction of day
**
**  Returned (function value):
**               int      status:
**                           0 = OK
**                          -1 = unacceptable date (Note 3)
**
**  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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).
**
*/
```

```
int iauJdcalf(int ndp, double dj1, double dj2, int iymdf[4])
/*
**  - - - - - - - - - - -
**   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.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     ndp       int      number of decimal places of days in fraction
**     dj1,dj2   double   dj1+dj2 = Julian Date (Note 1)
**
**  Returned:
**     iymdf     int[4]   year, month, day, fraction in Gregorian
**                        calendar
**
**  Returned (function value):
**               int      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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function iauJd2cal.
**
**  4) NDP should be 4 or less if internal overflows are to be
**     avoided on machines which use 16-bit integers.
**
**  Called:
**     iauJd2cal    JD to Gregorian calendar
**
**  Reference:
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     Section 12.92 (p604).
**
*/
```

```
void iauNum00a(double date1, double date2, double rmatn[3][3])
/*
**  - - - - - - - - - - -
**   i a u N u m 0 0 a
**  - - - - - - - - - - -
**
**  Form the matrix of nutation for a given date, IAU 2000A model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rmatn        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauNum00b function.
**
**  Called:
**     iauPn00a     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).
**
*/
```

```
void iauNum00b(double date1, double date2, double rmatn[3][3])
/*
**  - - - - - - - - - -
**   i a u N u m 0 0 b
**  - - - - - - - - - -
**
**  Form the matrix of nutation for a given date, IAU 2000B model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double         TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rmatn        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is faster, but slightly less accurate (about
**     1 mas), than the iauNum00a function.
**
**  Called:
**     iauPn00b     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).
**
*/
```

```
void iauNum06a(double date1, double date2, double rmatn[3][3])
/*
**  - - - - - - - - - - -
**   i a u N u m 0 6 a
**  - - - - - - - - - - -
**
**  Form the matrix of nutation for a given date, IAU 2006/2000A model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2   double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rmatn         double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauObl06     mean obliquity, IAU 2006
**     iauNut06a    nutation, IAU 2006/2000A
**     iauNumat     form nutation matrix
**
**  Reference:
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     Section 3.222-3 (p114).
**
*/
```

```
void iauNumat(double epsa, double dpsi, double deps, double rmatn[3][3])
/*
**  - - - - - - - - - 
**   i a u N u m a t
**  - - - - - - - - - 
**
**  Form the matrix of nutation.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     epsa        double         mean obliquity of date (Note 1)
**     dpsi,deps   double         nutation (Note 2)
**
**  Returned:
**     rmatn       double[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:
**     iauIr        initialize r-matrix to identity
**     iauRx        rotate around X-axis
**     iauRz        rotate around Z-axis
**
**  Reference:
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     Section 3.222-3 (p114).
**
*/
```

```
void iauNut00a(double date1, double date2, double *dpsi, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps     double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function 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 function and, if required, must be
**     independently computed.  With the FCN corrections included, the
**     present function 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 function 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 functions iauBi00  and
**     iauPr00.
**
**  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 iauPmat76,  to deliver GCRS-
**     to-true predictions of sub-mas accuracy at current dates.
**     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 functions do not generate the "total
**     nutations" directly, though they can of course easily be
**     generated by calling iauBi00, iauPr00 and the present function
**     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:
**     iauFal03     mean anomaly of the Moon
**     iauFaf03     mean argument of the latitude of the Moon
**     iauFaom03    mean longitude of the Moon's ascending node
**     iauFame03    mean longitude of Mercury
**     iauFave03    mean longitude of Venus
**     iauFae03     mean longitude of Earth
**     iauFama03    mean longitude of Mars
**     iauFaju03    mean longitude of Jupiter
**     iauFasa03    mean longitude of Saturn
**     iauFaur03    mean longitude of Uranus
**     iauFapa03    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)
**
*/
```

```
void iauNut00b(double date1, double date2, double *dpsi, double *deps)
/*
**  - - - - - - - - - -
**   i a u N u t 0 0 b
**  - - - - - - - - - -
**
**  Nutation, IAU 2000B model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double    TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps     double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function 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 function is an implementation of the IAU 2000B abridged
**     nutation model formally adopted by the IAU General Assembly in
**     2000.  The function 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 function 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
**     function iauNut00a (q.v.), delivers considerably greater accuracy
**     at current dates;  however, to realize this improved accuracy,
**     corrections for the essentially unpredictable free-core-nutation
```

```
**     (FCN) must also be included.
**
**  5) The present function 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 functions
**     iauBi00  and iauPr00.
**
**  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 iauPmat76,  to deliver GCRS-
**     to-true predictions of mas accuracy at current epochs.  However,
**     for symmetry with the iauNut00a  function (q.v. for the reasons),
**     the SOFA functions do not generate the "total nutations"
**     directly.  Should they be required, they could of course easily
**     be generated by calling iauBi00, iauPr00 and the present function
**     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)
**
*/
```

```
void iauNut06a(double date1, double date2, double *dpsi, double *deps)
/*
**  - - - - - - - - - -
**   i a u N u t 0 6 a
**  - - - - - - - - - -
**
**  IAU 2000A nutation with adjustments to match the IAU 2006
**  precession.
**
**  Given:
**     date1,date2   double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps     double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function 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 flattening.
**
**  4) The present function 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:
**     iauNut00a    nutation, IAU 2000A
**
**  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)
**
*/
```

```
void iauNut80(double date1, double date2, double *dpsi, double *deps)
/*
**  - - - - - - - - - 
**   i a u N u t 8 0
**  - - - - - - - - - 
**
**  Nutation, IAU 1980 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double    TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi          double    nutation in longitude (radians)
**     deps          double    nutation in obliquity (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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauAnpm      normalize angle into range +/- pi
**
**  Reference:
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     Section 3.222 (p111).
**
*/
```

```
void iauNutm80(double date1, double date2, double rmatn[3][3])
/*
**  - - - - - - - - - - 
**   i a u N u t m 8 0
**  - - - - - - - - - - 
**
**  Form the matrix of nutation for a given date, IAU 1980 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2    double          TDB date (Note 1)
**
**  Returned:
**     rmatn          double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauNut80     nutation, IAU 1980
**     iauObl80     mean obliquity, IAU 1980
**     iauNumat     form nutation matrix
**
*/
```

```
double iauObl06(double date1, double date2)
/*
**  - - - - - - - - -
**   i a u O b l 0 6
**  - - - - - - - - -
**
**  Mean obliquity of the ecliptic, IAU 2006 precession model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double   obliquity of the ecliptic (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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauObl80(double date1, double date2)
/*
**  - - - - - - - - -
**   i a u O b l 8 0
**  - - - - - - - - -
**
**  Mean obliquity of the ecliptic, IAU 1980 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                   double    obliquity of the ecliptic (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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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).
**
*/
```

