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

ICRS Working Group Task 5: Computation Tools

Standards Of Fundamental Astronomy Review Board

2003 March 17

# 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 69 "astronomy" routines supported by 52 "vector/matrix" routines, all written in Fortran. In the future the number of astronomy routines will increase, and implementations in other languages will be introduced.

## THE SOFA INITIATIVE

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.

The SOFA Review Board is now part of Task 5 (Computation Tools) of the IAU Working Group on the International Celestial Reference System (WGICRS), which is part of Division 1 (Fundamental Astronomy). The same person chairs both the SOFA Review Board and WGICRS Task 5.

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 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 69 routines. Coverage is limited in this early release, but the areas addressed include calendars, timescales, ephemerides, precession/nutation, star space-motion, and star catalog conversions.

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. Although this library can be used in its own right, it is limited in scope to what the other SOFA routines require. Some users may be better served by the many specialist libraries available elsewhere.

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 will be added if there is a 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 filename and routine name. 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 or no 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.

## PROGRAMMING STANDARDS

The first release is in Fortran 77 only. Work on C equivalents is about to start, and related software in other languages is being considered.

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.

Fortran 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 sets out the standard, but the code itself offers a wide range of examples of what is acceptable.

The routines contain explicit numerical constants (the INCLUDE statement is not part of ANSI Fortran 77). These are drawn from the file consts.lis, which is listed in an appendix.

## 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 use by private individuals for non-profit research and by non-profit educational, academic and research institutions. Potential commercial users of the Software should contact the Board.

Further details are included in the block of comments which concludes every routine. This block of comments is also given as 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 and for chairing the SOFA Review Board are provided by the UK's Particle Physics and Astronomy Research

Council through its various astronomy programs at the Rutherford Appleton Laboratory.

# SOFA Astronomy Library

## PREFACE

The routines described here are the second release of 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.

At present the routines are all written in Fortran 77, complying with the ANSI standard (X3.9-1978) except in two respects:

(1) All routine names are prefixed with the string "iau_". If necessary, the string can be removed globally; the result is correctly functioning code.

(2) All routines include an IMPLICIT NONE statement. This can be removed without affecting the behaviour of the code.

If the "iau_" string and/or the IMPLICIT NONE statements are removed globally, the resulting code is fully ANSI-compliant and is functionally unaffected.

## GENERAL PRINCIPLES

The principal function of the SOFA Astronomy Library is to define algorithms. A secondary function is to provide software suitable for 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 optimizions. Within these constraints, the intention is that the code corresponds to the published formulation (if any).

Epochs (or simply "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 timescales. 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 important reference.

## ROUTINES

```
Calendars

   CAL2JD    Gregorian Calendar to Julian Date
   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 including Earth rotation

   DAT       Delta(AT) (=TAI-UTC) for a given UTC date
   DTDB      TDB-TT
   EE00      equation of the equinoxes, IAU 2000, given nutation
   EE00A     equation of the equinoxes, IAU 2000A
   EE00B     equation of the equinoxes, IAU 2000B
   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-compatible
   GMST82    Greenwich Mean Sidereal Time, IAU 1982
   GST00A    Greenwich Apparent Sidereal Time, IAU 2000A
   GST00B    Greenwich Apparent Sidereal Time, IAU 2000B
   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, ICRS to mean J2000, IAU 2000
   BP00      frame bias and precession matrices, IAU 2000
   BPN2XY    bias-precession-nutation matrix given CIP
   C2I00A    celestial-to-intermediate matrix, IAU 2000A
   C2I00B    celestial-to-intermediate matrix, IAU 2000B
   C2IBPN    celestial-to-intermediate matrix given b-p-n
   C2IXY     celestial-to-intermediate matrix given CIP
   C2IXYS    celestial-to-intermediate matrix given CIP and s
   C2T00A    celestial-to-terrestrial matrix, IAU 2000A
   C2T00B    celestial-to-terrestrial matrix, IAU 2000B
   C2TCEO    celestial-to-terrestrial matrix, CEO-based
   C2TEQX    celestial-to-terrestrial matrix, classical
   C2TPE     celestial-to-terrestrial matrix given nutation
   C2TXY     celestial-to-terrestrial matrix given CIP
   NUM00A    nutation matrix, IAU 2000A
   NUM00B    nutation matrix, IAU 2000B
   NUMAT     nutation matrix, generic
   NUT00A    nutation, IAU 2000A
   NUT00B    nutation, IAU 2000B
   NUT80     nutation, IAU 1980
   OBL80     mean obliquity, IAU 1980
   NUTM80    nutation matrix, IAU 1980
   PMAT00    precession matrix (including frame bias), IAU 2000
   PMAT76    precession matrix, IAU 1976
   PN00      b,p,n matrices, IAU 2000, given nutation
   PN00A     b,p,n matrices, IAU 2000A
   PN00B     b,p,n matrices, IAU 2000B
   PNM00A    celestial-to-true (b-p-n) matrix, IAU 2000A
   PNM00B    celestial-to-true (b-p-n) matrix, IAU 2000B
```

```
      POM00     polar-motion matrix, IAU 2000
      PR00      adjustments to IAU 1976 precession, IAU 2000
      PNM80     precession/nutation matrix, IAU 1976/1980
      PREC76    precession, IAU 1976
      S00       the quantity s, IAU 2000, given CIP
      S00A      the quantity s, IAU 2000A
      S00B      the quantity s, IAU 2000B
      SP00      the quantity s', IERS 2000
      XYS00A    CIP and s, IAU 2000A
      XYS00B    CIP and s, IAU 2000B

   Star space motion

      PVSTAR    star position+velocity vector to catalog coordinates
      STARPV    star catalog coordinates to position+velocity vector

   Star catalog conversions

      FK52H     transform FK5 star data into the Hipparcos frame
      FK5HIP    FK5 orientation and spin with respect to Hipparcos
      FK5HZ     FK5 to Hipparcos assuming zero Hipparcos proper motion
      H2FK5     transform Hipparcos star data into the FK5 frame
      HFK5Z     Hipparcos to FK5 assuming zero Hipparcos proper motion
      STARPM    proper motion between two epochs


CALLS

    SUBROUTINE         iau_BI00   ( DPSIBI, DEPSBI, DRA )
    SUBROUTINE         iau_BP00   ( DATE1, DATE2, RB, RP, RBP )
    SUBROUTINE         iau_BPN2XY ( RBPN, X, Y )
    SUBROUTINE         iau_C2I00A ( DATE1, DATE2, RC2I )
    SUBROUTINE         iau_C2I00B ( DATE1, DATE2, RC2I )
    SUBROUTINE         iau_C2IBPN ( DATE1, DATE2, RBPN, RC2I )
    SUBROUTINE         iau_C2IXY  ( DATE1, DATE2, X, Y, RC2I )
    SUBROUTINE         iau_C2IXYS ( X, Y, S, RC2I )
    SUBROUTINE         iau_C2T00A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
    SUBROUTINE         iau_C2T00B ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
    SUBROUTINE         iau_C2TCEO ( RC2I, ERA, RPOM, RC2T )
    SUBROUTINE         iau_C2TEQX ( RBPN, GST, RPOM, RC2T )
    SUBROUTINE         iau_C2TPE  ( TTA, TTB, UTA, UTB, DPSI, DEPS,
                                    XP, YP, RC2T )
    SUBROUTINE         iau_C2TXY  ( TTA, TTB, UTA, UTB, X, Y, XP, YP, RC2T )
    SUBROUTINE         iau_CAL2JD ( IY, IM, ID, DJM0, DJM, J )
    SUBROUTINE         iau_DAT    ( IY, IM, ID, FD, DELTAT, J )
    DOUBLE PRECISION FUNCTION
                       iau_DTDB   ( EPOCH1, EPOCH2, UT, ELONG, U, V )
    DOUBLE PRECISION FUNCTION
                       iau_EE00   ( DATE1, DATE2, EPSA, DPSI )
    DOUBLE PRECISION FUNCTION
                       iau_EE00A  ( DATE1, DATE2 )
    DOUBLE PRECISION FUNCTION
                       iau_EE00B  ( DATE1, DATE2 )
    DOUBLE PRECISION FUNCTION
                       iau_EECT00 ( DATE1, DATE2 )
    DOUBLE PRECISION FUNCTION
                       iau_EPB    ( DJ1, DJ2 )
    SUBROUTINE         iau_EPB2JD ( EPB, DJM0, DJM )
    DOUBLE PRECISION FUNCTION
                       iau_EPJ    ( DJ1, DJ2 )
    SUBROUTINE         iau_EPJ2JD ( EPJ, DJM0, DJM )
    SUBROUTINE         iau_EPV00  ( DJ1, DJ2, PVH, PVB, J )
    DOUBLE PRECISION FUNCTION
```

```
                  iau_EQEQ94 ( EPOCH1, EPOCH2 )
DOUBLE PRECISION FUNCTION
                  iau_ERA00  ( DJ1, DJ2 )
SUBROUTINE        iau_FK52H  ( R5, D5, DR5, DD5, PX5, RV5,
                               RH, DH, DRH, DDH, PXH, RVH )
SUBROUTINE        iau_FK5HIP ( R5H, S5H )
SUBROUTINE        iau_FK5HZ  ( R5, D5, EPOCH1, EPOCH2, RH, DH)
DOUBLE PRECISION FUNCTION
                  iau_GMST00 ( UTA, UTB, TTA, TTB )
DOUBLE PRECISION FUNCTION
                  iau_GMST82 ( UTA, UTB )
DOUBLE PRECISION FUNCTION
                  iau_GST00A ( UTA, UTB, TTA, TTB )
DOUBLE PRECISION FUNCTION
                  iau_GST00B ( UTA, UTB )
DOUBLE PRECISION FUNCTION
                  iau_GST94 ( UTA, UTB )
SUBROUTINE        iau_H2FK5  ( RH, DH, DRH, DDH, PXH, RVH,
                               R5, D5, DR5, DD5, PX5, RV5 )
SUBROUTINE        iau_HFK5Z  ( RH, DH, EPOCH1, EPOCH2, R5, D5, DR5, DD5 )
SUBROUTINE        iau_JD2CAL ( DJ1, DJ2, IY, IM, ID, FD, J )
SUBROUTINE        iau_JDCALF ( NDP, DJ1, DJ2, IYMDF, J )
SUBROUTINE        iau_NUM00A ( DATE1, DATE2, RMATN )
SUBROUTINE        iau_NUM00B ( DATE1, DATE2, RMATN )
SUBROUTINE        iau_NUMAT  ( EPSA, DPSI, DEPS, RMATN )
SUBROUTINE        iau_NUT00A ( DATE1, DATE2, DPSI, DEPS )
SUBROUTINE        iau_NUT00B ( DATE1, DATE2, DPSI, DEPS )
SUBROUTINE        iau_NUT80  ( EPOCH1, EPOCH2, DPSI, DEPS )
SUBROUTINE        iau_NUTM80 ( EPOCH1, EPOCH2, RMATN )
DOUBLE PRECISION FUNCTION
                  iau_OBL80  ( EPOCH1, EPOCH2 )
SUBROUTINE        iau_PLAN94 ( EPOCH1, EPOCH2, NP, PV, J )
SUBROUTINE        iau_PMAT00 ( DATE1, DATE2, RBP )
SUBROUTINE        iau_PMAT76 ( DJ1, DJ2, RMATP )
SUBROUTINE        iau_PN00   ( DATE1, DATE2, DPSI, DEPS,
                               EPSA, RB, RP, RBP, RN, RBPN )
SUBROUTINE        iau_PN00A  ( DATE1, DATE2, DPSI, DEPS, EPSA,
                               RB, RP, RBP, RN, RBPN )
SUBROUTINE        iau_PN00B  ( DATE1, DATE2, DPSI, DEPS, EPSA,
                               RB, RP, RBP, RN, RBPN )
SUBROUTINE        iau_PNM00A ( DATE1, DATE2, RBPN )
SUBROUTINE        iau_PNM00B ( DATE1, DATE2, RBPN )
SUBROUTINE        iau_PNM80  ( EPOCH1, EPOCH2, RMATPN )
SUBROUTINE        iau_POM00  ( XP, YP, SP, RPOM )
SUBROUTINE        iau_PR00   ( DATE1, DATE2, DPSIPR, DEPSPR )
SUBROUTINE        iau_PREC76 ( EP01, EP02, EP11, EP12, ZETA, Z, THETA )
SUBROUTINE        iau_PVSTAR ( PV, RA, DEC, PMR, PMD, PX, RV, J )
DOUBLE PRECISION FUNCTION
                  iau_S00    ( DATE1, DATE2, X, Y )
DOUBLE PRECISION FUNCTION
                  iau_S00A   ( DATE1, DATE2 )
DOUBLE PRECISION FUNCTION
                  iau_S00B   ( DATE1, DATE2 )
DOUBLE PRECISION FUNCTION
                  iau_SP00   ( DATE1, DATE2 )
SUBROUTINE        iau_STARPM ( RA1, DEC1, PMR1, PMD1, PX1, RV1,
                               EP1A, EP1B, EP2A, EP2B,
                               RA2, DEC2, PMR2, PMD2, PX2, RV2, J )
SUBROUTINE        iau_STARPV ( RA, DEC, PMR, PMD, PX, RV, PV, J )
SUBROUTINE        iau_XYS00A ( DATE1, DATE2, X, Y, S )
SUBROUTINE        iau_XYS00B ( DATE1, DATE2, X, Y, S )
```

