NATIONAL RADIO ASTRONOMY OBSERVATORY
COMPUTER DIVISION
INTERNAL REPORT

A SYSTEM OF FORTRAN SUBPROGRAMS FOR SPHERICAL ASTRONOMY

BY

NEAL STOLTZFUS* and PETER STUMPFF

Report No. 4
January 1968

*1967 Summer Student at NRAO
Present address: Princeton University

In order to make that part of the astronomical data reduction which deals with the time dependent effects of spherical astronomy as easy as possible, we developed a system of FORTRAN subprograms solving all of these problems in a very general way. These programs will reside permanently on disk. They can be called from any main program in the same way as SIN. Table 1 contains a compilation of the names and of the purposes of the programs. If two names appear for the same purpose they indicate two versions of different accuracy. In most cases, the more accurate version has a name starting with the character "D"; the arguments, in these cases, have to be defined as double precision variables. If the name of a FUNCTION starts with a D, this name must be defined as double precision, also. For the computation of nutation and of the Besselian Day Numbers, the arguments of both versions (NUT1, NUT2, BDN1, BDN2) are in single precision but are more precisely computed in the versions NUT2, BDN2.

The theory of most of the routines had been described in (I). The basic concept of (I) was: 1) To use a COMMON set of constants for all programs in order to save memory space. 2) To truncate the theoretical expressions for the time dependent effects after the first order term in order to make the programs as simple, as short, and as quick as possible. 3) To update the values of the common constants once every year in order to make the programs applicable to any instant of time in spite of the truncation of the theoretical equations used inside the programs.

Discussions within the Computer Division have shown, however, that this concept would not be suitable for two reasons: 1) It would confuse the user if he had to define COMMON'S which are not related to his own program. 2) Practical difficulties would occur with the proper updating of the constants. We decided, therefore, to provide every program with its own set of constants and to allow for the application of the programs for any instant of time within the 20th century. The low accuracy versions of the programs (e.g., SIT, ZØT, NUT1, BDN1) are constructed with the methods described in (I). The internal initial epoch for these programs is the beginning of the year 1967. If one applies these programs to dates not too far from that epoch (a couple of years), they will have the accuracy mentioned in (I). Of course, one can apply them also to dates which are further away from 1967, but then the accuracy will decrease. There will be many radio-astronomical applications where the accuracy need is so low that these versions could be applied for any time within the 20th century with sufficient precision.

Of course, sometimes there will be a need for greater or full accuracy - for instance, in all interferometer applications. To fulfill such requests, we have added high accuracy versions of

Table 1

Compilation of the Names and of the Purposes of the Routines

I. TIME AND CALENDAR PROBLEMS

Name	Given	Results
SIT, DSIT	Calendar Date, Zone Time, Longitude	Local Mean Sidereal Time, Julian Date
ZØT, DZØT	Calendar Date, Local Mean Sidereal Time, Longitude	Zone Time, Julian Date
DJL	Calendar Date at Greenwich	Julian Date of 0^h U.T.
CLD	Julian Date	Calendar Date at Greenwich
EQT	Julian Date	Equation of Time, RA and DEC of the Sun

II. APPARENT PLACES

Name	Given	Results
NUT1, NUT2	Julian Date	Nutation in Longitude and in Obliquity
BSC	Mean Equinox, RA, DEC	Besselian Star Constants
BDN1, BDN2	Julian Date, Mean Equinox	Besselian Day Numbers

III. ORTHOGONAL COORDINATE TRANSFORMATION

Name	Given	Results
TRC, DTRC	Transformation Matrix, Spherical Coordinates in the old System	Spherical Coordinates in the new System
PRE, DPRE	Mean Equinoxes T_1, T_2	Matrix for the general Precession from T_1 to T_2
CUV, DCUV	Spherical Coordinates	Unit Vector
UVC, DUVC	Unit Vector	Spherical Coordinates