```
void iauP06e(double date1, double date2,
             double *eps0, double *psia, double *oma, double *bpa,
             double *bqa, double *pia, double *bpia,
             double *epsa, double *chia, double *za, double *zetaa,
             double *thetaa, double *pa,
             double *gam, double *phi, double *psi)
/*
**  - - - - - - - - -
**   i a u P 0 6 e
**  - - - - - - - - -
**
**  Precession angles, IAU 2006, equinox based.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical models.
**
**  Given:
**     date1,date2   double   TT as a 2-part Julian Date (Note 1)
**
**  Returned (see Note 2):
**     eps0          double   epsilon_0
**     psia          double   psi_A
**     oma           double   omega_A
**     bpa           double   P_A
**     bqa           double   Q_A
**     pia           double   pi_A
**     bpia          double   Pi_A
**     epsa          double   obliquity epsilon_A
**     chia          double   chi_A
**     za            double   z_A
**     zetaa         double   zeta_A
**     thetaa        double   theta_A
**     pa            double   p_A
**     gam           double   F-W angle gamma_J2000
**     phi           double   F-W angle phi_J2000
**     psi           double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function 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 functions:
**
**     . iauXy06  contains the polynomial parts of the X and Y series.
**
**     . iauS06  contains the polynomial part of the s+XY/2 series.
**
**     . iauPfw06  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 function.
**
**  5) The parameterization used by SOFA is the Fukushima-Williams angles
**     referred directly to the GCRS pole.  These are the final four
**     arguments returned by the present function, but are more
**     efficiently calculated by calling the function iauPfw06.   SOFA
**     also supports the direct computation of the CIP GCRS X,Y by
**     series, available by calling iauXy06.
**
**  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.
**
**  8) It is permissible to re-use the same variable in the returned
**     arguments.  The quantities are stored in the stated order.
**
**  Reference:
**
**     Hilton, J. et al., 2006, Celest.Mech.Dyn.Astron. 94, 351
**
**  Called:
**     iauObl06     mean obliquity, IAU 2006
**
*/
```

```
void iauP2pv(double p[3], double pv[2][3])
/*
**  - - - - - - - -
**   i a u P 2 p v
**  - - - - - - - -
**
**  Extend a p-vector to a pv-vector by appending a zero velocity.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p        double[3]       p-vector
**
**  Returned:
**     pv       double[2][3]    pv-vector
**
**  Called:
**     iauCp        copy p-vector
**     iauZp        zero p-vector
**
*/
```

```
void iauP2s(double p[3], double *theta, double *phi, double *r)
/*
**  - - - - - - -
**   i a u P 2 s
**  - - - - - - -
**
**  P-vector to spherical polar coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p        double[3]    p-vector
**
**  Returned:
**     theta    double       longitude angle (radians)
**     phi      double       latitude angle (radians)
**     r        double       radial distance
**
**  Notes:
**
**  1) If P is null, zero theta, phi and r are returned.
**
**  2) At either pole, zero theta is returned.
**
**  Called:
**     iauC2s       p-vector to spherical
**     iauPm        modulus of p-vector
**
*/
```

```
double iauPap(double a[3], double b[3])
/*
**  - - - - - - -
**   i a u P a p
**  - - - - - - -
**
**  Position-angle from two p-vectors.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a      double[3]  direction of reference point
**     b      double[3]  direction of point whose PA is required
**
**  Returned (function value):
**            double     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) The vectors a and b need not be of unit length.
**
**  3) Zero is returned if the two directions are the same or if either
**     vector is null.
**
**  4) If vector a is at a pole, the result is ill-defined.
**
**  Called:
**     iauPn        decompose p-vector into modulus and direction
**     iauPm        modulus of p-vector
**     iauPxp       vector product of two p-vectors
**     iauPmp       p-vector minus p-vector
**     iauPdp       scalar product of two p-vectors
**
*/
```

```
double iauPas(double al, double ap, double bl, double bp)
/*
**  - - - - - - -
**   i a u P a s
**  - - - - - - -
**
**  Position-angle from spherical coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     al     double     longitude of point A (e.g. RA) in radians
**     ap     double     latitude of point A (e.g. Dec) in radians
**     bl     double     longitude of point B
**     bp     double     latitude of point B
**
**  Returned (function value):
**            double     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.
**
*/
```

```
void iauPb06(double date1, double date2,
             double *bzeta, double *bz, double *btheta)
/*
**  - - - - - - - -
**   i a u P b 0 6
**  - - - - - - - -
**
**  This function 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     bzeta        double   1st rotation: radians cw around z
**     bz           double   3rd rotation: radians cw around z
**     btheta       double   2nd rotation: radians ccw around y
**
**  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function iauP06e.
**
**  Called:
**     iauPmat06    PB matrix, IAU 2006
**     iauRz        rotate around Z-axis
**
*/
```

```
double iauPdp(double a[3], double b[3])
/*
**  - - - - - - - -
**   i a u P d p
**  - - - - - - - -
**
**  p-vector inner (=scalar=dot) product.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a      double[3]     first p-vector
**     b      double[3]     second p-vector
**
**  Returned (function value):
**            double        a . b
**
*/
```

```
void iauPfw06(double date1, double date2,
              double *gamb, double *phib, double *psib, double *epsa)
/*
**  - - - - - - - - -
**   i a u P f w 0 6
**  - - - - - - - - -
**
**  Precession angles, IAU 2006 (Fukushima-Williams 4-angle formulation).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     gamb         double   F-W angle gamma_bar (radians)
**     phib         double   F-W angle phi_bar (radians)
**     psib         double   F-W angle psi_bar (radians)
**     epsa         double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauObl06     mean obliquity, IAU 2006
**
*/
```

```
int iauPlan94(double date1, double date2, int np, double pv[2][3])
/*
**  - - - - - - - - - -
**   i a u P l a n 9 4
**  - - - - - - - - - -
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  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  double       TDB date part A (Note 1)
**     date2  double       TDB date part B (Note 1)
**     np     int          planet (1=Mercury, 2=Venus, 3=EMB, 4=Mars,
**                             5=Jupiter, 6=Saturn, 7=Uranus, 8=Neptune)
**
**  Returned (argument):
**     pv     double[3][2] planet p,v (heliocentric, J2000.0, AU,AU/d)
**
**  Returned (function value):
**            int          status: -1 = illegal NP (outside 1-8)
**                                  0 = OK
**                                 +1 = warning: year outside 1000-3000
**                                 +2 = warning: failed to converge
**
**  Notes:
**
**  1) The date date1+date2 is in the TDB time scale (in practice TT can
**     be used) and 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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     (function value -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 function iauEpv00.
**
**  4) On successful return, the array pv contains the following:
**
**        pv[0][0]   x      }
**        pv[1][0]   y      } heliocentric position, AU
**        pv[2][0]   z      }
**
**        pv[0][1]   xdot   }
**        pv[1][1]   ydot   } heliocentric velocity, AU/d
**        pv[2][1]   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 function 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:
**
**       *  C instead of Fortran.
**
**       *  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 subroutines.
**
**       *  Different error/warning status values are used.
**
**       *  A different Kepler's-equation-solver is used (avoiding
**          use of double precision complex).
**
**       *  Polynomials in t are nested to minimize rounding errors.
```

```
**
**       *  Explicit double constants are used to avoid mixed-mode
**          expressions.
**
**     None of the above changes affects the result significantly.
**
**  7) The returned status indicates the most serious condition
**     encountered during execution of the function.  Illegal np is
**     considered the most serious, overriding failure to converge,
**     which in turn takes precedence over the remote date warning.
**
**  Called:
**     iauAnp       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).
**
*/
```

```
double iauPm(double p[3])
/*
**  - - - - - -
**   i a u P m
**  - - - - - -
**
**  Modulus of p-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p      double[3]     p-vector
**
**  Returned (function value):
**            double        modulus
**
*/
```