# SOFA Vector/Matrix Library

## PREFACE

The routines described here are the first release of 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.

At present the routines are all written in Fortran 77, complying with the ANSI standard (X3.9-1978) except in two respects:

(1) All routine names are prefixed with the string "iau_". If necessary, the string can be removed globally; the result is correctly functioning code.

(2) All routines include an IMPLICIT NONE statement. This can be removed without affecting the behaviour of the code.

If the "iau_" string and/or the IMPLICIT NONE statements are removed globally, the resulting code is fully ANSI-compliant and is functionally unaffected.

## 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

```
   DOUBLE PRECISION FUNCTION
                     iau_ANP     ( A )
   DOUBLE PRECISION FUNCTION
                     iau_ANPM    ( A )
   SUBROUTINE        iau_A2AF    ( NDP, ANGLE, SIGN, IDMSF )
   SUBROUTINE        iau_A2TF    ( NDP, ANGLE, SIGN, IHMSF )
   SUBROUTINE        iau_C2S     ( P, THETA, PHI )
   SUBROUTINE        iau_CP      ( P, C )
   SUBROUTINE        iau_CPV     ( PV, C )
   SUBROUTINE        iau_CR      ( R, C )
   SUBROUTINE        iau_D2TF    ( NDP, DAYS, SIGN, IHMSF )
   SUBROUTINE        iau_IR      ( R )
   SUBROUTINE        iau_P2PV    ( P, PV )
   SUBROUTINE        iau_P2S     ( P, THETA, PHI, R )
   SUBROUTINE        iau_PAP     ( A, B, THETA )
   SUBROUTINE        iau_PAS     ( AL, AP, BL, BP, THETA )
   SUBROUTINE        iau_PDP     ( A, B, ADB )
   SUBROUTINE        iau_PM      ( P, R )
   SUBROUTINE        iau_PMP     ( A, B, AMB )
   SUBROUTINE        iau_PN      ( P, R, U )
   SUBROUTINE        iau_PPP     ( A, B, APB )
   SUBROUTINE        iau_PPSP    ( A, S, B, APSB )
   SUBROUTINE        iau_PV2P    ( PV, P )
   SUBROUTINE        iau_PV2S    ( PV, THETA, PHI, R, TD, PD, RD )
   SUBROUTINE        iau_PVDPV   ( A, B, ADB )
   SUBROUTINE        iau_PVM     ( PV, R, S )
   SUBROUTINE        iau_PVMPV   ( A, B, AMB )
   SUBROUTINE        iau_PVPPV   ( A, B, APB )
   SUBROUTINE        iau_PVU     ( DT, PV, UPV )
   SUBROUTINE        iau_PVUP    ( DT, PV, P )
   SUBROUTINE        iau_PVXPV   ( A, B, AXB )
   SUBROUTINE        iau_PXP     ( A, B, AXB )
   SUBROUTINE        iau_RM2V    ( R, P )
   SUBROUTINE        iau_RV2M    ( P, R )
   SUBROUTINE        iau_RX      ( PHI, R )
   SUBROUTINE        iau_RXP     ( R, P, RP )
   SUBROUTINE        iau_RXPV    ( R, PV, RPV )
   SUBROUTINE        iau_RXR     ( A, B, ATB )
   SUBROUTINE        iau_RY      ( THETA, R )
   SUBROUTINE        iau_RZ      ( PSI, R )
   SUBROUTINE        iau_S2C     ( THETA, PHI, C )
   SUBROUTINE        iau_S2P     ( THETA, PHI, R, P )
   SUBROUTINE        iau_S2PV    ( THETA, PHI, R, TD, PD, RD, PV )
   SUBROUTINE        iau_S2XPV   ( S1, S2, PV )
   SUBROUTINE        iau_SEPP    ( A, B, S )
   SUBROUTINE        iau_SEPS    ( AL, AP, BL, BP, S )
   SUBROUTINE        iau_SXP     ( S, P, SP )
   SUBROUTINE        iau_SXPV    ( S, PV, SPV )
   SUBROUTINE        iau_TR      ( R, RT )
   SUBROUTINE        iau_TRXP    ( R, P, TRP )
   SUBROUTINE        iau_TRXPV   ( R, PV, TRPV )
   SUBROUTINE        iau_ZP      ( P )
   SUBROUTINE        iau_ZPV     ( PV )
   SUBROUTINE        iau_ZR      ( R )
```

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

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

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

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

```
      SUBROUTINE iau_BI00 ( DPSIBI, DEPSBI, DRA )
*+
*  - - - - - - - - -
*   i a u _ B I 0 0
*  - - - - - - - - -
*
*  Frame bias components of IAU 2000 precession-nutation models (part of
*  MHB_2000 with additions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Returned:
*     DPSIBI,DEPSBI  d   longitude and obliquity corrections
*     DRA            d   the ICRS RA of the J2000 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 pole.  They define, with respect to the
*     GCRS frame, a J2000 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 MHB_2000 code itself was obtained on 9th September 2002
*     from ftp //maia.usno.navy.mil/conv2000/chapter5/IAU2000A.
*
*-
```

```
      SUBROUTINE iau_BP00 ( DATE1, DATE2, RB, RP, RBP )
*+
*  - - - - - - - - -
*   i a u _ B P 0 0
*  - - - - - - - - -
*
*  Frame bias and precession, IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RB           d(3,3)    frame bias matrix (Note 2)
*     RP           d(3,3)    precession matrix (Note 3)
*     RBP          d(3,3)    bias-precession matrix (Note 4)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RB transforms vectors from GCRS to mean J2000 by
*     applying frame bias.
*
*  3) The matrix RP transforms vectors from mean J2000 to mean of date
*     by applying precession.
*
*  4) The matrix RBP transforms vectors from GCRS to mean of date by
*     applying frame bias then precession.  It is the product RP x RB.
*
*  Called:
*     iau_BI00     IAU 2000 frame bias components
*     iau_PR00     IAU 2000 precession adjustments
*     iau_IR       initialize r-matrix to identity
*     iau_RX       rotate around X-axis
*     iau_RY       rotate around Y-axis
*     iau_RZ       rotate around Z-axis
*     iau_RXR      r-matrix product
*
*  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", submitted to A&A (2002)
*
*-
```

```
      SUBROUTINE iau_BPN2XY ( RBPN, X, Y )
*+
*  - - - - - - - - - - -
*   i a u _ B P N 2 X Y
*  - - - - - - - - - - -
*
*  Extract from the bias-precession-nutation matrix the X,Y coordinates
*  of the Celestial Intermediate Pole.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RBPN      d(3,3)    celestial-to-true matrix (Note 1)
*
*  Returned:
*     X,Y         d       Celestial Intermediate Pole (Note 2)
*
*  Notes:
*
*  1) The matrix RBPN transforms vectors from GCRS to true of date
*     (CIP/equinox), and so the Celestial Intermediate Pole unit vector
*     is the bottom row of the matrix.
*
*  2) X,Y are components of the Celestial Intermediate Pole unit vector
*     in the Geocentric Celestial Reference System.
*
*  Reference:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", submitted to A&A (2002)
*
*-
```

```
      SUBROUTINE iau_C2I00A ( DATE1, DATE2, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I 0 0 A
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date using the
*  IAU 2000A precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  3) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_C2I00B routine.
*
*  Called:
*     iau_PNM00A  classical bias-precession-nutation matrix, IAU 2000A
*     iau_C2IBPN  celestial-to-intermediate matrix given BPN matrix
*
*  References:
```

```
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", submitted to A&A (2002)
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2I00B ( DATE1, DATE2, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I 0 0 B
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date using the
*  IAU 2000B precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  3) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_C2I00A routine.
*
*  Called:
*     iau_PNM00B  classical bias-precession-nutation matrix, IAU 2000B
*     iau_C2IBPN  celestial-to-intermediate matrix given BPN matrix
*
*  References:
```

```
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", submitted to A&A (2002)
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2IBPN ( DATE1, DATE2, RBPN, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I B P N
*  - - - - - - - - - - -
*
*  Form the celestial-to-intermediate matrix for a given date given
*  the bias-precession-nutation matrix.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d       TT as a 2-part Julian Date (Note 1)
*     RBPN         d(3,3)    celestial-to-true matrix (Note 2)
*
*  Returned:
*     RC2I         d(3,3)    celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix RBPN transforms vectors from GCRS to true of date
*     (CIP/equinox).
*
*  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 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_BPN2XY  extract CIP coordinates from b-p-n matrix
*     iau_C2IXY   celestial-to-intermediate matrix given X,Y
*
```

```
*  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", submitted to A&A (2002)
*
*-
```

```
      SUBROUTINE iau_C2IXY ( DATE1, DATE2, X, Y, RC2I )
*+
*  - - - - - - - - - -
*   i a u _ C 2 I X Y
*  - - - - - - - - - -
*
*  Form the celestial to intermediate-frame-of-date matrix for a given
*  date when the CIP X,Y coordinates are known.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     X,Y            d      Celestial Intermediate Pole (Note 2)
*
*  Returned:
*     RC2I         d(3,3)   celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The matrix RC2I is the first stage in the transformation from
*     celestial to terrestrial coordinates:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_C2IXYS   celestial-to-intermediate matrix from X,Y,s
*     iau_S00      the quantity s, given X,Y
*
```

```
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2IXYS ( X, Y, S, RC2I )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 I X Y S
*  - - - - - - - - - - -
*
*  Form the celestial to intermediate-frame-of-date matrix given the CIP
*  X,Y and the quantity s.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     X,Y        d       Celestial Intermediate Pole (Note 1)
*     S          d       the quantity s (Note 2)
*
*  Returned:
*     RC2I     d(3,3)    celestial-to-intermediate matrix (Note 3)
*
*  Notes:
*
*  1) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  2) The quantity s (in radians) positions the Celestial Ephemeris
*     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 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RY       rotate around Y-axis
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