IV. AUXILIARY PROGRAMS

Name	Given	Results
RED, DRED	Arbitrary Value of an Angle in one of four possible standard units	Value of the angle after reduction to one of three possible standard intervals
A13, DA13	Julian Date	Longitudes A1, A2, A3 connected with the mean motion of the Sun
A46, DA46	Julian Date	Longitudes A4, A5, A6 connected with the mean motion of the Moon
SUL, DSUL	Julian Date	True Longitude of the Sun
EPS, DEPS	Julian Date	Obliquity of the Ecliptic
ECC, DECC	Julian Date	Eccentricity of the orbit of the Earth
ECE, DECE	ECC and A3 (or DECC and DA3)	Equation of the center for the orbit of the Earth

most of the programs. Usually, these are not only formally in double precision but use the definitions of fundamental astronomy with their full accuracy, i.e., without the truncations mentioned above. For instance, the time conversion routines DSIT and DZØT can be applied to any instant of time and will yield results with the same precision as those obtained from the annual volumes of the American Ephemeris. The nutation is calculated (in NUT2) with full precision, i.e., with all known theoretical terms. In the case of the Besselian Day Numbers, however, the routine BDN2, although more accurate than BDN1, will not give the accuracy of these numbers as published in the American Ephemeris. The reason for this is that aberration can only be approximated neglecting the periodic perturbations in the motion of the earth and neglecting the transition from the heliocentrum to the barycentrum. In principle, of course, one could develop much better approximations for this, but that would increase the length of the program considerably. Since the Besselian Day Numbers A, B, and E will have the full accuracy of the American Ephemeris, and since C and D will not have errors larger than about 0"02, we decided not to try any further improvements.

In addition to those programs already mentioned, a rather large number of other routines have been added to the system. Part of these will have advantages for the direct explicit use in any main program, such as the calendar routine CLD* and the julian date routine DJL*; or the coordinate transformation routines TRC, DTRC, CUV, DCUV, UVC, DUVC; and the two routines which generate the matrix of general precession PRE and DPRE. All the remaining routines have a more auxiliary character in that they are used internally by many of the other programs. For example, A13, DA13, A46, DA46 generate the basic longitudes A1, A2, A3, A4, A5, A6 connected with the mean motion of the sun and the moon (see (I), p.25 ff). They are used directly in the two nutation routines (NUT1 and NUT2). They are also used by the routines SUL and DSUL which compute approximate values for the ecliptic longitude of the sun - neglecting only the periodic perturbations and the transition from heliocentrum to barycentrum. SUL and DSUL are directly used for the computation of the aberrational day numbers C and D. Later we will add routines which compute the reduction from observed to heliocentric radial velocities which also will use SUL and DSUL. If one can generate approximate values for the ecliptic longitude of the sun, approximate values for right ascension and declination of the sun can easily be obtained. This is done by the FUNCTION EQT which calculates the sun's coordinates for any time with an accuracy of about 30" or better. The value of the function itself is the so-called "equation of time", which might be useful sometime.

As far as general precession is concerned, we have adopted

*The authors of these two programs are Johann Schraml and Barry Clark, respectively.

the following procedure. The precession transformation consists of a time dependent part and of a position dependent part The time dependent part is given by the matrix of general precession which depends on the two mean equinoxes only. Essentially, the position dependent part is the multiplication of an initial unit vector with the precession matrix. If one has to precess many sources from one given equinox T_1 to a second equinox T_2, the same matrix can be used for all sources. It saves computer time, therefore, to calculate the matrix only once. This is the justification for having the programs PRE and DPRE. The position dependent part is completely general; in other words, the corresponding program can be applied to any type of coordinate transformation such as the conversion from RA, DEC to ℓ, b; or from HA, DEC to Azimuth, Elevation; or the opposite conversions. Which conversion is performed depends completely on the matrix elements. Equations and numerical values of matrix elements for some such conversions are compiled in the description of the routine TRC (=NRAO 13/1S) in the NRAO program library. The conversion consists of the following steps: Compute the unit vector which corresponds to the given spherical coordinates (CUV, DCUV); multiply that vector with the matrix - whatever this matrix may mean - to obtain the unit vector in the new system; compute the spherical coordinates in the new system from the unit vector (UVC, DUVC). All these steps could be done explicitly in a main program using two of these four routines (CUV, DCUV, UVC, DUVC). This might have advantages if one source has to be precessed to many different equinoxes, in which case one would compute the initial unit vector only once by calling CUV or DCUV. In most cases, however, one wishes to do the transformation in one step which can be realized by calling TRC or DTRC (which uses internally the coordinate to unit vector conversion routines).