```
void iauPmat00(double date1, double date2, double rbp[3][3])
/*
**  - - - - - - - - - -
**   i a u P m a t 0 0
**  - - - - - - - - - -
**
**  Precession matrix (including frame bias) from GCRS to a specified
**  date, IAU 2000 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rbp          double[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 two arguments.  For example,
**     JD(TT)=2450123.7 could be expressed in any of these ways,
**     among others:
**
**            date1          date2
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauBp00      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)
**
*/
```

```
void iauPmat06(double date1, double date2, double rbp[3][3])
/*
**  - - - - - - - - - -
**   i a u P m a t 0 6
**  - - - - - - - - - -
**
**  Precession matrix (including frame bias) from GCRS to a specified
**  date, IAU 2006 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rbp          double[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 two arguments.  For example,
**     JD(TT)=2450123.7 could be expressed in any of these ways,
**     among others:
**
**            date1          date2
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPfw06     bias-precession F-W angles, IAU 2006
**     iauFw2m      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
**
*/
```

```
void iauPmat76(double date1, double date2, double rmatp[3][3])
/*
**  - - - - - - - - - -
**   i a u P m a t 7 6
**  - - - - - - - - - -
**
**  Precession matrix from J2000.0 to a specified date, IAU 1976 model.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       ending date, TT (Note 1)
**
**  Returned:
**     rmatp       double[3][3] precession matrix, J2000.0 -> date1+date2
**
**  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPrec76    accumulated precession angles, IAU 1976
**     iauIr        initialize r-matrix to identity
**     iauRz        rotate around Z-axis
**     iauRy        rotate around Y-axis
**     iauCr        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.
**
```

```
*/
```

```
void iauPmp(double a[3], double b[3], double amb[3])
/*
**  - - - - - - -
**   i a u P m p
**  - - - - - - -
**
**  P-vector subtraction.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[3]      first p-vector
**     b        double[3]      second p-vector
**
**  Returned:
**     amb      double[3]      a - b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
*/
```

```
void iauPn(double p[3], double *r, double u[3])
/*
**  - - - - - -
**   i a u P n
**  - - - - - -
**
**  Convert a p-vector into modulus and unit vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     p        double[3]      p-vector
**
**  Returned:
**     r        double         modulus
**     u        double[3]      unit vector
**
**  Notes:
**
**  1) If p is null, the result is null.  Otherwise the result is a unit
**     vector.
**
**  2) It is permissible to re-use the same array for any of the
**     arguments.
**
**  Called:
**     iauPm        modulus of p-vector
**     iauZp        zero p-vector
**     iauSxp       multiply p-vector by scalar
**
*/
```

```
void iauPn00(double date1, double date2, double dpsi, double deps,
             double *epsa,
             double rb[3][3], double rp[3][3], double rbp[3][3],
             double rn[3][3], double rbpn[3][3])
/*
**  - - - - - - - -
**   i a u P n 0 0
**  - - - - - - - -
**
**  Precession-nutation, IAU 2000 model:  a multi-purpose function,
**  supporting classical (equinox-based) use directly and CIO-based
**  use indirectly.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**     dpsi,deps    double          nutation (Note 2)
**
**  Returned:
**     epsa         double          mean obliquity (Note 3)
**     rb           double[3][3]    frame bias matrix (Note 4)
**     rp           double[3][3]    precession matrix (Note 5)
**     rbp          double[3][3]    bias-precession matrix (Note 6)
**     rn           double[3][3]    nutation matrix (Note 7)
**     rbpn         double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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.
**
**  9) It is permissible to re-use the same array in the returned
**     arguments.  The arrays are filled in the order given.
**
**  Called:
**     iauPr00      IAU 2000 precession adjustments
**     iauObl80     mean obliquity, IAU 1980
**     iauBp00      frame bias and precession matrices, IAU 2000
**     iauCr        copy r-matrix
**     iauNumat     form nutation matrix
**     iauRxr       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", Astronomy & Astrophysics, 400, 1145-1154 (2003)
**
**     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
**          intermediate origin" (CIO) by IAU 2006 Resolution 2.
**
*/
```

```
void iauPn00a(double date1, double date2,
              double *dpsi, double *deps, double *epsa,
              double rb[3][3], double rp[3][3], double rbp[3][3],
              double rn[3][3], double rbpn[3][3])
/*
**  - - - - - - - - -
**   i a u P n 0 0 a
**  - - - - - - - - -
**
**  Precession-nutation, IAU 2000A model:  a multi-purpose function,
**  supporting classical (equinox-based) use directly and CIO-based
**  use indirectly.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps    double          nutation (Note 2)
**     epsa         double          mean obliquity (Note 3)
**     rb           double[3][3]    frame bias matrix (Note 4)
**     rp           double[3][3]    precession matrix (Note 5)
**     rbp          double[3][3]    bias-precession matrix (Note 6)
**     rn           double[3][3]    nutation matrix (Note 7)
**     rbpn         double[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
**
**          2450123.7           0.0       (JD method)
**          2451545.0       -1421.3       (J2000 method)
**          2400000.5       50123.2       (MJD method)
**          2450123.5           0.2       (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 iauPn00  function, where the
**      nutation components are caller-specified.  For faster but
**      slightly less accurate results, use the iauPn00b function.
**
**  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.
**
**  10) It is permissible to re-use the same array in the returned
**      arguments.  The arrays are filled in the order given.
**
**  Called:
**     iauNut00a    nutation, IAU 2000A
**     iauPn00      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", Astronomy & Astrophysics, 400, 1145-1154 (2003)
**
**     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
**          intermediate origin" (CIO) by IAU 2006 Resolution 2.
**
*/
```

```
void iauPn00b(double date1, double date2,
              double *dpsi, double *deps, double *epsa,
              double rb[3][3], double rp[3][3], double rbp[3][3],
              double rn[3][3], double rbpn[3][3])
/*
**  - - - - - - - - -
**   i a u P n 0 0 b
**  - - - - - - - - -
**
**  Precession-nutation, IAU 2000B model:  a multi-purpose function,
**  supporting classical (equinox-based) use directly and CIO-based
**  use indirectly.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps    double          nutation (Note 2)
**     epsa         double          mean obliquity (Note 3)
**     rb           double[3][3]    frame bias matrix (Note 4)
**     rp           double[3][3]    precession matrix (Note 5)
**     rbp          double[3][3]    bias-precession matrix (Note 6)
**     rn           double[3][3]    nutation matrix (Note 7)
**     rbpn         double[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
**
**          2450123.7           0.0       (JD method)
**          2451545.0       -1421.3       (J2000 method)
**          2400000.5       50123.2       (MJD method)
**          2450123.5           0.2       (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 iauPn00a function.
**      For the utmost accuracy, use the iauPn00  function, 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.
**
**  10) It is permissible to re-use the same array in the returned
**      arguments.  The arrays are filled in the stated order.
**
**  Called:
**     iauNut00b    nutation, IAU 2000B
**     iauPn00      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", Astronomy & Astrophysics, 400, 1145-1154 (2003).
**
**     n.b. The celestial ephemeris origin (CEO) was renamed "celestial
**          intermediate origin" (CIO) by IAU 2006 Resolution 2.
**
*/
```