```
      SUBROUTINE iau_C2T00A ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T 0 0 A
*  - - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1 and
*  the polar motion, using the IAU 2000A nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 3)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth Rotation Angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) XP and YP are the "coordinates of the pole", in radians, which
*     position the Celestial Intermediate Pole in the International
*     Terrestrial Reference System (see IERS Conventions 2000).  In a
*     geocentric right-handed triad u,v,w, where the w-axis points at
*     the north geographic pole, the v-axis points towards the origin
*     of longitudes and the u axis completes the system, XP = +u and
*     YP = -v.
*
*  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 2000), RC2I is the
*     celestial-to-intermediate matrix, ERA is the Earth Rotation Angle
*     and RPOM is the polar motion matrix.
*
*  4) A faster, but slightly less accurate result (about 1 mas), can be
*     obtained by using instead the iau_C2T00B routine.  n.b. The
*     argument list for the latter omits SP.
*
*  Called:
*     iau_C2I00A   celestial-to-intermediate matrix, IAU 2000A
*     iau_ERA00    Earth Rotation Angle, IAU 2000
*     iau_SP00     the quantity s'
*     iau_POM00    polar motion matrix
*     iau_C2TCEO   construct CEO-based celestial-to-terrestrial matrix
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2T00B ( TTA, TTB, UTA, UTB, XP, YP, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T 0 0 B
*  - - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1 and
*  the polar motion, using the IAU 2000B nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     XP,YP      d       coordinates of the pole (radians, Note 2)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 3)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth Rotation Angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) XP and YP are the "coordinates of the pole", in radians, which
*     position the Celestial Intermediate Pole in the International
*     Terrestrial Reference System (see IERS Conventions 2000).  In a
*     geocentric right-handed triad u,v,w, where the w-axis points at
*     the north geographic pole, the v-axis points towards the origin
*     of longitudes and the u axis completes the system, XP = +u and
*     YP = -v.
*
*  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 2000), RC2I is the
*     celestial-to-intermediate matrix, ERA is the Earth Rotation Angle
*     and RPOM is the polar motion matrix.
*
*  4) The present routine is faster, but slightly less accurate (about
*     1 mas), than the iau_C2T00A routine.
*
*  Called:
*     iau_C2I00B   celestial-to-intermediate matrix, IAU 2000B
*     iau_ERA00    Earth Rotation Angle, IAU 2000
*     iau_POM00    polar motion matrix
*     iau_C2TCEO   construct CEO-based celestial-to-terrestrial matrix
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2TCEO ( RC2I, ERA, RPOM, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T C E O
*  - - - - - - - - - - -
*
*  Assemble the celestial to terrestrial matrix from CEO-based
*  components (the celestial-to-intermediate matrix, the Earth Rotation
*  Angle and the polar motion matrix).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RC2I     d(3,3)    celestial-to-intermediate matrix
*     ERA        d       Earth Rotation Angle
*     RPOM     d(3,3)    polar-motion matrix
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix
*
*  Notes:
*
*  1) This routine constructs the rotation matrix that transforms
*     vectors in the celestial system into vectors in the terrestrial
*     system.  It does so starting from precomputed components, namely
*     the matrix which rotates from celestial coordinates to the
*     intermediate frame, the Earth Rotation Angle and the polar motion
*     matrix.  One use of the present routine is when generating a
*     series of celestial-to-terrestrial matrices where only the Earth
*     Rotation Angle changes, avoiding the considerable overhead of
*     recomputing the precession-nutation more often than necessary to
*     achieve given accuracy objectives.
*
*  2) The relationship between the arguments is as follows:
*
*        [TRS]  =  RPOM * R_3(ERA) * RC2I * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2000).
*
*  Called:
*     iau_CR       copy r-matrix
*     iau_RZ       rotate around Z-axis
*     iau_RXR      r-matrix product
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2TEQX ( RBPN, GST, RPOM, RC2T )
*+
*  - - - - - - - - - - -
*   i a u _ C 2 T E Q X
*  - - - - - - - - - - -
*
*  Assemble the celestial to terrestrial matrix from equinox-based
*  components (the celestial-to-true matrix, the Greenwich Apparent
*  Sidereal Time and the polar motion matrix).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RBPN     d(3,3)    celestial-to-true matrix
*     GST        d       Greenwich (apparent) Sidereal Time
*     RPOM     d(3,3)    polar-motion matrix
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 2)
*
*  Notes:
*
*  1) This routine constructs the rotation matrix that transforms
*     vectors in the celestial system into vectors in the terrestrial
*     system.  It does so starting from precomputed components, namely
*     the matrix which rotates from celestial coordinates to the
*     true equator and equinox of date, the Greenwich Apparent Sidereal
*     Time and the polar motion matrix.  One use of the present routine
*     is when generating a series of celestial-to-terrestrial matrices
*     where only the Sidereal Time changes, avoiding the considerable
*     overhead of recomputing the precession-nutation more often than
*     necessary to achieve given accuracy objectives.
*
*  2) The relationship between the arguments is as follows:
*
*        [TRS]  =  RPOM * R_3(GST) * RBPN * [CRS]
*
*               =  RC2T * [CRS]
*
*     where [CRS] is a vector in the Geocentric Celestial Reference
*     System and [TRS] is a vector in the International Terrestrial
*     Reference System (see IERS Conventions 2000).
*
*  Called:
*     iau_CR       copy r-matrix
*     iau_RZ       rotate around Z-axis
*     iau_RXR      r-matrix product
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2TPE ( TTA, TTB, UTA, UTB, DPSI, DEPS, XP, YP,
     :                       RC2T )
*+
*  - - - - - - - - - -
*   i a u _ C 2 T P E
*  - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1, the
*  nutation and the polar motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB     d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB     d       UT1 as a 2-part Julian Date (Note 1)
*     DPSI,DEPS   d       nutation (Note 2)
*     XP,YP       d       coordinates of the pole (radians, Note 3)
*
*  Returned:
*     RC2T      d(3,3)    celestial-to-terrestrial matrix (Note 4)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth Rotation Angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) The caller is responsible for providing the nutation components;
*     they are in longitude and obliquity, in radians and are with
*     respect to the equinox and ecliptic of date.  For high-accuracy
*     applications, free core nutation should be included as well as
*     any other relevant corrections to the position of the CIP.
*
*  3) XP and YP are the "coordinates of the pole", in radians, which
*     position the Celestial Intermediate Pole in the International
*     Terrestrial Reference System (see IERS Conventions 2000).  In a
*     geocentric right-handed triad u,v,w, where the w-axis points at
*     the north geographic pole, the v-axis points towards the origin
*     of longitudes and the u axis completes the system, XP = +u and
*     YP = -v.
```

```
*
*  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 2000), RBPN is the
*     bias-precession-nutation matrix, GST is the Greenwich (apparent)
*     Sidereal Time and RPOM is the polar motion matrix.
*
*  Called:
*     iau_PN00     bias/precession/nutation results
*     iau_GMST00   Greenwich Mean Sidereal Time, IAU 2000
*     iau_SP00     the quantity s'
*     iau_EE00     equation of the equinoxes, IAU 2000
*     iau_POM00    polar motion matrix
*     iau_C2TEQX   form equinox-based celestial-to-terrestrial matrix
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      SUBROUTINE iau_C2TXY ( TTA, TTB, UTA, UTB, X, Y, XP, YP, RC2T )
*+
*  - - - - - - - - - -
*   i a u _ C 2 T X Y
*  - - - - - - - - - -
*
*  Form the celestial to terrestrial matrix given the date, the UT1, the
*  CIP coordinates and the polar motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     TTA,TTB    d       TT as a 2-part Julian Date (Note 1)
*     UTA,UTB    d       UT1 as a 2-part Julian Date (Note 1)
*     X,Y        d       Celestial Intermediate Pole (Note 2)
*     XP,YP      d       coordinates of the pole (radians, Note 3)
*
*  Returned:
*     RC2T     d(3,3)    celestial-to-terrestrial matrix (Note 4)
*
*  Notes:
*
*  1) The TT and UT1 dates TTA+TTB and UTA+UTB are Julian Dates,
*     apportioned in any convenient way between the arguments UTA and
*     UTB.  For example, JD(UT1)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  In the case of UTA,UTB, the
*     date & time method is best matched to the Earth Rotation Angle
*     algorithm used:  maximum accuracy (or, at least, minimum noise) is
*     delivered when the UTA argument is for 0hrs UT1 on the day in
*     question and the UTB argument lies in the range 0 to 1, or vice
*     versa.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) XP and YP are the "coordinates of the pole", in radians, which
*     position the Celestial Intermediate Pole in the International
*     Terrestrial Reference System (see IERS Conventions 2000).  In a
*     geocentric right-handed triad u,v,w, where the w-axis points at
*     the north geographic pole, the v-axis points towards the origin
*     of longitudes and the u axis completes the system, XP = +u and
*     YP = -v.
*
*  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 2000), ERA is the Earth
*     Rotation Angle and RPOM is the polar motion matrix.
*
*  Called:
*     iau_C2IXY    celestial-to-intermediate matrix given X,Y
*     iau_ERA00    Earth Rotation Angle, IAU 2000
*     iau_SP00     the quantity s'
*     iau_POM00    polar motion matrix
*     iau_C2TCEO   construct CEO-based celestial-to-terrestrial matrix
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