The same split into time dependent and position dependent steps holds for the reduction from mean to apparent places. The reduction consists of adding products of Besselian Day Numbers and star constants. This simple step can be left to the main program. The star constants should be computed by calling a subprogram (BSC), and the day numbers should be computed by calling a subprogram (BDN1, BDN2). For each date of observation one has to compute the day numbers only once, and then compute the reduction for every source. If there are many dates within one day, one should call the programs BDN1 or BDN2 only for the beginning and the end of the observation period, and then interpolate linearly. In particular, BDN2 is a very slow routine (61 ms) because it computes all nutation terms.

Among the auxiliary programs are the two routines, RED and DRED, which reduce a given angle to one of three possible standard intervals: $0 \rightarrow 2\pi$, $-\pi \rightarrow +\pi$, $-\pi/2 \rightarrow +\pi/2$. The unit in which the angle is given can be either revolutions (REV), radians (RAD), hours (H), or degrees (DEG). The result will be in the same unit. This is a very helpful routine; internally, it is used in many of the other programs. Of course, it is also very helpful for external purposes, for example, to answer the following type of question: Two sources are given whose right ascensions are RA1 and RA2; both of them are expressed in RAD and are reduced to $0 \rightarrow 2\pi$. Is source I east or west of source II? Answer: If RED(RA1-RA2,2,2) is positive, source I is east of source II. Essentially, these routines correspond to the "reduction operator" which was used in (I) as a very helpful tool for handling all types of time conversion problems.

The rest of the auxiliary programs (EPS,DEPS,ECC,DECC,ECE, DECE) compute values of the obliquity of the ecliptic, of the eccentricity of the orbit of the earth, and of the so-called equation of the center. They are called frequently inside the other programs. Except for the case of the obliquity of the ecliptic which is frequently used in transformation problems, there will not be much external use of these routines.

Those effects of spherical astronomy which depend on absolute time (for example, as nutation) need a time argument. The easiest way to provide such an argument is to compute the julian date, DATJUL, which corresponds to the given instant. Since one usually starts with a calendar date and a local sidereal or zone time, one can use one of the first four routines (SIT,DSIT,ZØT,DZØT) in order to get DATJUL. Later, DATJUL can be used as the basic argument for many of the other routines.

Altogether, there are 34 FORTRAN subprograms which solve some of the everyday problems of data reduction. They are, at the same time, the first 34 programs of the NRAO PROGRAM LIBRARY. In the description of the library (see (II)) it was stated that we would add the program descriptions of all of these astronomical routines to the present report in the form of an appendix. However, after more realistic thought on this promise, we found that Xerox copies of these complete descriptions would require an effort which probably cannot be justified. Therefore, we will leave it up to the users whether or not they want their own copy of the program description. Instead, we will list (see Table 2) the calling sequences of all the routines. The column on the right limb of the table ("Accuracy") gives an upper limit for the error of the results within the 20th century. If there is more than one result, the error is concerned with the most inaccurate one. In most cases the actual errors will be smaller, especially within the next years. The names of the arguments have been chosen differently from the actual names in the programs (and in their descriptions):

Names starting with a "D" are	REAL*8
Names starting with one of the characters "I", "K", "L" are	INTEGER*4
All other names are	REAL*4

The underlined names represent output variables. If an argument is an array, the dimension of the array is put after the name. For example, BDN1(...,<u>BDNS</u>(5),...) in Table 2 means that in a main program one has to CALL BDN1(...,BDNS,...) and that this actually fills the 5 elements of the array BDNS.