```
void iauPn06(double date1, double date2, double dpsi, double deps,
             double *epsa,
             double rb[3][3], double rp[3][3], double rbp[3][3],
             double rn[3][3], double rbpn[3][3])
/*
**  - - - - - - - -
**   i a u P n 0 6
**  - - - - - - - -
**
**  Precession-nutation, IAU 2006 model:  a multi-purpose function,
**  supporting classical (equinox-based) use directly and CIO-based use
**  indirectly.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**     dpsi,deps    double          nutation (Note 2)
**
**  Returned:
**     epsa         double          mean obliquity (Note 3)
**     rb           double[3][3]    frame bias matrix (Note 4)
**     rp           double[3][3]    precession matrix (Note 5)
**     rbp          double[3][3]    bias-precession matrix (Note 6)
**     rn           double[3][3]    nutation matrix (Note 7)
**     rbpn         double[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
**
**          2450123.7           0.0       (JD method)
**          2451545.0       -1421.3       (J2000 method)
**          2400000.5       50123.2       (MJD method)
**          2450123.5           0.2       (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 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.
**
**  10) It is permissible to re-use the same array in the returned
**      arguments.  The arrays are filled in the stated order.
**
**  Called:
**     iauPfw06     bias-precession F-W angles, IAU 2006
**     iauFw2m      F-W angles to r-matrix
**     iauCr        copy r-matrix
**     iauTr        transpose r-matrix
**     iauRxr       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
**
*/
```

```
void iauPn06a(double date1, double date2,
              double *dpsi, double *deps, double *epsa,
              double rb[3][3], double rp[3][3], double rbp[3][3],
              double rn[3][3], double rbpn[3][3])
/*
**  - - - - - - - - -
**   i a u P n 0 6 a
**  - - - - - - - - -
**
**  Precession-nutation, IAU 2006/2000A models:  a multi-purpose function,
**  supporting classical (equinox-based) use directly and CIO-based use
**  indirectly.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double          TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsi,deps    double          nutation (Note 2)
**     epsa         double          mean obliquity (Note 3)
**     rb           double[3][3]    frame bias matrix (Note 4)
**     rp           double[3][3]    precession matrix (Note 5)
**     rbp          double[3][3]    bias-precession matrix (Note 6)
**     rn           double[3][3]    nutation matrix (Note 7)
**     rbpn         double[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
**
**          2450123.7           0.0       (JD method)
**          2451545.0       -1421.3       (J2000 method)
**          2400000.5       50123.2       (MJD method)
**          2450123.5           0.2       (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 iauPn06 function, 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 (1,1-3) of the matrix rbpn.
**
**  10) It is permissible to re-use the same array in the returned
**      arguments.  The arrays are filled in the stated order.
**
**  Called:
**     iauNut06a    nutation, IAU 2006/2000A
**     iauPn06      bias/precession/nutation results, IAU 2006
**
**  Reference:
**
**     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855
**
*/
```

```
void iauPnm00a(double date1, double date2, double rbpn[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double     TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rbpn         double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauPnm00b function.
**
**  Called:
**     iauPn00a     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)
**
*/
```

```
void iauPnm00b(double date1, double date2, double rbpn[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rbpn        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is faster, but slightly less accurate (about
**     1 mas), than the iauPnm00a function.
**
**  Called:
**     iauPn00b     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)
**
*/
```

```
void iauPnm06a(double date1, double date2, double rnpb[3][3])
/*
**  - - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2 double       TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     rnpb        double[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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPfw06     bias-precession F-W angles, IAU 2006
**     iauNut06a    nutation, IAU 2006/2000A
**     iauFw2m      F-W angles to r-matrix
**
**  Reference:
**
**     Capitaine, N. & Wallace, P.T., 2006, Astron.Astrophys. 450, 855.
**
*/
```

```
void iauPnm80(double date1, double date2, double rmatpn[3][3])
/*
**  - - - - - - - - -
**   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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2    double         TDB date (Note 1)
**
**  Returned:
**     rmatpn         double[3][3]   combined precession/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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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:
**     iauPmat76    precession matrix, IAU 1976
**     iauNutm80    nutation matrix, IAU 1980
**     iauRxr       product of two r-matrices
**
**  Reference:
**
**     Explanatory Supplement to the Astronomical Almanac,
**     P. Kenneth Seidelmann (ed), University Science Books (1992),
**     Section 3.3 (p145).
**
*/
```

```
void iauPom00(double xp, double yp, double sp, double rpom[3][3])
/*
**  - - - - - - - - - -
**   i a u P o m 0 0
**  - - - - - - - - - -
**
**  Form the matrix of polar motion for a given date, IAU 2000.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     xp,yp    double    coordinates of the pole (radians, Note 1)
**     sp       double    the TIO locator s' (radians, Note 2)
**
**  Returned:
**     rpom     double[3][3]   polar-motion matrix (Note 3)
**
**  Notes:
**
**  1) The arguments 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) The argument 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 function iauSp00
**     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:
**     iauIr        initialize r-matrix to identity
**     iauRz        rotate around Z-axis
**     iauRy        rotate around Y-axis
**     iauRx        rotate around X-axis
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
void iauPpp(double a[3], double b[3], double apb[3])
/*
**  - - - - - - -
**   i a u P p p
**  - - - - - - -
**
**  P-vector addition.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[3]      first p-vector
**     b        double[3]      second p-vector
**
**  Returned:
**     apb      double[3]      a + b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
*/
```

```
void iauPpsp(double a[3], double s, double b[3], double apsb[3])
/*
**  - - - - - - - -
**   i a u P p s p
**  - - - - - - - -
**
**  P-vector plus scaled p-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a      double[3]     first p-vector
**     s      double        scalar (multiplier for b)
**     b      double[3]     second p-vector
**
**  Returned:
**     apsb   double[3]     a + s*b
**
**  Note:
**     It is permissible for any of a, b and apsb to be the same array.
**
**  Called:
**     iauSxp       multiply p-vector by scalar
**     iauPpp       p-vector plus p-vector
**
*/
```

```
void iauPr00(double date1, double date2, double *dpsipr, double *depspr)
/*
**  - - - - - - - -
**   i a u P r 0 0
**  - - - - - - - -
**
**  Precession-rate part of the IAU 2000 precession-nutation models
**  (part of MHB2000).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2    double  TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     dpsipr,depspr  double  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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).
**
```

```
*/
```

```
void iauPrec76(double ep01, double ep02, double ep11, double ep12,
               double *zeta, double *z, double *theta)
/*
**  - - - - - - - - - -
**   i a u P r e c 7 6
**  - - - - - - - - - -
**
**  IAU 1976 precession model.
**
**  This function forms the three Euler angles which implement general
**  precession between two epochs, using the IAU 1976 model (as for
**  the FK5 catalog).
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     ep01,ep02   double    TDB starting epoch (Note 1)
**     ep11,ep12   double    TDB ending epoch (Note 1)
**
**  Returned:
**     zeta        double    1st rotation: radians cw around z
**     z           double    3rd rotation: radians cw around z
**     theta       double    2nd rotation: radians ccw around y
**
**  Notes:
**
**  1) The epochs ep01+ep02 and ep11+ep12 are Julian Dates, apportioned
**     in any convenient way between the arguments epn1 and epn2.  For
**     example, JD(TDB)=2450123.7 could be expressed in any of these
**     ways, among others:
**
**             epn1          epn2
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 epochs 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.
**
```