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

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

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

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

```
      SUBROUTINE iau_DAT ( IY, IM, ID, FD, DELTAT, J )
*+
*  - - - - - - - -
*   i a u _ D A T
*  - - - - - - - -
*
*  For a given UTC date, calculate delta(AT) = TAI-UTC.
*
*     :------------------------------------------:
*     :                                          :
*     :                 IMPORTANT                :
*     :                                          :
*     :  A new version of this routine must be   :
*     :  produced whenever a new leap second is  :
*     :  announced.  There are three items to    :
*     :  change on each such occasion:           :
*     :                                          :
*     :  1) The parameter NDAT must be           :
*     :     increased by 1.                      :
*     :                                          :
*     :  2) A new line must be added to the set  :
*     :     of DATA statements that initialize   :
*     :     the arrays IDATE and DATS.           :
*     :                                          :
*     :  3) The parameter IYV must be set to     :
*     :     the current year.                    :
*     :                                          :
*     :  Change (3) must also be carried out     :
*     :  whenever the routine is re-issued,      :
*     :  even if no leap seconds have been       :
*     :  added.                                  :
*     :                                          :
*     :  Latest leap second:  1999 January 1     :
*     :                                          :
*     :__________________________________________:
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     IY       i     UTC:  year (Notes 1 and 2)
*     IMO      i           month (Note 2)
*     ID       i           day (Notes 2 and 3)
*     FD       d           fraction of day (Note 4)
*
*  Returned:
*     DELTAT   d     TAI minus UTC, seconds
*     J        i     status (Note 5):
*                       1 = dubious year (Note 1)
*                       0 = OK
*                      -1 = bad year
*                      -2 = bad month
*                      -3 = bad day (Note 3)
*                      -4 = bad fraction (Note 4)
*
*  Notes:
*
*  1) UTC began at 1960 January 1.0 (JD 2436934.5) and it is improper
*     to call the routine with an earlier epoch.  If this is attempted,
```

```
*     zero is returned together with a warning status.
*
*     Because leap seconds cannot, in principle, be predicted in
*     advance, a reliable check for dates beyond the valid range is
*     impossible.  To guard against gross errors, a year five or more
*     after the release year of the present routine (see parameter IYV)
*     is considered dubious.  In this case a warning status is returned
*     but the result is computed in the normal way.
*
*     For both too-early and too-late years, the warning status is J=+1.
*     This is distinct from the error status J=-1, which signifies a
*     year so early that JD could not be computed.
*
*  2) If the specified date is for a day which ends with a leap second,
*     the 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
*     routine, in order to avoid confusion near leap seconds.
*
*  4) The fraction of day is used only for dates before the introduction
*     of leap seconds, the first of which occurred at the end of 1971.
*     It is tested for validity (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, JSTAT=-2 (bad month)
*     will be returned.
*
*  6) In cases where a valid result is not available, zero is returned.
*
*  References:
*
*  1) For epochs from 1961 January 1 onwards, the expressions from the
*     file ftp://maia.usno.navy.mil/ser7/tai-utc.dat are used.
*
*  2) The 5ms timestep at 1961 January 1 is taken from 2.58.1 (p87) of
*     the 1992 Explanatory Supplement.
*
*  Called:
*     iau_CAL2JD  Gregorian calendar to Julian Day Number
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_DTDB ( EPOCH1, EPOCH2, UT,
     :                                     ELONG, U, V )
*+
*  - - - - - - - - -
*   i a u _ D T D B
*  - - - - - - - - -
*
*  The periodic part of the offset between coordinate time in the
*  solar system barycenter frame of reference and proper time for a
*  clock on the Earth.
*
*  The routine is an implementation of TDB-TT, where TDB is barycentric
*  dynamical time and TT is terrestrial time.  It also models the
*  periodic component of the conversion from geocentric coordinate time
*  (TCG) to barycentric coordinate time (TCB).
*
*  The different proper/coordinate timescales are related to each other
*  and to TAI as shown:
*
*                TAI                <- physically realized
*                 :
*              offset               <- observed (currently +32.184s)
*                 :
*                TT                 <- proper time on the geoid
*                 :
*        rate adjustment (L_G)      <- definition of TT
*                 :
*                TCG                <- coordinate time at the geocenter
*                 :
*           periodic terms          <- iau_DTDB is an implementation
*                 :
*        rate adjustment (L_C)      <- modeled
*                 :
*                TCB                <- coordinate time at the barycenter
*                 :
*        rate adjustment (-L_B)     <- modeled
*                 :
*                TDB                <- superseded
*                 :
*           periodic terms          <- -iau_DTDB is an implementation
*                 :
*                TT
*
*  Adopted values for the various constants can be found in the IERS
*  Conventions (McCarthy 1996).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     EPOCH1,EPOCH2   d   epoch, TDB (Notes 1-3)
*     UT              d   universal time (UT1, fraction of one day)
*     ELONG           d   longitude (east +ve, radians)
*     U               d   distance from Earth spin axis (km)
*     V               d   distance north of equatorial plane (km)
*
*  Returned:
*    iau_DTDB         d   TDB-TT (seconds)
*
```

```
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 is a Julian Date, apportioned in any
*     convenient way between the arguments EPOCH1 and EPOCH2.  For
*     example, JD(TDB)=2450123.7 could be expressed in any of these
*     ways, among others:
*
*           EPOCH1          EPOCH2
*
*          2450123.7D0        0D0        (JD method)
*           2451545D0      -1421.3D0     (J2000 method)
*          2400000.5D0     50123.2D0     (MJD method)
*          2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*     Although the epoch 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) TDB is a canonical implementation of the coordinate time in the
*     Solar System barycenter frame of reference, and TT is the proper
*     time given by clocks at mean sea level on the Earth.  The present
*     routine provides a TDB-TT implementation.  The result, which is
*     in seconds, has 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.
*     The variation arises from the transverse Doppler effect and the
*     gravitational red-shift as the observer varies in speed and
*     experiences different gravitational potentials.
*
*  3) The IAU 1976 definition of TDB was that it must differ from TT
*     (then called TDT) only by periodic terms.  Though practical, this
*     is an imprecise definition which ignores the existence of very
*     long-period and secular effects in the dynamics of the solar
*     system.  Consequently, different implementations of TDB will, in
*     general, have different zero-points and will drift linearly
*     relative to one other.  Subsequent IAU resolutions (1991)
*     introduced new timescales to overcome these objections, namely
*     TCG and TCB (see Seidelmann et al. 1992 for details).
*
*  4) The geocentric model is that of Fairhead & Bretagnon (1990), in
*     its full form.  It was originally supplied by Fairhead (private
*     communications with P.T.Wallace, 1990) in the form of a Fortran
*     subroutine.  The present routine contains an adaptation of the
*     Fairhead code.  The numerical results are essentially unaffected
*     by the changes, the differences with respect to the Fairhead &
*     Bretagnon original being at the 1E-20 sec level.  This model has
*     been endorsed by the IERS Conventions (1996).
*
*     The topocentric part of the model is from Moyer (1981) and
*     Murray (1983), with fundamental arguments adapted from
*     Simon et al. 1994.  It is an approximation to the expression
*     ( v / c ) . ( r / c ), where v is the barycentric velocity of
*     the Earth, r is the geocentric position of the observer and
*     c is the speed of light.
*
*     By supplying zeroes for U and V, the topocentric part of the
```

```
*     model can be nullified, and the routine will return the Fairhead
*     & Bretagnon result alone.
*
*  5) During the interval 1950-2050, the absolute accuracy is better
*     than +/- 3 nanoseconds relative to direct numerical integrations
*     using the JPL DE200/LE200 solar system ephemeris.
*
*  6) It must be stressed that the present routine is merely a model,
*     and that numerical integration of solar-system ephemerides is the
*     definitive method for predicting the relationship between TCG and
*     TCB.  In particular, published values of L_C, the constant
*     component of the relationship, may not be optimally matched with
*     the periodic component modeled here.
*
*  References:
*
*     Fairhead,L., & Bretagnon,P., Astron.Astrophys., 229, 240-247
*     (1990).
*
*     McCarthy,D.D., IERS Conventions (1996), IERS Technical Note 21,
*     Observatoire de Paris (1996).
*
*     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., 1994, Astron.Astrophys., 282, 663-683.
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE00 ( DATE1, DATE2, EPSA, DPSI )
*+
*  - - - - - - - - -
*   i a u _ E E 0 0
*  - - - - - - - - -
*
*  The equation of the equinoxes, compatible with IAU 2000 resolutions,
*  given the nutation in longitude and the mean obliquity.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     EPSA           d      mean obliquity (Note 2)
*     DPSI           d      nutation in longitude (Note 3)
*
*  Returned:
*     iau_EE00       d      equation of the equinoxes (Note 4)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The obliquity, in radians, is mean of date.
*
*  3) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  4) The result is compatible with the IAU 2000 resolutions.  For
*     further details, see McCarthy (2002) and Capitaine et al. (2002).
*
*  Called:
*     iau_EECT00    equation of the equinoxes complementary terms
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
```

```
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EE00A ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ E E 0 0 A
*  - - - - - - - - - -
*
*  Equation of the equinoxes, compatible with IAU 2000 resolutions.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EE00A      d      equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  3) The result is compatible with the IAU 2000 resolutions.  For
*     further details, see McCarthy (2002) and Capitaine et al. (2002).
*
*  Called:
*     iau_PR00      IAU 2000 precession adjustments
*     iau_OBL80     mean obliquity, IAU 1980
*     iau_NUT00A    nutation, IAU 2000A
*     iau_EE00      equation of the equinoxes, IAU 2000
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
```

*–

```
      DOUBLE PRECISION FUNCTION iau_EE00B ( DATE1, DATE2 )
*+
*  - - - - - - - - - -
*   i a u _ E E 0 0 B
*  - - - - - - - - - -
*
*  Equation of the equinoxes, compatible with IAU 2000 resolutions but
*  using the truncated nutation model IAU 2000B.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d     TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_EE00B     d     equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  3) The result is compatible with the IAU 2000 resolutions except that
*     accuracy has been compromised for the sake of speed.  For further
*     details, see McCarthy & Luzum (2001), McCarthy (2002), Capitaine
*     et al. (2002).
*
*  Called:
*     iau_PR00      IAU 2000 precession adjustments
*     iau_OBL80     mean obliquity, IAU 1980
*     iau_NUT00B    nutation, IAU 2000B
*     iau_EE00      equation of the equinoxes, IAU 2000
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
```

```
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*     McCarthy, D.D. & Luzum, B.J., "An Abridged Model of the Motion
*     of the Celestial Pole", preprint, 2001.
*
*-
```

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

```
*     By convention, the complementary terms are included as part of the
*     equation of the equinoxes rather than as part of the mean Sidereal
*     Time.  This slightly compromises the "geometrical" interpretation
*     of mean sidereal time but is otherwise inconsequential.
*
*     The present routine computes CT in the above expression, compatible
*     with IAU 2000 resolutions (Capitaine et al., 2002, and McCarthy,
*     2002).
*
*  Called:
*     iau_ANPM      normalize angle into range +/- pi
*
*  References:
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*     Capitaine, N. & Gontier, A.-M., Astron. Astrophys., 275,
*     645-650 (1993)
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

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

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

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

```
      SUBROUTINE iau_EPV00 ( EPOCH1, EPOCH2, PVH, PVB, JSTAT )
*+
*  - - - - - - - - - -
*   i a u _ E P V 0 0
*  - - - - - - - - - -
*
*  Earth position and velocity, heliocentric and barycentric, with
*  respect to the International Celestial Reference Frame.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPOCH1    d         TDB epoch part A (Note 1)
*     EPOCH2    d         TDB epoch part B (Note 1)
*
*  Returned:
*     PVH       d(3,2)    heliocentric Earth position/velocity (AU,AU/day)
*     PVB       d(3,2)    barycentric Earth position/velocity (AU,AU/day)
*     JSTAT     i         status: 0 = OK
*                                +1 = warning: date outside 1900-2100 AD
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     However, the accuracy of the result is more likely to be
*     limited by the algorithm itself than the way the epoch has been
*     expressed.
*
*  2) On return, the arrays PVH and PVB contain the following:
*
*        PVH(1,1)  x       }
*        PVH(2,1)  y       } heliocentric position, AU
*        PVH(3,1)  z       }
*
*        PVH(1,2)  xdot    }
*        PVH(2,2)  ydot    } heliocentric velocity, AU/d
*        PVH(3,2)  zdot    }
*
*        PVB(1,1)  x       }
*        PVB(2,1)  y       } barycentric position, AU
*        PVB(3,1)  z       }
```