Table 2

Calling Sequence and Accuracy of the Routines

		Calling Sequence	Accuracy
I.	SIT	(IY, IM, ID, ZTIM, ØBL, ZTL, STIM, DATJUL)	$4^{s}.5$
	DSIT	(IY, IM, ID, DZTIM, DØBL, DZTL, DSTIM, DATJUL)	full
	ZØT	(IY, IM, ID, STIM, IAMB, ØBL, ZTL, ZTIM, DATJUL)	$4^{s}.5$
	DZØT	(IY, IM, ID, DSTIM, IAMB, DØBL, DZTL, DZTIM, DATJUL)	full
	DJL	(IY, IM, ID)	full
	CLD	(DATJUL, IY, IM, ID, IWD)	full
	EQT	(DATJUL, SUNRA, SUNDEC)	1'
II.	NUT1	(DATJUL, PSIDEL, EPSDEL)	$0''.1$
	NUT2	(DATJUL, PSIDEL, EPSDEL, SPSI, SEPS)	full
	BSC	(EQ, EQALPH, EQDELT, SCRA(4), SCDEC(4))	10^{-6}
	BDN1	(DATJUL, EQ, BDNS(5), PSIDEL)	$0''.05$
	BDN2	(DATJUL, EQ, BDNS(5), PSIDEL, SPSI, SEPS)	$0''.02$
III.	TRC	(MAT(3,3), ALPH1, BETA1, ALPH2, BETA2)	1"
	DTRC	(DMAT(3,3), DALPH1, DBETA1, DALPH2, DBETA2)	full
	PRE	(EQ1, EQ2, PRMAT(3,3))	1"
	DPRE	(DEQ1, DEQ2, DPRMAT(3,3))	full
	CUV	(ALPH, BETA, X, Y, Z)	10^{-6}
	DCUV	(DALPH, DBETA, DX, DY, DZ)	full
	UVC	(X, Y, Z, ALPH, BETA)	1"
	DUVC	(DX, DY, DZ, DALPH, DBETA)	full
IV.	RED	(ANGLE, K, L)	1"
	DRED	(DANGLE, K, L)	full
	A13	(DATJUL, A1, A2, A3)	$0^{\circ}.02$
	DA13	(DATJUL, DA1, DA2, DA3)	full
	A46	(DATJUL, A4, A5, A6)	$0^{\circ}.2$
	DA46	(DATJUL, DA4, DA5, DA6)	full
	SUL	(DATJUL, A1, A3)	2'
	DSUL	(DATJUL, DA1, DA3)	1'
	EPS	(DATJUL)	1"
	DEPS	(DATJUL)	full
	ECC	(DATJUL)	10^{-8}
	DECC	(DATJUL)	full
	ECE	(ECC, A3)	1"
	DECE	(DECC, DA3)	$0''.02$

In the following is a brief description of the majority of the names contained in Table 2. We will list the single precision names only, since the double precision names have the same explanation and are in the same units:

I. TIME AND CALENDAR PROBLEMS

IY, IM, ID	=	calendar date on the zone time meridian (example: IY=1967, IM=12, ID=31)	
IWD	=	day in the week (IWD=1 for Monday,....,=7 for Sunday)	
ZTL	=	longitude of zone time meridian	in REV i.e., $24^h=1$
ØBL	=	longitude of observer	in REV i.e., $24^h=1$
STIM	=	local mean sidereal time	in REV i.e., $24^h=1$
ZTIM	=	zone time	in REV i.e., $24^h=1$
IAMB	=	ambiguity control; IAMB=0 in the first half of the calendar day, IAMB=1 in the second half	
DATJUL	=	julian date of any instant of time (example: for IY=1967, IM=12, ID=31, ZTL=0, ØBL=0, ZTIM=0: DATJUL=2439855.5000..)	
DJL	=	julian date of 0^h U.T., for the given Greenwich calendar date	
EQT	=	equation of time = apparent minus mean solar time, in REV	
SUNRA	=	right ascension of the sun	in RAD
SUNDEC	=	declination of the sun	in RAD

II. APPARENT PLACES

PSIDEL	=	total nutation in longitude	in RAD
EPSDEL	=	total nutation in obliquity	in RAD
SPSI	=	short period nutation in longitude	in RAD
SEPS	=	short period nutation in obliquity	in RAD
EQ	=	mean equinox of given source position (example: 1968.0 for observing dates between July 1, 1967 and June 30, 1968)	
EQALPH	=	mean right ascension for EQ	in RAD
EQDELT	=	mean declination for EQ	in RAD
SCRA	=	Besselian Star Constants in right ascension in RAD (SCRA(1)=a, SCRA(2)=b,...,SCRA(4)=d)	
SCDEC	=	Besselian Star Constants in declination in RAD (SCDEC(1)=a', SCDEC(2)=b',..,SCDEC(4)=d')	
BDNS	=	Besselian Day Numbers in RAD (BDNS(1)=A, BDNS(2)=B,...,BDNS(5)=E)	