```
*/
```

```
void iauPv2p(double pv[2][3], double p[3])
/*
**  - - - - - - - -
**   i a u P v 2 p
**  - - - - - - - -
**
**  Discard velocity component of a pv-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     pv      double[2][3]     pv-vector
**
**  Returned:
**     p       double[3]        p-vector
**
**  Called:
**     iauCp        copy p-vector
**
*/
```

```
void iauPv2s(double pv[2][3],
             double *theta, double *phi, double *r,
             double *td, double *pd, double *rd)
/*
**  - - - - - - - -
**   i a u P v 2 s
**  - - - - - - - -
**
**  Convert position/velocity from Cartesian to spherical coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     pv       double[2][3]  pv-vector
**
**  Returned:
**     theta    double        longitude angle (radians)
**     phi      double        latitude angle (radians)
**     r        double        radial distance
**     td       double        rate of change of theta
**     pd       double        rate of change of phi
**     rd       double        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.
**
*/
```

```
void iauPvdpv(double a[2][3], double b[2][3], double adb[2])
/*
**  - - - - - - - - -
**   i a u P v d p v
**  - - - - - - - - -
**
**  Inner (=scalar=dot) product of two pv-vectors.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[2][3]      first pv-vector
**     b        double[2][3]      second pv-vector
**
**  Returned:
**     adb      double[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:
**     iauPdp       scalar product of two p-vectors
**
*/
```

```
void iauPvm(double pv[2][3], double *r, double *s)
/*
**  - - - - - - -
**   i a u P v m
**  - - - - - - -
**
**  Modulus of pv-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     pv     double[2][3]   pv-vector
**
**  Returned:
**     r      double         modulus of position component
**     s      double         modulus of velocity component
**
**  Called:
**     iauPm        modulus of p-vector
**
*/
```

```
void iauPvmpv(double a[2][3], double b[2][3], double amb[2][3])
/*
**  - - - - - - - - -
**   i a u P v m p v
**  - - - - - - - - -
**
**  Subtract one pv-vector from another.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a       double[2][3]      first pv-vector
**     b       double[2][3]      second pv-vector
**
**  Returned:
**     amb     double[2][3]      a - b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
**  Called:
**     iauPmp       p-vector minus p-vector
**
*/
```

```
void iauPvppv(double a[2][3], double b[2][3], double apb[2][3])
/*
**  - - - - - - - - - -
**   i a u P v p p v
**  - - - - - - - - - -
**
**  Add one pv-vector to another.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[2][3]      first pv-vector
**     b        double[2][3]      second pv-vector
**
**  Returned:
**     apb      double[2][3]      a + b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
**  Called:
**     iauPpp       p-vector plus p-vector
**
*/
```

```
int iauPvstar(double pv[2][3], double *ra, double *dec,
              double *pmr, double *pmd, double *px, double *rv)
/*
**  - - - - - - - - - - -
**   i a u P v s t a r
**  - - - - - - - - - - -
**
**  Convert star position+velocity vector to catalog coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given (Note 1):
**     pv     double[2][3]   pv-vector (AU, AU/day)
**
**  Returned (Note 2):
**     ra     double         right ascension (radians)
**     dec    double         declination (radians)
**     pmr    double         RA proper motion (radians/year)
**     pmd    double         Dec proper motion (radians/year)
**     px     double         parallax (arcsec)
**     rv     double         radial velocity (km/s, positive = receding)
**
**  Returned (function value):
**            int            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 date at which the light leaving the star
**     reached the solar-system barycenter.
**
**  2) The star data returned by this function 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
**     function iauStarpv), 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 function aborts with an error status.
**
**  6) The inverse transformation is performed by the function iauStarpv.
**
**  Called:
**     iauPn        decompose p-vector into modulus and direction
**     iauPdp       scalar product of two p-vectors
**     iauSxp       multiply p-vector by scalar
**     iauPmp       p-vector minus p-vector
**     iauPm        modulus of p-vector
**     iauPpp       p-vector plus p-vector
**     iauPv2s      pv-vector to spherical
**     iauAnp       normalize angle into range 0 to 2pi
**
**  Reference:
**
**     Stumpff, P., 1985, Astron.Astrophys. 144, 232-240.
**
*/
```

```
void iauPvu(double dt, double pv[2][3], double upv[2][3])
/*
**  - - - - - - -
**   i a u P v u
**  - - - - - - -
**
**  Update a pv-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     dt       double           time interval
**     pv       double[2][3]     pv-vector
**
**  Returned:
**     upv      double[2][3]     p updated, v unchanged
**
**  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.
**
**  3) It is permissible for pv and upv to be the same array.
**
**  Called:
**     iauPpsp      p-vector plus scaled p-vector
**     iauCp        copy p-vector
**
*/
```

```
void iauPvup(double dt, double pv[2][3], double p[3])
/*
**  - - - - - - - -
**   i a u P v u p
**  - - - - - - - -
**
**  Update a pv-vector, discarding the velocity component.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     dt       double            time interval
**     pv       double[2][3]      pv-vector
**
**  Returned:
**     p        double[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.
**
*/
```

```
void iauPvxpv(double a[2][3], double b[2][3], double axb[2][3])
/*
**  - - - - - - - - -
**   i a u P v x p v
**  - - - - - - - - -
**
**  Outer (=vector=cross) product of two pv-vectors.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[2][3]      first pv-vector
**     b        double[2][3]      second pv-vector
**
**  Returned:
**     axb      double[2][3]      a x b
**
**  Notes:
**
**  1) 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.
**
**  2) It is permissible to re-use the same array for any of the
**     arguments.
**
**  Called:
**     iauCpv       copy pv-vector
**     iauPxp       vector product of two p-vectors
**     iauPpp       p-vector plus p-vector
**
*/
```

```
void iauPxp(double a[3], double b[3], double axb[3])
/*
**  - - - - - - -
**   i a u P x p
**  - - - - - - -
**
**  p-vector outer (=vector=cross) product.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[3]      first p-vector
**     b        double[3]      second p-vector
**
**  Returned:
**     axb      double[3]      a x b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
*/
```

```
void iauRm2v(double r[3][3], double w[3])
/*
**  - - - - - - - -
**   i a u R m 2 v
**  - - - - - - - -
**
**  Express an r-matrix as an r-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    rotation matrix
**
**  Returned:
**     w        double[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 function 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 function iauPn.)
**
**  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.
**
*/
```

```
void iauRv2m(double w[3], double r[3][3])
/*
**  - - - - - - - -
**   i a u R v 2 m
**  - - - - - - - -
**
**  Form the r-matrix corresponding to a given r-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     w        double[3]      rotation vector (Note 1)
**
**  Returned:
**     r        double[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 function 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.
**
*/
```

```
void iauRx(double phi, double r[3][3])
/*
**  - - - - - -
**   i a u R x
**  - - - - - -
**
**  Rotate an r-matrix about the x-axis.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     phi    double          angle (radians)
**
**  Given and returned:
**     r      double[3][3]    r-matrix
**
**  Sign convention:  The matrix can be used to rotate the reference
**  frame of a vector.  Calling this function with positive phi
**  incorporates in the matrix an additional rotation, about the x-axis,
**  anticlockwise as seen looking towards the origin from positive x.
**
**  Called:
**     iauIr        initialize r-matrix to identity
**     iauRxr       product of two r-matrices
**     iauCr        copy r-matrix
**
*/
```