```
*
*        PVB(1,2)  xdot    }
*        PVB(2,2)  ydot    } barycentric velocity, AU/d
*        PVB(3,2)  zdot    }
*
*     The vectors are with respect to the International Celestial
*     Reference Frame.  The time unit is one day in TDB.
*
*  3) The routine is a SIMPLIFIED SOLUTION from the planetary theory
*     VSOP2000 (X. Moisson, P. Bretagnon, 2001, Celes. Mech., to be
*     published) and is an adaptation of original Fortran code
*     supplied by P. Bretagnon (private communication, 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
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_EQEQ94 ( EPOCH1, EPOCH2 )
*+
*  - - - - - - - - - - -
*   i a u _ E Q E Q 9 4
*  - - - - - - - - - - -
*
*  Equation of the equinoxes, IAU 1994 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     EPOCH1,EPOCH2   d         TDB epoch (Note 1)
*
*  Returned:
*     iau_EQEQ94      d         equation of the equinoxes (Note 2)
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result, which is in radians, operates in the following sense:
*
*        Greenwich apparent ST = GMST + equation of the equinoxes
*
*  Called:
*     iau_NUT80     nutation, IAU 1980
*     iau_OBL80     mean obliquity, IAU 1980
*
*  References:
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*     Capitaine, N. & Gontier, A.-M., Astron. Astrophys., 275,
*     645-650 (1993)
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_ERA00 ( DJ1, DJ2 )
*+
*  - - - - - - - - - -
*   i a u _ E R A 0 0
*  - - - - - - - - - -
*
*  Earth rotation angle (IAU 2000 model).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DJ1,DJ2     d      UT1 as a 2-part Julian Date (see note)
*
*  The result is the Earth Rotation Angle (radians), in the range 0 to
*  2pi.
*
*  Notes:
*
*  1) The UT1 date DJ1+DJ2 is a Julian Date, apportioned in any
*     convenient way between the arguments DJ1 and DJ2.  For example,
*     JD(UT1)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*             DJ1            DJ2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  The date & time method is
*     best matched to the algorithm used:  maximum accuracy (or, at
*     least, minimum noise) is delivered when the DJ1 argument is for
*     0hrs UT1 on the day in question and the DJ2 argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The algorithm is adapted from Expression 22 of Capitaine et al
*     2000.  The time argument has been expressed in days directly,
*     and, to retain precision, integer contributions have been
*     eliminated.  A similar formulation is given on p35 of the IERS
*     Conventions (1996).
*
*  Called:
*     iau_ANP        normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine N., Guinot B. and McCarthy D.D, 2000, A&A 355,
*     398-405.
*
*     McCarthy D.D., 1996, IERS Conventions (IERS Technical Note 21),
*     Observatoire de Paris.
*
*-
```

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

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

```
      SUBROUTINE iau_FK5HZ ( R5, D5, EPOCH1, EPOCH2, RH, DH )
*+
*  - - - - - - - - - -
*   i a u _ F K 5 H Z
*  - - - - - - - - - -
*
*  Transform an FK5 (J2000) star position into the frame of the
*  Hipparcos catalogue, assuming zero Hipparcos proper motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     R5              d      FK5 RA (radians), equinox J2000, at epoch
*     D5              d      FK5 Dec (radians), equinox J2000, at epoch
*     EPOCH1,EPOCH2   d      TDB epoch (Notes 1,2)
*
*  Returned:
*     RH              d      Hipparcos RA (radians)
*     DH              d      Hipparcos Dec (radians)
*
*  Notes:
*
*  1) This routine converts a star position from the FK5 system to
*     the Hipparcos system, in such a way that the Hipparcos proper
*     motion is zero.  Because such a star has, in general, a non-zero
*     proper motion in the FK5 system, the routine requires the epoch
*     at which the position in the FK5 system was determined.
*
*  2) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure
*     rotation and spin;  zonal errors in the FK5 catalogue are
*     not taken into account.
*
*  4) It was the intention that Hipparcos should be a close
*     approximation to an inertial frame, so that distant objects
*     have zero proper motion;  such objects have (in general)
*     non-zero proper motion in FK5, and this routine returns those
*     fictitious proper motions.
*
*  5) The position returned by this routine is in the FK5 J2000
```

```
*     reference frame but at Julian epoch EPOCH1+EPOCH2.
*
*  6) See also iau_FK52H, iau_H2FK5, iau_HFK5Z.
*
*  Called:
*     iau_S2C      spherical to unit vector
*     iau_FK5HIP   FK5 rotation and spin wrt to Hipparcos
*     iau_SXP      product of scalar and p-vector
*     iau_RV2M     r-vector to r-matrix
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_PXP      outer (=vector=cross) product of two p-vectors
*     iau_C2S      unit vector to spherical
*     iau_ANP      normalize radians to range 0 to 2pi
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GMST00 ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G M S T 0 0
*  - - - - - - - - - - -
*
*  Greenwich Mean Sidereal Time (model consistent with IAU 2000
*  resolutions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  The result is the Greenwich Mean Sidereal Time (radians), in the
*  range 0 to 2pi.
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*        2450123.7D0        0D0        (JD method)
*         2451545D0      -1421.3D0     (J2000 method)
*        2400000.5D0     50123.2D0     (MJD method)
*        2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession.  If UT1 is used for
*     both purposes, errors of order 100 microarcseconds result.
*
*  3) This GMST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation and equation of the equinoxes.
*
*  4) The algorithm is from Capitaine et al. (2002) and McCarthy (2002).
*
*  Called:
*     iau_ERA00      Earth Rotation Angle, IAU 2000
*     iau_ANP        normalize angle into range 0 to 2pi
*
*  References:
```

```
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

```
*
*     Aoki et al., Astron. Astrophys. 105, 359-361 (1982).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST00A ( UTA, UTB, TTA, TTB )
*+
*  - - - - - - - - - - -
*   i a u _ G S T 0 0 A
*  - - - - - - - - - - -
*
*  Greenwich Apparent Sidereal Time (consistent with IAU 2000
*  resolutions).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*     TTA, TTB     d      TT as a 2-part Julian Date (Notes 1,2)
*
*  The result is the Greenwich Apparent Sidereal Time (radians), in the
*  range 0 to 2pi.
*
*  Notes:
*
*  1) The UT1 and TT dates UTA+UTB and TTA+TTB respectively, are both
*     Julian Dates, apportioned in any convenient way between the
*     argument pairs.  For example, JD=2450123.7 could be expressed in
*     any of these ways, among others:
*
*            Part A         Part B
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable (in the case of UT;  the TT is not at all critical
*     in this respect).  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) Both UT1 and TT are required, UT1 to predict the Earth rotation
*     and TT to predict the effects of precession-nutation.  If UT1 is
*     used for both purposes, errors of order 100 microarcseconds result.
*
*  3) This GAST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation.
*
*  4) The algorithm is from Capitaine et al. (2002) and McCarthy (2002).
*
*  Called:
*     iau_GMST00     Greenwich Mean Sidereal Time, IAU 2000
*     iau_EE00A      equation of the equinoxes, IAU 2000A
*     iau_ANP        normalize angle into range 0 to 2pi
*
```

```
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_GST00B ( UTA, UTB )
*+
*  - - - - - - - - - - -
*   i a u _ G S T 0 0 B
*  - - - - - - - - - - -
*
*  Greenwich Apparent Sidereal Time (consistent with IAU 2000
*  resolutions but using the truncated nutation model IAU 2000B).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     UTA, UTB     d      UT1 as a 2-part Julian Date (Notes 1,2)
*
*  The result is the Greenwich Apparent Sidereal Time (radians), in the
*  range 0 to 2pi.
*
*  Notes:
*
*  1) The UT1 date UTA+UTB is a Julian Date, apportioned in any
*     convenient way between the argument pair.  For example,
*     JD=2450123.7 could be expressed in any of these ways, among
*     others:
*
*             UTA            UTB
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 and MJD methods are good compromises
*     between resolution and convenience.  For UT, the date & time
*     method is best matched to the algorithm that is used by the Earth
*     Rotation Angle routine, called internally:  maximum accuracy (or,
*     at least, minimum noise) is delivered when the UTA argument is for
*     0hrs UT1 on the day in question and the UTB argument lies in the
*     range 0 to 1, or vice versa.
*
*  2) The result is compatible with the IAU 2000 resolutions, except
*     that accuracy has been compromised for the sake of speed and
*     convenience in two respects:
*
*     . UT is used instead of TDB (or TT) to compute the precession
*       component of GMST and the equation of the equinoxes.  This
*       results in errors of order 0.1 mas at present.
*
*     . The IAU 2000B abridged nutation model (McCarthy & Luzum, 2001)
*       is used, introducing errors of up to 1 mas.
*
*  3) This GAST is compatible with the IAU 2000 resolutions and must be
*     used only in conjunction with other IAU 2000 compatible components
*     such as precession-nutation.
*
*  4) The algorithm is from Capitaine et al. (2002) and McCarthy (2002).
*
```

```
*  Called:
*     iau_GMST00     Greenwich Mean Sidereal Time, IAU 2000
*     iau_EE00B      equation of the equinoxes, IAU 2000B
*     iau_ANP        normalize angle into range 0 to 2pi
*
*  References:
*
*     Capitaine, N., Wallace, P.T. and McCarthy, D.D., "Expressions for
*     the Earth Rotation Angle and Sidereal Time consistent with the IAU
*     2000A precession-nutation model", in preparation (2002).
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*     McCarthy, D.D. & Luzum, B.J., "An Abridged Model of the Motion
*     of the Celestial Pole", preprint, 2001.
*
*-
```

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

```
*     P. Kenneth Seidelmann (ed), University Science Books (1992)
*
*     IAU Resolution C7, Recommendation 3 (1994)
*
*-
```

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

```
      SUBROUTINE iau_HFK5Z ( RH, DH, EPOCH1, EPOCH2, R5, D5, DR5, DD5 )
*+
*  - - - - - - - - - -
*   i a u _ H F K 5 Z
*  - - - - - - - - - -
*
*  Transform a Hipparcos star position into FK5 J2000, assuming
*  zero Hipparcos proper motion.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     RH              d      Hipparcos RA (radians)
*     DH              d      Hipparcos Dec (radians)
*     EPOCH1,EPOCH2   d      TDB epoch (Note 1)
*
*  Returned (all FK5, equinox J2000, epoch EPOCH1+EPOCH2):
*     R5              d      RA (radians)
*     D5              d      Dec (radians)
*     DR5             d      FK5 RA proper motion (rad/year, Note 4)
*     DD5             d      Dec proper motion (rad/year, Note 4)
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The proper motion in RA is dRA/dt rather than cos(Dec)*dRA/dt.
*
*  3) The FK5 to Hipparcos transformation is modeled as a pure
*     rotation and spin;  zonal errors in the FK5 catalogue are
*     not taken into account.
*
*  4) It was the intention that Hipparcos should be a close
*     approximation to an inertial frame, so that distant objects
*     have zero proper motion;  such objects have (in general)
*     non-zero proper motion in FK5, and this routine returns those
*     fictitious proper motions.
*
*  5) The position returned by this routine is in the FK5 J2000
*     reference frame but at Julian epoch EPOCH1+EPOCH2.
*
```

```
*  6) See also iau_FK52H, iau_H2FK5, iau_FK5ZHZ.
*
*  Called:
*     iau_S2C      spherical to unit vector
*     iau_FK5HIP   FK5 rotation and spin wrt to Hipparcos
*     iau_RXP      product of r-matrix and p-vector
*     iau_SXP      product of scalar and p-vector
*     iau_RXR      r-matrix multiply
*     iau_TRXP     product of transpose of r-matrix and p-vector
*     iau_PXP      vector product of two p-vectors
*     iau_PV2S     pv-vector to spherical
*     iau_ANP      normalize radians to range 0 to 2pi
*
*  Reference:
*
*     F.Mignard & M.Froeschle, Astron. Astrophys. 354, 732-739 (2000).
*
*-
```

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

```
      SUBROUTINE iau_JD2CAL ( DJ1, DJ2, IY, IM, ID, FD, J )
*+
*  - - - - - - - - - - -
*   i a u _ J D 2 C A L
*  - - - - - - - - - - -
*
*  Julian Date to Gregorian year, month, day, and fraction of a day.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DJ1,DJ2     d     Julian Date (Notes 1, 2)
*
*  Returned:
*     IY          i     year
*     IM          i     month
*     ID          i     day
*     FD          d     fraction of day
*     J           i     status:
*                           0 = OK
*                          -1 = unacceptable date (Note 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
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*  3) In early eras the conversion is from the "Proleptic Gregorian
*     Calendar";  no account is taken of the date(s) of adoption of
*     the Gregorian Calendar, nor is the AD/BC numbering convention
*     observed.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 12.92 (p604).
*
*-
```