III. ORTHOGONAL COORDINATE TRANSFORMATIONS

MAT = Transformation Matrix
ALPH1 = given spherical angle along fundamental plane in RAD
BETA1 = given spherical angle perpendicular fundamental plane in RAD
ALPH2 = value of ALPH1 in new system } in RAD
BETA2 = value of BETA1 in new system
EQ1,EQ2 = the two mean equinoxes for general precession (for example: EQ1=1950.0, EQ2=1968.0 (forward); EQ1=1968.0, EQ2=1950.0 (backwards))
PRMAT = matrix of general precession. If one has to carry out the precession transformation, one calls TRC(PRMAT,....).
ALPH = spherical angle along fundamental plane } in RAD
BETA = spherical angle perpendicular fundamental plane
X,Y,Z = components of unit vector corresponding to ALPH,BETA

IV. AUXILIARY PROGRAMS

ANGLE = value of any angle
K = unit control; depending on the unit in which the Angle is given, K must obtain one of the following values: K=1 for REV, K=2 for RAD, K=3 for H, K=4 for DEG
RED = value of ANGLE after reduction to standard interval
L = interval control:

L = 1	$0 \le \text{RED} < 2\pi$
L = 2	$-\pi \le \text{RED} \le +\pi$
L = 3	$-\pi/2 \le \text{RED} \le +\pi/2$

RED is, of course, in the same unit as ANGLE. If one calls RED with L=-1, the argument L has after the RETURN the value of N (the number of revolutions included in the reduction; see Ref. (I), section 1, "reduction operator"). If one wishes to use this trick, one should call RED with the following statements:

```
N=-1
REDANG = RED(ANGLE, K, N)
```

If one would call RED(ANGLE, K, -1), the numerical constant "-1" could be "destroyed".

The remaining names are not described here since they will not be used very frequently. For these as well as for all other details the reader is referred to the NRAO PROGRAM DESCRIPTION FILE.

In order to illustrate at least some of these routines, two examples are given.

EXAMPLE 1: <u>Calculation of 1950.0 Positions from Observed Apparent Positions</u>

It is assumed that very accurate apparent hour angles and declinations are obtained from interferometer observations. The mean right ascensions and declinations for the mean equinox 1950.0 are to be computed. The names are defined in the same way as previously mentioned: Names starting with I are INTEGER*4, names starting with D are REAL*8, all other names are REAL*4.

The data input file consists of:

I. IY = calendar year (four digits) } on the zone time meridian
IM = calendar month (two digits)
ID = calendar day (two digits)
CE = clock error (in seconds) for that calendar day, in the sense:
True loc. mean sid. time = indicated loc. mean sid. time plus CE.

II. DSTIND = indicated local mean sidereal time (REV)
IAMB = ambiguity control

IAMB = 0 during the first half of the calendar day.
IAMB = 1 during the second half of the calendar day.

DAPHA = apparent hour angle } RAD
DAPDEC = apparent declination

The logical sequence in which these data appear in the input file will probably be: I-II-II-II-....-II-I-II-II-... and so on, i.e., we will read the calendar date and the clock error, and then the times and the apparent places for several sources observed on that day; then, another date follows with corresponding observational data for other sources. Each time a date is read, the mean equinox belonging to the nearest beginning of a Besselian year must be determined and the precession matrix must be calculated. Of course, if the mean equinox turns out to be the same for consecutive type I data, the precession matrix will be the same and should not be recalculated. In the example we will omit this trivial decision. In the computation of apparent right ascension from apparent hour angle and mean sidereal time, we need the equation of the equinoxes (difference between mean and apparent sidereal time) which includes a factor cos ε (ε=obliquity of the ecliptic). This is, within the accuracy needed, a constant for rather large fractions of the year. However, for illustration we compute it in the example from the given calendar date.