```
void iauRxp(double r[3][3], double p[3], double rp[3])
/*
**  - - - - - - - -
**   i a u R x p
**  - - - - - - - -
**
**  Multiply a p-vector by an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    r-matrix
**     p        double[3]       p-vector
**
**  Returned:
**     rp       double[3]       r * p
**
**  Note:
**     It is permissible for p and rp to be the same array.
**
**  Called:
**     iauCp        copy p-vector
**
*/
```

```
void iauRxpv(double r[3][3], double pv[2][3], double rpv[2][3])
/*
**  - - - - - - - -
**   i a u R x p v
**  - - - - - - - -
**
**  Multiply a pv-vector by an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    r-matrix
**     pv       double[2][3]    pv-vector
**
**  Returned:
**     rpv      double[2][3]    r * pv
**
**  Note:
**     It is permissible for pv and rpv to be the same array.
**
**  Called:
**     iauRxp       product of r-matrix and p-vector
**
*/
```

```
void iauRxr(double a[3][3], double b[3][3], double atb[3][3])
/*
**  - - - - - - -
**   i a u R x r
**  - - - - - - -
**
**  Multiply two r-matrices.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a        double[3][3]    first r-matrix
**     b        double[3][3]    second r-matrix
**
**  Returned:
**     atb      double[3][3]    a * b
**
**  Note:
**     It is permissible to re-use the same array for any of the
**     arguments.
**
**  Called:
**     iauCr        copy r-matrix
**
*/
```

```
void iauRy(double theta, double r[3][3])
/*
**  - - - - - -
**   i a u R y
**  - - - - - -
**
**  Rotate an r-matrix about the y-axis.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     theta  double          angle (radians)
**
**  Given and returned:
**     r      double[3][3]    r-matrix
**
**  Sign convention:  The matrix can be used to rotate the reference
**  frame of a vector.  Calling This function with positive theta
**  incorporates in the matrix an additional rotation, about the y-axis,
**  anticlockwise as seen looking towards the origin from positive y.
**
**  Called:
**     iauIr        initialize r-matrix to identity
**     iauRxr       product of two r-matrices
**     iauCr        copy r-matrix
**
*/
```

```
void iauRz(double psi, double r[3][3])
/*
**  - - - - - -
**   i a u R z
**  - - - - - -
**
**  Rotate an r-matrix about the z-axis.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     psi    double          angle (radians)
**
**  Given and returned:
**     r      double[3][3]    r-matrix, rotated
**
**  Sign convention:  The matrix can be used to rotate the reference
**  frame of a vector.  Calling This function with positive psi
**  incorporates in the matrix an additional rotation, about the z-axis,
**  anticlockwise as seen looking towards the origin from positive z.
**
**  Called:
**     iauIr        initialize r-matrix to identity
**     iauRxr       product of two r-matrices
**     iauCr        copy r-matrix
**
*/
```

```
double iauS00(double date1, double date2, double x, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double    TT as a 2-part Julian Date (Note 1)
**     x,y           double    CIP coordinates (Note 3)
**
**  Returned (function value):
**                   double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function 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:
**     iauFal03     mean anomaly of the Moon
**     iauFalp03    mean anomaly of the Sun
**     iauFaf03     mean argument of the latitude of the Moon
**     iauFad03     mean elongation of the Moon from the Sun
**     iauFaom03    mean longitude of the Moon's ascending node
**     iauFave03    mean longitude of Venus
**     iauFae03     mean longitude of Earth
**     iauFapa03    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", Astronomy & Astrophysics, 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 iauS00a(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function 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 function iauS00b, which uses
**     instead the IAU 2000B truncated model.
**
**  Called:
**     iauPnm00a    classical NPB matrix, IAU 2000A
**     iauBnp2xy    extract CIP X,Y from the BPN matrix
**     iauS00       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", Astronomy & Astrophysics, 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 iauS00b(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function uses the IAU 2000B truncated nutation model when
**     predicting the CIP position.  The function iauS00a uses instead
**     the full IAU 2000A model, but with no significant increase in
**     accuracy and at some cost in speed.
**
**  Called:
**     iauPnm00b     classical NPB matrix, IAU 2000B
**     iauBnp2xy     extract CIP X,Y from the BPN matrix
**     iauS00        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", Astronomy & Astrophysics, 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 iauS06(double date1, double date2, double x, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2   double    TT as a 2-part Julian Date (Note 1)
**     x,y           double    CIP coordinates (Note 3)
**
**  Returned (function value):
**                   double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function 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:
**     iauFal03     mean anomaly of the Moon
**     iauFalp03    mean anomaly of the Sun
**     iauFaf03     mean argument of the latitude of the Moon
**     iauFad03     mean elongation of the Moon from the Sun
**     iauFaom03    mean longitude of the Moon's ascending node
**     iauFave03    mean longitude of Venus
**     iauFae03     mean longitude of Earth
**     iauFapa03    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 iauS06a(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function uses the full IAU 2000A nutation model when predicting
**     the CIP position.
**
**  Called:
**     iauPnm06a    classical NPB matrix, IAU 2006/2000A
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS06       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", Astronomy & Astrophysics, 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
**
*/
```

```
void iauS2c(double theta, double phi, double c[3])
/*
**  - - - - - - -
**   i a u S 2 c
**  - - - - - - -
**
**  Convert spherical coordinates to Cartesian.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     theta    double       longitude angle (radians)
**     phi      double       latitude angle (radians)
**
**  Returned:
**     c        double[3]    direction cosines
**
*/
```

```
void iauS2p(double theta, double phi, double r, double p[3])
/*
**  - - - - - - -
**   i a u S 2 p
**  - - - - - - -
**
**  Convert spherical polar coordinates to p-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     theta   double       longitude angle (radians)
**     phi     double       latitude angle (radians)
**     r       double       radial distance
**
**  Returned:
**     p       double[3]    Cartesian coordinates
**
**  Called:
**     iauS2c       spherical coordinates to unit vector
**     iauSxp       multiply p-vector by scalar
**
*/
```

```
void iauS2pv(double theta, double phi, double r,
             double td, double pd, double rd,
             double pv[2][3])
/*
**  - - - - - - - -
**   i a u S 2 p v
**  - - - - - - - -
**
**  Convert position/velocity from spherical to Cartesian coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     theta    double          longitude angle (radians)
**     phi      double          latitude angle (radians)
**     r        double          radial distance
**     td       double          rate of change of theta
**     pd       double          rate of change of phi
**     rd       double          rate of change of r
**
**  Returned:
**     pv       double[2][3]    pv-vector
**
*/
```

```
void iauS2xpv(double s1, double s2, double pv[2][3], double spv[2][3])
/*
**  - - - - - - - - -
**   i a u S 2 x p v
**  - - - - - - - - -
**
**  Multiply a pv-vector by two scalars.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     s1     double         scalar to multiply position component by
**     s2     double         scalar to multiply velocity component by
**     pv     double[2][3]   pv-vector
**
**  Returned:
**     spv    double[2][3]   pv-vector: p scaled by s1, v scaled by s2
**
**  Note:
**     It is permissible for pv and spv to be the same array.
**
**  Called:
**     iauSxp       multiply p-vector by scalar
**
*/
```