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

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

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

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

```
      SUBROUTINE iau_NUT00A ( DATE1, DATE2, DPSI, DEPS )
*+
*  - - - - - - - - - - -
*   i a u _ N U T 0 0 A
*  - - - - - - - - - - -
*
*  Nutation, IAU 2000A model (MHB_2000 luni-solar and planetary nutation
*  with free core nutation omitted).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSI,DEPS     d    nutation, luni-solar + planetary (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components in longitude and obliquity are with
*     respect to the equinox and ecliptic of date.  The obliquity at
*     J2000 is assumed to be the Lieske et al. (1977) value of
*     84381.448 arcsec.
*
*     Both the luni-solar and planetary nutations are included.  The
*     latter are due to direct planetary nutations and the perturbations
*     of the lunar and terrestrial orbits.
*
*  3) The routine computes the MHB_2000 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 MHB_2000 model also contains contributions to the
*     nutations in longitude and obliquity due to the free-excitation of
*     the free-core-nutation during the period 1979-2000.  These FCN
*     terms, which are time-dependent and unpredictable, are NOT
```

```
*     included in the present routine and, if required, must be
*     independently computed.  With the FCN corrections included, the
*     present routine delivers a pole which is at current epochs
*     accurate to a few hundred microarcseconds.  The omission of FCN
*     introduces further errors of about that size.
*
*  5) The present routine provides classical nutation.  The MHB_2000
*     algorithm, from which it is adapted, deals also with (i) the
*     offsets between the GCRS and mean poles and (ii) the adjustments
*     in longitude and obliquity due to the changed precession rates.
*     These additional functions, namely frame bias and precession
*     adjustments, are supported by the SOFA routines iau_BI00 and
*     iau_PR00.
*
*  6) The MHB_2000 algorithm also provides "total" nutations, comprising
*     the arithmetic sum of the frame bias, precession adjustments,
*     luni-solar nutation and planetary nutation.  These total nutations
*     can be used in combination with an existing IAU 1976 precession
*     implementation, such as iau_PMAT76, to deliver GCRS-to-true
*     predictions of sub-mas accuracy at current epochs.  However, there
*     are three shortcomings in the MHB_2000 model that must be taken
*     into account if more accurate or definitive results are required
*     (see Wallace 2002)
*
*       (i) The MHB_2000 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 MHB_2000 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 MHB_2000 model predates the determination by Chapront
*           et al. (2002) of a 14.6 mas displacement between the J2000
*           mean equinox and the origin of the ICRS frame.  It should,
*           however, be noted that neglecting this displacement when
*           calculating star coordinates does not lead to a 14.6 mas
*           change in right ascension, only a small second-order
*           distortion in the pattern of the precession-nutation effect.
*
*     For these reasons, the SOFA routines do not generate the "total
*     nutations" directly, though they can of course easily be generated
*     by calling iau_BI00, iau_PR00 and the present routine and adding
*     the results.
*
*  References:
*
*     Chapront, J., Chapront-Touze, M. & Francou, G., Astron.Astrophys.,
*     387, 700, 2002.
*
*     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 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., A&A282, 663-683 (1994).
*
*     Souchay, J., Loysel, B., Kinoshita, H., Folgueira, M., A&A Supp.
*     Ser. 135, 111 (1999)
*
*     Wallace, P.T., "Software for Implementing the IAU 2000
*     Resolutions", in IERS Workshop 5.1, 2002.
*
*-
```

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

```
*     suitable for most practical applications.
*
*     The routine delivers a pole accurate to 1 mas from 1900 to 2100
*     (usually better than 1 mas, very occasionally just outside 1 mas).
*     The full IAU 2000A model, which is implemented in the routine
*     iau_NUT00A (qv), delivers considerably greater accuracy at
*     current epochs;  however, to realize this improved accuracy,
*     corrections for the essentially unpredictable free-core-nutation
*     must also be included.
*
*  5) The present routine provides classical nutation.  The
*     MHB_2000_SHORT algorithm, from which it is adapted, deals also
*     with (i) the offsets between the GCRS and mean poles and (ii) the
*     adjustments in longitude and obliquity due to the changed
*     precession rates.  These additional functions, namely frame bias
*     and precession adjustments, are supported by the SOFA routines
*     iau_BI00 and iau_PR00.
*
*  6) The MHB_2000_SHORT algorithm also provides "total" nutations,
*     comprising the arithmetic sum of the frame bias, precession
*     adjustments, and nutation (luni-solar + planetary).  These total
*     nutations can be used in combination with an existing IAU 1976
*     precession implementation, such as iau_PMAT76, to deliver GCRS-to-
*     true predictions of mas accuracy at current epochs.  However, for
*     symmetry with the iau_NUT00A routine (qv for the reasons), the
*     SOFA routines do not generate the "total nutations" directly.  They
*     can of course easily be generated by calling iau_BI00, iau_PR00 and
*     the present routine and adding the results.
*
*  References:
*
*     Luzum, B., private communication, 2001 (Fortran code
*     MHB_2000_SHORT)
*
*     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.
*
*     McCarthy, D.D. & Luzum, B.J., "An Abridged Model of the Motion
*     of the Celestial Pole", preprint, 2001.
*
*     Simon, J.-L., Bretagnon, P., Chapront, J., Chapront-Touze, M.,
*     Francou, G., Laskar, J., A&A282, 663-683 (1994).
*
*-
```

```
      SUBROUTINE iau_NUT80 ( EPOCH1, EPOCH2, DPSI, DEPS )
*+
*  - - - - - - - - - -
*   i a u _ N U T 8 0
*  - - - - - - - - - -
*
*  Nutation, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     EPOCH1,EPOCH2   d      TDB starting epoch (Note 1)
*
*  Returned:
*     DPSI            d      nutation in longitude (radians)
*     DEPS            d      nutation in obliquity (radians)
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The nutation components are with respect to the ecliptic of
*     date.
*
*  Called:
*     iau_ANP     normalize radians to range -pi to +pi
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.222 (p111).
*
*-
```

```
      SUBROUTINE iau_NUTM80 ( EPOCH1, EPOCH2, RMATN )
*+
*  - - - - - - - - - - -
*   i a u _ N U T M 8 0
*  - - - - - - - - - - -
*
*  Form the matrix of nutation for a given date, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPOCH1,EPOCH2    d       TDB epoch (Note 1)
*
*  Returned:
*     RMATN          d(3,3)    nutation matrix
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(true) = RMATN * V(mean),
*     where the p-vector V(true) is with respect to the true
*     equatorial triad of date and the p-vector V(mean) is with
*     respect to the mean equatorial triad of date.
*
*  Called:
*     iau_NUT80   nutation, IAU 1980
*     iau_OBL80   mean obliquity, IAU 1980
*     iau_NUMAT   form nutation matrix
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_OBL80 ( EPOCH1, EPOCH2 )
*+
*  - - - - - - - - - -
*   i a u _ O B L 8 0
*  - - - - - - - - - -
*
*  Mean obliquity of the ecliptic, IAU 1980 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     EPOCH1,EPOCH2   d      TDB starting epoch (Note 1)
*
*  Returned:
*     iau_OBL80       d      obliquity of the ecliptic (radians, Note 2)
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The result is the angle between the ecliptic of J2000 and the mean
*     equator of date EPOCH1+EPOCH2.
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Expression 3.222-1 (p114).
*
*-
```

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

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

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

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

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

```
      SUBROUTINE iau_PLAN94 ( EPOCH1, EPOCH2, NP, PV, J )
*+
*  - - - - - - - - - - -
*   i a u _ P L A N 9 4
*  - - - - - - - - - - -
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Approximate heliocentric position and velocity of a nominated major
*  planet:  Mercury, Venus, EMB, Mars, Jupiter, Saturn, Uranus or
*  Neptune (but not Pluto nor the Earth itself).
*
*  Given:
*     EPOCH1   d       TDB epoch part A (Note 1)
*     EPOCH2   d       TDB epoch part B (Note 1)
*     NP       i       planet (1=Mercury, 2=Venus, 3=EMB ... 8=Neptune)
*
*  Returned:
*     PV       d(3,2)  planet pos,vel (heliocentric, J2000, AU, AU/d)
*     J        i       status: -1 = illegal NP (outside 1-8)
*                               0 = OK
*                              +1 = warning: date outside 1000-3000 AD
*                              +2 = warning: solution failed to converge
*
*  Notes
*
*  1) The epoch EPOCH1+EPOCH2 is in the TDB timescale 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:
*
*             EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     The limited accuracy of the present algorithm is such that any
*     of the methods is satisfactory.
*
*  2) If an NP value outside the range 1-8 is supplied, an error
*     status (J = -1) is returned and the PV vector set to zeroes.
*
*  3) For NP=3 the result is for the Earth-Moon Barycenter.  To
*     obtain the heliocentric position and velocity of the Earth,
*     use instead the SOFA routine iau_EPV00.
*
*  4) On successful return, the array PV contains the following:
*
*        PV(1,1)  x       }
*        PV(2,1)  y       } heliocentric position, AU
```

```
*        PV(3,1)  z       }
*
*        PV(1,2)  xdot    }
*        PV(2,2)  ydot    } heliocentric velocity, AU/d
*        PV(3,2)  zdot    }
*
*     The reference frame is equatorial and is with respect to the
*     mean equator and equinox of epoch J2000.
*
*  5) The algorithm is due to J.L. Simon, P. Bretagnon, J. Chapront,
*     M. Chapront-Touze, G. Francou and J. Laskar (Bureau des
*     Longitudes, Paris, France).  From comparisons with JPL
*     ephemeris DE102, they quote the following maximum errors
*     over the interval 1800-2050:
*
*                     L (arcsec)    B (arcsec)      R (km)
*
*        Mercury          4             1             300
*        Venus            5             1             800
*        EMB              6             1            1000
*        Mars            17             1            7700
*        Jupiter         71             5           76000
*        Saturn          81            13          267000
*        Uranus          86             7          712000
*        Neptune         11             1          253000
*
*     Over the interval 1000-3000, they report that the accuracy is no
*     worse than 1.5 times that over 1800-2050.  Outside 1000-3000 the
*     accuracy declines.
*
*     Comparisons of the present routine with the JPL DE200 ephemeris
*     give the following RMS errors over the interval 1960-2025:
*
*                      position (km)     velocity (m/s)
*
*        Mercury            334               0.437
*        Venus             1060               0.855
*        EMB               2010               0.815
*        Mars              7690               1.98
*        Jupiter          71700               7.70
*        Saturn          199000              19.4
*        Uranus          564000              16.4
*        Neptune         158000              14.4
*
*     Comparisons against DE200 over the interval 1800-2100 gave the
*     following maximum absolute differences.  (The results using
*     DE406 were essentially the same.)
*
*                   L (arcsec)   B (arcsec)     R (km)   Rdot (m/s)
*
*        Mercury        7            1            500       0.7
*        Venus          7            1           1100       0.9
*        EMB            9            1           1300       1.0
*        Mars          26            1           9000       2.5
*        Jupiter       78            6          82000       8.2
*        Saturn        87           14         263000      24.6
*        Uranus        86            7         661000      27.4
*        Neptune       11            2         248000      21.4
*
*  6) The present SOFA re-implementation of the original Simon et al.
*     Fortran code differs from the original in the following respects:
*
*       *  The date is supplied in two parts.
```

```
*
*       *  The result is returned only in equatorial Cartesian form;
*          the ecliptic longitude, latitude and radius vector are not
*          returned.
*
*       *  The result is in the J2000 equatorial frame, not ecliptic.
*
*       *  More is done in-line:  there are fewer calls to other
*          routines.
*
*       *  Different error/warning status values are used.
*
*       *  A different Kepler's-equation-solver is used (avoiding
*          use of COMPLEX*16).
*
*       *  Polynomials in T are nested to minimize rounding errors.
*
*       *  Explicit double-precision constants are used to avoid
*          mixed-mode expressions.
*
*       *  There are other, cosmetic, changes to comply with SOFA
*          style conventions.
*
*     None of the above changes affects the result significantly.
*
*  7) The returned status, J, indicates the most serious condition
*     encountered during execution of the routine.  Illegal NP is
*     considered the most serious, overriding failure to converge,
*     which in turn takes precedence over the remote epoch warning.
*
*  Called:
*     iau_ANP     normalize radians to range -pi to +pi
*
*  Reference:  Simon, J.L, Bretagnon, P., Chapront, J.,
*              Chapront-Touze, M., Francou, G., and Laskar, J.,
*              Astron. Astrophys. 282, 663 (1994).
*
*-
```