Each time the type II data are read, we have to compute the Besselian Day Numbers and the Besselian Star Constants. The Besselian Day Numbers of course, could be computed for the beginning and for the end of the day and then be interpolated for any time on that day. For the sake of brevity, however, we will compute them independently for every time. Since we need the julian date for that purpose, we call the routine DZØT which results in DZTIM (the zone time) and in DATJUL (the julian date). The Besselian Day Numbers are computed from DATJUL and EQ (the mean equinox which was already defined after the date was read). The equation of the equinoxes is computed with the factor cos ϵ (mentioned above) and with the nutation in longitude (PSIDEL) which is one of the results of the Besselian Day Number routine. In the rigorous application of the Bessel method, the star constants must belong to the mean equinox, EQ. However, the positions which are observed are apparent positions. In order to take this into account, we first compute approximate mean positions for the beginning of the nearest Besselian year (using the apparent positions in the Besselian Star Constant routine) and carry out one iteration. The source program is listed on the next page (omitting all peripheral organization). The following is a list of some of the names which appear in the source program:

DPRMAT = precession matrix
SCRA = star constants in right ascension
SCDEC = star constants in declination
BDNS = day numbers
DØBL = observers longitude (REV)
DZTL = longitude of zone time meridian (REV)
C2PI = 2π
DEQ = mean equinox (double precision)
EQ = mean equinox (single precision)
DSTIM = true mean local sidereal time (REV)
DAPST = true apparent local sidereal time (RAD)
DAPRA = apparent observed right ascension
DEQRA = mean right ascension at EQ
DEQDEC = mean declination at EQ
DELRA = DAPRA - DEQRA
DELDEC = DAPDEC - DEQDEC

Essential Parts of Source Program (Example 1):

```
C      DEFINITIONS AT THE BEGINNING
       IMPLICIT REAL 8 (D)
       DIMENSION DPRMAT(3,3),SCRA(4),SCDEC(4),BDNS(5)
C      INSERT PROPER VALUES FOR YOUR STATION IN THE NEXT TWO
C      STATEMENTS
       DØBL=
       DZTL=
       C2PI=6.2831853071D0
       .
       .
       .
C    1 READ THE TYPE I DATA
       READ (.....)  IY, IM, ID, CE
       CE=CE/86400.0
       DEQ=IY
       IF(IM.GE.7) DEQ=IY+1
       EQ=DEQ
       CØSEPS=CØS(EPS(DJL(IY,IM,ID)))
       CALL DPRE(DEQ,1950.0D0,DPRMAT)
       .
       .
       .
C    5 READ THE TYPE II DATA
       READ(.....)  DSTIND, IAMB, DAPHA, DAPDEC
       DSTIM=DSTIND+CE
       CALL DZØT(IY,IM,ID,DSTIM,IAMB,DØBL,DZTL,DZTIM,DATJUL)
       CALL BDN2(DATJUL,EQ,BDNS,PSIDEL,SPSI,SEPS)               *)
       DAPST=DSTIM*C2PI + CØSEPS*PSIDEL
       DAPRA=DAPST-DAPHA
       DEQRA=DAPRA
       DEQDEC=DAPDEC
       DØ 20  I=1,2
       CALL BSC(EQ,SNGL(DEQRA),SNGL(DEQDEC),SCRA,SCDEC)
       DELRA=BDNS(5)
       DELDEC=0.0D0
       DØ 10  K=1,4
       DELRA =DELRA   + BDNS(K)*SCRA(K)
   10  DELDEC=DELDEC + BDNS(K)*SCDEC(K)
       DEQRA =DAPRA - DELRA
   20  DEQDEC=DAPDEC- DELDEC
       CALL DTRC(DPRMAT,DEQRA,DEQDEC,DRA50,DEC50)
       .
       .
       .
```

The results are, in each step, the right ascension (DRA50) and the declination (DEC50) for the mean equinox 1950.0, in RAD.

---

*)SPSI and SEPS are results which are not used in this application.

EXAMPLE 2: Calculate Apparent Right Ascension and Declination for Sources with Given Galactic Coordinates

We assume that 25 sources are given by their galactic coordinates ℓ^{II} and b^{II}. We want to calculate a list of their apparent positions with low accuracy for a number of days. The input of the program shall consist of:

GALØNG(25)	= galactic longitude (RAD)
GALAT(25)	= galactic latitude (RAD)
IYB, IMB, IDB	= calendar date of the first day of the list
NDAY	= total number of days, including the first
ZTIM	= zone time (the same for all days) at which the apparent places have to be calculated (in hours)
ØBL	= observer's longitude (RAD)
ZTL	= longitude of the zone time meridian (hours)

The apparent places of the 25 sources for each day are listed on one page. The page starts with a header (year, month, day, name of the weekday, zone time, local mean sidereal time). Apparent right ascensions and declinations of the sources follow. We omit the formats and other unessential items such as names or numbers of sources, etc.