```
double iauSepp(double a[3], double b[3])
/*
**  - - - - - - - -
**   i a u S e p p
**  - - - - - - - -
**
**  Angular separation between two p-vectors.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     a      double[3]    first p-vector (not necessarily unit length)
**     b      double[3]    second p-vector (not necessarily unit length)
**
**  Returned (function value):
**            double       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:
**     iauPxp       vector product of two p-vectors
**     iauPm        modulus of p-vector
**     iauPdp       scalar product of two p-vectors
**
*/
```

```
double iauSeps(double al, double ap, double bl, double bp)
/*
**  - - - - - - - -
**   i a u S e p s
**  - - - - - - - -
**
**  Angular separation between two sets of spherical coordinates.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     al     double       first longitude (radians)
**     ap     double       first latitude (radians)
**     bl     double       second longitude (radians)
**     bp     double       second latitude (radians)
**
**  Returned (function value):
**            double       angular separation (radians)
**
**  Called:
**     iauS2c       spherical coordinates to unit vector
**     iauSepp      angular separation between two p-vectors
**
*/
```

```
double iauSp00(double date1, double 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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double    TT as a 2-part Julian Date (Note 1)
**
**  Returned (function value):
**                  double    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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function.
**
**  Reference:
**
**     McCarthy, D. D., Petit, G. (eds.), IERS Conventions (2003),
**     IERS Technical Note No. 32, BKG (2004)
**
*/
```

```
int iauStarpm(double ra1, double dec1,
              double pmr1, double pmd1, double px1, double rv1,
              double ep1a, double ep1b, double ep2a, double ep2b,
              double *ra2, double *dec2,
              double *pmr2, double *pmd2, double *px2, double *rv2)
/*
**  - - - - - - - - - -
**   i a u S t a r p m
**  - - - - - - - - - -
**
**  Star proper motion:  update star catalog data for space motion.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     ra1    double     right ascension (radians), before
**     dec1   double     declination (radians), before
**     pmr1   double     RA proper motion (radians/year), before
**     pmd1   double     Dec proper motion (radians/year), before
**     px1    double     parallax (arcseconds), before
**     rv1    double     radial velocity (km/s, +ve = receding), before
**     ep1a   double     "before" epoch, part A (Note 1)
**     ep1b   double     "before" epoch, part B (Note 1)
**     ep2a   double     "after" epoch, part A (Note 1)
**     ep2b   double     "after" epoch, part B (Note 1)
**
**  Returned:
**     ra2    double     right ascension (radians), after
**     dec2   double     declination (radians), after
**     pmr2   double     RA proper motion (radians/year), after
**     pmd2   double     Dec proper motion (radians/year), after
**     px2    double     parallax (arcseconds), after
**     rv2    double     radial velocity (km/s, +ve = receding), after
**
**  Returned (function value):
**            int        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 dates 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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauStarpv
**     function 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 function iauStarpv),  it is arbitrarily set
**     to zero.  When this action occurs, 2 is added to the status.
**
**  8) The relativistic adjustment carried out in the iauStarpv function
**     involves an iterative calculation.  If the process fails to
**     converge within a set number of iterations, 4 is added to the
**     status.
**
**  Called:
**     iauStarpv    star catalog data to space motion pv-vector
**     iauPvu       update a pv-vector
**     iauPdp       scalar product of two p-vectors
**     iauPvstar    space motion pv-vector to star catalog data
**
*/
```

```
int iauStarpv(double ra, double dec,
              double pmr, double pmd, double px, double rv,
              double pv[2][3])
/*
**  - - - - - - - - - -
**   i a u S t a r p v
**  - - - - - - - - - -
**
**  Convert star catalog coordinates to position+velocity vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given (Note 1):
**     ra     double        right ascension (radians)
**     dec    double        declination (radians)
**     pmr    double        RA proper motion (radians/year)
**     pmd    double        Dec proper motion (radians/year)
**     px     double        parallax (arcseconds)
**     rv     double        radial velocity (km/s, positive = receding)
**
**  Returned (Note 2):
**     pv     double[2][3]  pv-vector (AU, AU/day)
**
**  Returned (function value):
**            int           status:
**                              0 = no warnings
**                              1 = distance overridden (Note 6)
**                              2 = excessive speed (Note 7)
**                              4 = solution didn't converge (Note 8)
**                           else = binary logical OR of the above
**
**  Notes:
**
**  1) The star data accepted by this function 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[0][0-2] dividing by the speed of light in
**     AU/day to give the light-time, and then multiplying the space
**     velocity pv[1][0-2] by this light-time and adding the result to
**     pv[0][0-2].
**
**     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 function
**     iauPvstar.
**
**  Called:
**     iauS2pv      spherical coordinates to pv-vector
**     iauPm        modulus of p-vector
**     iauZp        zero p-vector
**     iauPn        decompose p-vector into modulus and direction
**     iauPdp       scalar product of two p-vectors
**     iauSxp       multiply p-vector by scalar
**     iauPmp       p-vector minus p-vector
**     iauPpp       p-vector plus p-vector
**
**  Reference:
**
**     Stumpff, P., 1985, Astron.Astrophys. 144, 232-240.
**
*/
```

```
void iauSxp(double s, double p[3], double sp[3])
/*
**  - - - - - - -
**   i a u S x p
**  - - - - - - -
**
**  Multiply a p-vector by a scalar.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     s      double        scalar
**     p      double[3]     p-vector
**
**  Returned:
**     sp     double[3]     s * p
**
**  Note:
**     It is permissible for p and sp to be the same array.
**
*/
```

```
void iauSxpv(double s, double pv[2][3], double spv[2][3])
/*
**  - - - - - - - -
**   i a u S x p v
**  - - - - - - - -
**
**  Multiply a pv-vector by a scalar.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     s       double          scalar
**     pv      double[2][3]    pv-vector
**
**  Returned:
**     spv     double[2][3]    s * pv
**
**  Note:
**     It is permissible for pv and psv to be the same array
**
**  Called:
**     iauS2xpv     multiply pv-vector by two scalars
**
*/
```

```
void iauTr(double r[3][3], double rt[3][3])
/*
**  - - - - - -
**   i a u T r
**  - - - - - -
**
**  Transpose an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    r-matrix
**
**  Returned:
**     rt       double[3][3]    transpose
**
**  Note:
**     It is permissible for r and rt to be the same array.
**
**  Called:
**     iauCr        copy r-matrix
**
*/
```

```
void iauTrxp(double r[3][3], double p[3], double trp[3])
/*
**  - - - - - - - -
**   i a u T r x p
**  - - - - - - - -
**
**  Multiply a p-vector by the transpose of an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]   r-matrix
**     p        double[3]      p-vector
**
**  Returned:
**     trp      double[3]      r * p
**
**  Note:
**     It is permissible for p and trp to be the same array.
**
**  Called:
**     iauTr        transpose r-matrix
**     iauRxp       product of r-matrix and p-vector
**
*/
```

```
void iauTrxpv(double r[3][3], double pv[2][3], double trpv[2][3])
/*
**  - - - - - - - - - -
**   i a u T r x p v
**  - - - - - - - - - -
**
**  Multiply a pv-vector by the transpose of an r-matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Given:
**     r        double[3][3]    r-matrix
**     pv       double[2][3]    pv-vector
**
**  Returned:
**     trpv     double[2][3]    r * pv
**
**  Note:
**     It is permissible for pv and trpv to be the same array.
**
**  Called:
**     iauTr        transpose r-matrix
**     iauRxpv      product of r-matrix and pv-vector
**
*/
```