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

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

```
      SUBROUTINE iau_PMAT76 ( EPOCH1, EPOCH2, RMATP )
*+
*  - - - - - - - - - - -
*   i a u _ P M A T 7 6
*  - - - - - - - - - - -
*
*  Precession matrix from J2000 to a specified date, IAU 1976 model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPOCH1,EPOCH2  d       ending epoch, TDB (Note 1)
*
*  Returned:
*     RMATP          d(3,3)  precession matrix, J2000 -> EPOCH1+EPOCH2
*
*  Notes:
*
*  1) The ending epoch EPOCH1+EPOCH2 is a Julian Date, apportioned
*     in any convenient way between the arguments EPOCH1 and EPOCH2.
*     For example, JD(TDB)=2450123.7 could be expressed in any of
*     these ways, among others:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RMATP * V(J2000),
*     where the p-vector V(J2000) is with respect to the mean
*     equatorial triad of epoch J2000 and the p-vector V(date)
*     is with respect to the mean equatorial triad of the given
*     epoch.
*
*  3) Though the matrix method itself is rigorous, the precession
*     angles are expressed through canonical polynomials which are
*     valid only for a limited time span.  In addition, the IAU 1976
*     precession rate is known to be imperfect.  The absolute accuracy
*     of the present formulation is better than 0.1 arcsec from
*     1960AD to 2040AD, better than 1 arcsec from 1640AD to 2360AD,
*     and remains below 3 arcsec for the whole of the period
*     500BC to 3000AD.  The errors exceed 10 arcsec outside the
*     range 1200BC to 3900AD, exceed 100 arcsec outside 4200BC to
*     5600AD and exceed 1000 arcsec outside 6800BC to 8200AD.
*
*  Called:
*     iau_PREC76  accumulated precession angles, IAU 1976
*     iau_IR      initialize r-matrix to identity
*     iau_RZ      rotate around Z-axis
```

```
*     iau_RY      rotate around Y-axis
*     iau_CR      copy r-matrix
*
*  References:
*
*     Lieske,J.H., 1979. Astron.Astrophys.,73,282.
*      equations (6) & (7), p283.
*
*     Kaplan,G.H., 1981. USNO circular no. 163, pA2.
*
*-
```

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

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

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

```
*  5) The matrix RP transforms vectors from mean J2000 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.
*
*  Called:
*     iau_PR00     IAU 2000 precession adjustments
*     iau_OBL80    mean obliquity, IAU 1980
*     iau_BP00     frame bias and precession matrices
*     iau_NUMAT    form nutation matrix
*     iau_RXR      r-matrix product
*
*  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", submitted to A&A (2002)
*
*-
```

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

```
*
*  5) The matrix RP transforms vectors from mean J2000 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 2000A Celestial Intermediate Pole
*     are elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_NUT00A   nutation, IAU 2000A
*     iau_PN00     bias/precession/nutation results
*
*  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", submitted to A&A (2002)
*
*-
```

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

```
*
*  5) The matrix RP transforms vectors from mean J2000 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 2000B Celestial Intermediate Pole
*     are elements (3,1-3) of the matrix RBPN.
*
*  Called:
*     iau_NUT00B   nutation, IAU 2000B
*     iau_PN00     bias/precession/nutation results
*
*  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", submitted to A&A (2002)
*
*-
```

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

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

```
      SUBROUTINE iau_PNM80 ( EPOCH1, EPOCH2, RMATPN )
*+
*  - - - - - - - - - -
*   i a u _ P N M 8 0
*  - - - - - - - - - -
*
*  Form the matrix of precession/nutation for a given date, IAU 1976
*  precession model, IAU 1980 nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     EPOCH1,EPOCH2   d         TDB epoch (Note 1)
*
*  Returned:
*     RMATPN          d(3,3)    combined precession/nutation matrix
*
*  Notes:
*
*  1) The epoch EPOCH1+EPOCH2 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:
*
*            EPOCH1        EPOCH2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The matrix operates in the sense V(date) = RMATPN * V(J2000),
*     where the p-vector V(date) is with respect to the true
*     equatorial triad of epoch EPOCH1+EPOCH2 and the p-vector
*     V(J2000) is with respect to the mean equatorial triad of
*     epoch J2000.
*
*  Called:
*     iau_PMAT76  precession matrix, IAU 1976
*     iau_NUTM80  nutation matrix, IAU 1980
*     iau_RXR     r-matrix multiply
*
*  Reference:
*
*     Explanatory Supplement to the Astronomical Almanac,
*     P. Kenneth Seidelmann (ed), University Science Books (1992),
*     Section 3.3 (p145).
*
*-
```

```
      SUBROUTINE iau_POM00 ( XP, YP, SP, RPOM )
*+
*  - - - - - - - - - - -
*   i a u _ P O M 0 0
*  - - - - - - - - - - -
*
*  Form the matrix of polar motion for a given date, IAU 2000.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     XP,YP      d      coordinates of the pole (radians, Note 1)
*     SP         d      the quantity s' (radians, Note 2)
*
*  Returned:
*     RPOM     d(3,3)   polar-motion matrix (Note 3)
*
*  Notes:
*
*  1) XP and YP are the "coordinates of the pole", in radians, which
*     position the Celestial Intermediate Pole in the International
*     Terrestrial Reference System (see IERS Conventions 2000).  In a
*     geocentric right-handed triad u,v,w, where the w-axis points at
*     the north geographic pole, the v-axis points towards the origin
*     of longitudes and the u axis completes the system, XP = +u and
*     YP = -v.
*
*  2) SP is the quantity s', in radians, which positions the Terrestrial
*     Ephemeris 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.
*     The routine iau_SP00 implements this approximation.
*
*  3) The matrix operates in the sense V(TRS) = RPOM * V(CIP), meaning
*     that it is the final rotation when computing the pointing
*     direction to a celestial source.
*
*  Called:
*     iau_IR       initialize r-matrix to identity
*     iau_RZ       rotate around Z-axis
*     iau_RY       rotate around Y-axis
*     iau_RX       rotate around X-axis
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

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

```
      SUBROUTINE iau_PR00 ( DATE1, DATE2, DPSIPR, DEPSPR )
*+
*  - - - - - - - - -
*   i a u _ P R 0 0
*  - - - - - - - - -
*
*  Precession-rate part of the IAU 2000 precession-nutation models
*  (part of MHB_2000).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d   TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     DPSIPR,DEPSPR  d   precession corrections (Notes 2,3)
*
*  Notes
*
*  1) The T date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The precession adjustments are expressed as "nutation components",
*     corrections in longitude and obliquity with respect to the J2000
*     equinox and ecliptic.
*
*  3) Although the precession adjustments are stated to be with respect
*     to Lieske et al. (1977), the MHB_2000 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
*     (Wallace 2002).
*
*  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 MHB_2000 code itself was obtained on 9th September 2002
*     from ftp //maia.usno.navy.mil/conv2000/chapter5/IAU2000A.
*
*     Wallace, P.T., "Software for Implementing the IAU 2000
*     Resolutions", in IERS Workshop 5.1, 2002.
*
*-
```

```
      SUBROUTINE iau_PREC76 ( EP01, EP02, EP11, EP12, ZETA, Z, THETA )
*+
*  - - - - - - - - - - -
*   i a u _ P R E C 7 6
*  - - - - - - - - - - -
*
*  IAU 1976 precession model.
*
*  This routine forms the three Euler angles which implement general
*  precession between two epochs, using the IAU 1976 model (as for
*  the FK5 catalog).
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     EP01,EP02   d      TDB starting epoch (Note 1)
*     EP11,EP12   d      TDB ending epoch (Note 1)
*
*  Returned:
*     ZETA        d      1st rotation: radians clockwise around z
*     Z           d      3rd rotation: radians clockwise around z
*     THETA       d      2nd rotation: radians counterclockwise around y
*
*  Notes:
*
*  1) The 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
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in cases
*     where the loss of several decimal digits of resolution is
*     acceptable.  The J2000 method is best matched to the way the
*     argument is handled internally and will deliver the optimum
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*     The two 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 1000 arcsec outside 6800BC to
*     8200AD.
*
*  3) The three angles are returned in the conventional order, which
```

```
*     is not the same as the order of the corresponding Euler rotations.
*     The precession matrix is R_3(-z) x R_2(+theta) x R_3(-zeta).
*
*  Reference:
*
*     Lieske,J.H., 1979. Astron.Astrophys.,73,282.
*      equations (6) & (7), p283.
*
*-
```

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

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

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

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

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

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

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

```
*
*     (ii) The transformation complies with special relativity.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds; the radial velocity is in km/s, but
*     the pv-vector result is in AU and AU/day.
*
*  4) The proper motions are the rate of change of the right ascension
*     and declination at the catalog epoch and are in radians per Julian
*     year.  The RA proper motion is in terms of coordinate angle, not
*     true angle, and will thus be numerically larger at high
*     declinations.
*
*  5) Straight-line motion at constant speed in the inertial frame is
*     assumed.  If the speed is greater than or equal to the speed of
*     light, the routine aborts with an error status.
*
*  6) The inverse transformation is performed by the routine iau_STARPV.
*
*  Called:
*     iau_PN       normalize p-vector returning modulus
*     iau_PDP      scalar product
*     iau_SXP      multiply p-vector by scalar
*     iau_PMP      p-vector minus p-vector
*     iau_PM       modulus of p-vector
*     iau_PPP      p-vector plus p-vector
*     iau_PV2S     pv-vector to spherical coordinates
*     iau_ANP      normalize radians to range 0 to 2pi
*
*  Reference:
*
*     Stumpff, P., Astron.Astrophys. 144, 232-240 (1985).
*
*-
```

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

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

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

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

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

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

```
      SUBROUTINE iau_RX ( PHI, R )
*+
*  - - - - - - -
*   i a u _ R X
*  - - - - - - -
*
*  Rotate an r-matrix about the x-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PHI      d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix
*
*  Sign convention:  The matrix can be used to rotate the
*  reference frame of a vector.  Calling this routine 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:
*     iau_IR       initialize r-matrix to identity
*     iau_RXR      r-matrix multiply
*     iau_CR       r-matrix copy
*
*-
```

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

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

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

```
      SUBROUTINE iau_RY ( THETA, R )
*+
*  - - - - - - -
*   i a u _ R Y
*  - - - - - - -
*
*  Rotate an r-matrix about the y-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     THETA    d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix
*
*  Sign convention:  The matrix can be used to rotate the
*  reference frame of a vector.  Calling this routine 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:
*     iau_IR       initialize r-matrix to identity
*     iau_RXR      r-matrix multiply
*     iau_CR       r-matrix copy
*
*-
```

```
      SUBROUTINE iau_RZ ( PSI, R )
*+
*  - - - - - - -
*   i a u _ R Z
*  - - - - - - -
*
*  Rotate an r-matrix about the z-axis.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  vector/matrix support routine.
*
*  Given:
*     PSI      d         angle (radians)
*
*  Given and returned:
*     R        d(3,3)    r-matrix, rotated
*
*  Sign convention:  The matrix can be used to rotate the
*  reference frame of a vector.  Calling this routine 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:
*     iau_IR       initialize r-matrix to identity
*     iau_RXR      r-matrix multiply
*     iau_CR       r-matrix copy
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00 ( DATE1, DATE2, X, Y )
*+
*  - - - - - - - -
*   i a u _ S 0 0
*  - - - - - - - -
*
*  The quantity s, positioning the Celestial Ephemeris Origin on the
*  equator of the Celestial Intermediate Pole, given the CIP's X,Y
*  coordinates.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*     X,Y            d      CIP coordinates (Note 3)
*
*  Returned:
*     iau_S00        d      the quantity s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The quantity s is the difference between the right ascensions
*     of the same point in two frames.  The two systems are the GCRS
*     and the CIP,CEO, and the point is the ascending node of the
*     respective equators.  The quantity 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.  This routine
*     requires X,Y to be supplied by the caller, who is responsible for
*     providing values that are consistent with the supplied date.
*
*  Called:
*     iau_ANPM      normalize angle into range +/- pi
*
*  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", submitted to A&A (2002)
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00A ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S 0 0 A
*  - - - - - - - - -
*
*  The quantity s, positioning the Celestial Ephemeris Origin on the
*  equator of the Celestial Intermediate Pole, using the IAU 2000A
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_S00A       d      the quantity s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The quantity s is the difference between the right ascensions
*     of the same point in two frames.  The two systems are the GCRS
*     and the CIP,CEO, and the point is the ascending node of the
*     respective equators.  The quantity 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
*     uses the full IAU 2000A nutation model when predicting the CIP
*     position.  Faster results, with no significant loss of accuracy.
*     can be obtained via the routine iau_s00B, which uses instead the
*     IAU 2000B truncated model.
*
*  Called:
*     iau_PNM00A    bias-precession-nutation matrix, IAU 2000A
*     iau_BNP2XY    extract CIP X,Y from the BPN matrix
*     iau_S00       the quantity s, given X,Y
*
```

```
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", submitted to A&A (2002)
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

```
      DOUBLE PRECISION FUNCTION iau_S00B ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S 0 0 B
*  - - - - - - - - -
*
*  The quantity s, positioning the Celestial Ephemeris Origin on the
*  equator of the Celestial Intermediate Pole, using the IAU 2000B
*  precession-nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_S00B       d      the quantity s in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The quantity s is the difference between the right ascensions
*     of the same point in two frames.  The two systems are the GCRS
*     and the CIP,CEO, and the point is the ascending node of the
*     respective equators.  The quantity 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
*     uses the full IAU 2000A nutation model when predicting the CIP
*     position.  Faster results, with no significant loss of accuracy.
*     can be obtained via the routine iau_s00B, which uses instead the
*     IAU 2000B truncated model.
*
*  Called:
*     iau_PNM00B    bias-precession-nutation matrix, IAU 2000B
*     iau_BNP2XY    extract CIP X,Y from the BPN matrix
*     iau_S00       the quantity s, given X,Y
*
```

```
*  References:
*
*     Capitaine, N., Chapront, J., Lambert, S. and Wallace, P.,
*     "Expressions for the Celestial Intermediate Pole and Celestial
*     Ephemeris Origin consistent with the IAU 2000A precession-nutation
*     model", submitted to A&A (2002)
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

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

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

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

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

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

```
      DOUBLE PRECISION FUNCTION iau_SP00 ( DATE1, DATE2 )
*+
*  - - - - - - - - -
*   i a u _ S P 0 0
*  - - - - - - - - -
*
*  The quantity s', positioning the Terrestrial Ephemeris Origin on the
*  equator of the Celestial Intermediate Pole.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  canonical model.
*
*  Given:
*     DATE1,DATE2    d      TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     iau_SP00       d      the quantity s' in radians (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The quantity s' is obtained from polar motion observations by
*     numerical integration, and so is in essence unpredictable.
*     However, it is dominated by a secular drift of about
*     47 microarcseconds per century, which is the approximation
*     evaluated by the present routine.
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
*-
```

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

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

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

```
*
*     (i) In stars with significant radial velocity and proper motion,
*     the constantly changing light-time distorts the apparent proper
*     motion.  Note that this is a classical, not a relativistic,
*     effect.
*
*     (ii) The transformation complies with special relativity.
*
*  3) Care is needed with units.  The star coordinates are in radians
*     and the proper motions in radians per Julian year, but the
*     parallax is in arcseconds; the radial velocity is in km/s, but
*     the pv-vector result is in AU and AU/day.
*
*  4) The RA proper motion is in terms of coordinate angle, not true
*     angle.  If the catalog uses arcseconds for both RA and Dec proper
*     motions, the RA proper motion will need to be divided by cos(Dec)
*     before use.
*
*  5) Straight-line motion at constant speed, in the inertial frame,
*     is assumed.
*
*  6) An extremely small (or zero or negative) parallax is interpreted
*     to mean that the object is on the "celestial sphere", the radius
*     of which is an arbitrary (large) value (see the constant PXMIN).
*     When the distance is overridden in this way, the status, initially
*     zero, has 1 added to it.
*
*  7) If the space velocity is a significant fraction of c (see the
*     constant VMAX), it is arbitrarily set to zero.  When this action
*     occurs, 2 is added to the status.
*
*  8) The relativistic adjustment involves an iterative calculation.
*     If the process fails to converge within a set number (IMAX) of
*     iterations, 4 is added to the status.
*
*  9) The inverse transformation is performed by the routine iau_PVSTAR.
*
*  Called:
*     iau_S2PV     spherical coordinates to pv-vector
*     iau_PM       modulus of p-vector
*     iau_ZP       zero a p-vector
*     iau_PN       normalize p-vector returning modulus
*     iau_PDP      dot product of two p-vectors
*     iau_SXP      multiply p-vector by scalar
*     iau_PMP      p-vector minus p-vector
*     iau_PPP      p-vector plus p-vector
*
*  Reference:
*
*     Stumpff, P., Astron.Astrophys. 144, 232-240 (1985).
*
*-
```

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

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

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

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

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

```
      SUBROUTINE iau_XYS00A ( DATE1, DATE2, X, Y, S )
*+
*  - - - - - - - - - - -
*   i a u _ X Y S 0 0 A
*  - - - - - - - - - - -
*
*  For a given TT date, compute the X,Y coordinates of the Celestial
*  Intermediate Pole and the quantity s, using the IAU 2000A precession-
*  nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y           d    Celestial Intermediate Pole (Note 2)
*     S             d    the quantity s (Note 2)
*
*  Notes:
**
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The quantity s (in radians) positions the Celestial Ephemeris
*     Origin on the equator of the CIP.
*
*  4) A faster, but slightly less accurate result (about 1 mas for X,Y),
*     can be obtained by using instead the iau_XYS00B routine.
*
*  Called:
*     iau_PNM00A   bias-precession-nutation matrix, IAU 2000A
*     iau_BPN2XY   extract CIP coordinates from b-p-n matrix
*     iau_S00      the quantity s, given X,Y
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
```

* –

```
      SUBROUTINE iau_XYS00B ( DATE1, DATE2, X, Y, S )
*+
*  - - - - - - - - - - -
*   i a u _ X Y S 0 0 B
*  - - - - - - - - - - -
*
*  For a given TT date, compute the X,Y coordinates of the Celestial
*  Intermediate Pole and the quantity s, using the IAU 2000B precession-
*  nutation model.
*
*  This routine is part of the International Astronomical Union's
*  SOFA (Standards of Fundamental Astronomy) software collection.
*
*  Status:  support routine.
*
*  Given:
*     DATE1,DATE2   d    TT as a 2-part Julian Date (Note 1)
*
*  Returned:
*     X,Y           d    Celestial Intermediate Pole (Note 2)
*     S             d    the quantity s (Note 2)
*
*  Notes:
*
*  1) The TT date DATE1+DATE2 is a Julian Date, apportioned in any
*     convenient way between the two arguments.  For example,
*     JD(TT)=2450123.7 could be expressed in any of these ways,
*     among others:
*
*            DATE1          DATE2
*
*         2450123.7D0        0D0        (JD method)
*          2451545D0      -1421.3D0     (J2000 method)
*         2400000.5D0     50123.2D0     (MJD method)
*         2450123.5D0       0.2D0       (date & time method)
*
*     The JD method is the most natural and convenient to use in
*     cases where the loss of several decimal digits of resolution
*     is acceptable.  The J2000 method is best matched to the way
*     the argument is handled internally and will deliver the
*     optimum resolution.  The MJD method and the date & time methods
*     are both good compromises between resolution and convenience.
*
*  2) The Celestial Intermediate Pole coordinates are the x,y components
*     of the unit vector in the Geocentric Celestial Reference System.
*
*  3) The quantity s (in radians) positions the Celestial Ephemeris
*     Origin on the equator of the CIP.
*
*  4) The present routine is faster, but slightly less accurate (about
*     1 mas in X,Y), than the iau_XYS00A routine.
*
*  Called:
*     iau_PNM00B   bias-precession-nutation matrix, IAU 2000B
*     iau_BPN2XY   extract CIP coordinates from b-p-n matrix
*     iau_S00      the quantity s, given X,Y
*
*  Reference:
*
*     McCarthy, D.D., IERS Conventions 2000, Chapter 5 (2002).
*
```

*–

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

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

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

COPYRIGHT NOTICE

Text like the following appears at the end of every SOFA routine.

```
*  8. The user shall not cause the Software to be brought into
*     disrepute, either by misuse, or use for inappropriate tasks, or by
*     inappropriate modification.
*
*  9. The Software is provided to the user "as is" and the Board makes
*     no warranty as to its use or performance.   The Board does not and
*     cannot warrant the performance or results which the user may
*     obtain by using the Software.  The Board 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 the Board be liable to the user for any consequential,
*     incidental, or special damages, including any lost profits or lost
*     savings, even if a Board representative has been advised of such
*     damages, or for any claim by any third party.
*
*  Correspondence concerning SOFA software should be addressed as
*  follows:
*
*     Internet email: sofa@rl.ac.uk
*     Postal address: IAU SOFA Center
*                     Rutherford Appleton Laboratory
*                     Chilton, Didcot, Oxon OX11 0QX
*                     United Kingdom
*
*
*-----------------------------------------------------------------------

      END
```

## 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 = 206264.8062470963551564733D0 )

*  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 )
```

board.lis 2001 March 2

IAU STANDARDS OF FUNDAMENTAL ASTRONOMY REVIEW BOARD

Current Membership

| | | |
|---|---|---|
| John Bangert | United States Naval Observatory | |
| Wim Brouw | Australia Telescope National Facility | |
| Anne-Marie Gontier | Paris Observatory | |
| Catherine Hohenkerk | Rutherford Appleton Laboratory | |
| Wen-Jing Jin | Shanghai Observatory | |
| Zinovy Malkin | Inst. Appl. Astron., St Petersburg | |
| Dennis McCarthy | United States Naval Observatory | |
| Jeffrey Percival | University of Wisconsin | |
| Patrick Wallace | Rutherford Appleton Laboratory | (chair) |

Past Members

| | |
|---|---|
| Skip Newhall | Jet Propulsion Laboratory |
| George Kaplan | United States Naval Observatory |

The e-mail address for the Board chair is ptw@star.rl.ac.uk