```
void iauXy06(double date1, double date2, double *x, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  canonical model.
**
**  Given:
**     date1,date2  double     TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     x,y          double     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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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
**     function iauFw2xy and as used in iauXys06a 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:
**     iauFal03     mean anomaly of the Moon
**     iauFalp03    mean anomaly of the Sun
**     iauFaf03     mean argument of the latitude of the Moon
**     iauFad03     mean elongation of the Moon from the Sun
**     iauFaom03    mean longitude of the Moon's ascending node
**     iauFame03    mean longitude of Mercury
**     iauFave03    mean longitude of Venus
**     iauFae03     mean longitude of Earth
**     iauFama03    mean longitude of Mars
**     iauFaju03    mean longitude of Jupiter
**     iauFasa03    mean longitude of Saturn
**     iauFaur03    mean longitude of Uranus
**     iauFane03    mean longitude of Neptune
**     iauFapa03    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
**
*/
```

```
void iauXys00a(double date1, double date2,
               double *x, double *y, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     x,y          double   Celestial Intermediate Pole (Note 2)
**     s            double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauXys00b function.
**
**  Called:
**     iauPnm00a    classical NPB matrix, IAU 2000A
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS00       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)
**
*/
```

```
void iauXys00b(double date1, double date2,
               double *x, double *y, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double   TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     x,y          double   Celestial Intermediate Pole (Note 2)
**     s            double   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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 function is faster, but slightly less accurate (about
**     1 mas in X,Y), than the iauXys00a function.
**
**  Called:
**     iauPnm00b    classical NPB matrix, IAU 2000B
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS00       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)
**
*/
```

```
void iauXys06a(double date1, double date2,
               double *x, double *y, double *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 function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  support function.
**
**  Given:
**     date1,date2  double  TT as a 2-part Julian Date (Note 1)
**
**  Returned:
**     x,y          double  Celestial Intermediate Pole (Note 2)
**     s            double  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
**
**         2450123.7           0.0       (JD method)
**         2451545.0       -1421.3       (J2000 method)
**         2400000.5       50123.2       (MJD method)
**         2450123.5           0.2       (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 iauXy06.
**
**  Called:
**     iauPnm06a    classical NPB matrix, IAU 2006/2000A
**     iauBpn2xy    extract CIP X,Y coordinates from NPB matrix
**     iauS06       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
**
*/
```

```
void iauZp(double p[3])
/*
**  - - - - - -
**   i a u Z p
**  - - - - - -
**
**  Zero a p-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Returned:
**     p        double[3]      p-vector
**
*/
```

```
void iauZpv(double pv[2][3])
/*
**  - - - - - - -
**   i a u Z p v
**  - - - - - - -
**
**  Zero a pv-vector.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Returned:
**     pv       double[2][3]      pv-vector
**
**  Called:
**     iauZp        zero p-vector
**
*/
```

```
void iauZr(double r[3][3])
/*
**  - - - - - -
**   i a u Z r
**  - - - - - -
**
**  Initialize an r-matrix to the null matrix.
**
**  This function is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  Status:  vector/matrix support function.
**
**  Returned:
**     r        double[3][3]    r-matrix
**
*/
```

COPYRIGHT NOTICE

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

```
*  5. You shall not cause the SOFA software to be brought into
*     disrepute, either by misuse, or use for inappropriate tasks, or
*     by inappropriate modification.
*
*  6. 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.
*
*  7. 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.
*
*  Correspondence concerning SOFA software should be addressed as
*  follows:
*
*      By email:  sofa@rl.ac.uk
*      By post:   IAU SOFA Center
*                 STFC Rutherford Appleton Laboratory
*                 Harwell Science and Innovation Campus
*                 Didcot, Oxfordshire, OX11 0QX
*                 United Kingdom
*
*-----------------------------------------------------------------------
```

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.

```
#ifndef SOFAMHDEF
#define SOFAMHDEF

/*
**  - - - - - - - - -
**   s o f a m . h
**  - - - - - - - - -
**
**  Macros used by SOFA library.
**
**  This file is part of the International Astronomical Union's
**  SOFA (Standards Of Fundamental Astronomy) software collection.
**
**  This revision:   2009 December 18
**
**  SOFA release 2009-12-31
**
**  Copyright (C) 2009 IAU SOFA Review Board.  See notes at end.
*/

#include "sofa.h"

/* Seconds of time per radian */
#define DS2R (7.2722052166430399038487112e-5)

/* Pi */
#define DPI (3.141592653589793238462643)

/* 2Pi */
#define D2PI (6.283185307179586476925287)

/* Degrees to radians */
#define DD2R (1.745329251994329576923691e-2)

/* Radians to arcseconds */
#define DR2AS (206264.8062470963551564734)

/* Arcseconds to radians */
#define DAS2R (4.848136811095359935899141e-6)

/* Arcseconds in a full circle */
#define TURNAS (1296000.0)

/* Milliarcseconds to radians */
#define DMAS2R (DAS2R / 1e3)

/* Length of tropical year B1900 (days) */
#define DTY (365.242198781)

/* Reference epoch (J2000.0), Julian Date */
#define DJ00 (2451545.0)

/* Julian Date of Modified Julian Date zero */
#define DJM0 (2400000.5)

/* Reference epoch (J2000.0), Modified Julian Date */
#define DJM00 (51544.5)

/* Seconds per day. */
#define DAYSEC (86400.0)

/* Days per Julian year */
#define DJY (365.25)

/* Days per Julian century */
#define DJC (36525.0)

/* Days per Julian millennium */
#define DJM (365250.0)

/* AU (m) */
#define DAU (149597870e3)
```

```c
/* Speed of light (AU per day) */
#define DC (DAYSEC / 499.004782)

/* dint(A) - truncate to nearest whole number towards zero (double) */
#define dint(A) ((A)<0.0?ceil(A):floor(A))

/* dnint(A) - round to nearest whole number (double) */
#define dnint(A) ((A)<0.0?ceil((A)-0.5):floor((A)+0.5))

/* dsign(A,B) - magnitude of A with sign of B (double) */
#define dsign(A,B) ((B)<0.0?-(A):(A))

#endif
```

IAU STANDARDS OF FUNDAMENTAL ASTRONOMY REVIEW BOARD

Current Membership

John Bangert	United States Naval Observatory	
Mark Calabretta	Australia Telescope National Facility	
Anne−Marie Gontier	Paris Observatory	
Catherine Hohenkerk	Her Majesty’s Nautical Almanac Office	
Wen−Jing Jin	Shanghai Observatory	
Brian Luzum	United States Naval Observatory	
Zinovy Malkin	Pulkovo Observatory, St Petersburg	
Jeffrey Percival	University of Wisconsin	
Patrick Wallace	Rutherford Appleton Laboratory	(chair)

Past Members

Wim Brouw	University of Groningen
Dennis McCarthy	United States Naval Observatory
George Kaplan	United States Naval Observatory
Skip Newhall	Jet Propulsion Laboratory

The e−mail address for the Board chair is Patrick.Wallace@stfc.ac.uk