In a first loop, the given galactic coordinates are converted into right ascensions and declinations for 1950.0 (RA50, DEC50). The second loop which proceeds from one day to the other is realized by computing the julian dates corresponding to the first and the last calendar date. Inside the loop the actual calendar date and the day of the week is calculated with the calendar routine (CLD). For each new day the mean equinox of the nearest beginning of a Besselian year (EQ) is computed; if the value of EQ differs from the last computed one the precession matrix is calculated and the source positions are transformed to the new equinox (EQRA,EQDEC); then, the Besselian star constants are computed and stored for each source (SRA(N,K) and SDC(N,K)). The julian date for each day and for the given constant zone time on each day is calculated with the routine SIT which also gives the corresponding local mean sidereal time. The computation of the Besselian Day Numbers with BDN1 follows. Finally, the apparent places (APRA, APDEC) are calculated and printed.

Essential Parts of Source Program (Example 2):

```
      REAL*8  DJL,DJUL,DJULE,DATJUL
      REAL*4  GALMAT(3,3)/-0.6698874,-0.8727558,-0.4835389,
     1 0.4927285,-0.4503470,0.7445846,-0.8676008,-0.1883746,
     2 0.4601998/, DAY(7)/' MØN TUE WED THU FRI SAT SUN'/
      REAL*4  PREMAT(3,3),BDNS(5),SCRA(4),SRA(25,4),SCDEC(4),
     1 SDC(25,4),GALØNG(25),GALAT(25),RA50(25),DEC50(25),
     2 EQRA(25),EQDEC(25)
      EQØLD=0.0
      READ(.....)GALONG, GALAT,
      DØ 10  N=1,25
   10 CALL TRC(GALMAT,GALØNG(N),GALAT(N),RA50(N) DEC50(N))
      READ(.....)IYB,IMB,IDB,NDAY,ZTIM,ØBL,ZTL
      ZT  =ZTIM/24.0
      ØBL =ØBL/6.2831853
      ZTL =ZTL/24.0
      DJUL=DJL(IYB,IMB,IDB) = 1
      DJULE=DJUL + NDAY
   20 DJUL=DJUL + 1
      CALL CLD(DJUL,IY,IM,ID,IWD)
      DAYNAM=DAY(IWD)
      CALL SIT(IY,IM,ID,ZT,ØBL,ZTL,ST,DATJUL)
      STIM=ST*6.2831853
      PRINT .....,IY,IM,ID,DAYNAM,ZTIM,STIM
      EQ=IY
      IF(IM.GE.7) EQ=IY+1
      IF(EQ.EQ.EQØLD) GØ TØ 50
      EQØLD=EQ
      CALL PRE(1950.0,EQ,PREMAT)
      DØ 40  N=1,25
      CALL TRC(PREMAT,RA50(N),DEC50(N),EQRA(N),EQDEC(N))
      CALL BSC(EQ,EQRA(N),EQDEC(N),SCRA,SCDEC)
      DØ 30  K=1,4
      SRA(N,K)=SCRA(K)
   30 SDC(N,K)=SCDEC(K)
   40 CØNTINUE
   50 CALL BDN1(DATJUL,EQ,BDNS,PSIDEL)
      DØ 60  N=1,25
      APRA=EQRA(N)  + BDNS(1)*SRA(N,1)+BDNS(2)*SRA(N,2)+BDNS(3)*SRA(N,3)
     1 + BDNS(4)*SRA(N,4)
      APDEC=EQDEC(N)  + BDNS(1)*SDC(N,1)+BDNS(2)*SDC(N,2)  +
     1 BDNS(3)*SDC(N,3)  + BDNS(4)*SDC(N,4)
   60 PRINT  .....,   APRA, APDEC
      IF(DJUL.GT.DJULE)  STØP
      GØ TØ 20
      .
      .
      .
```

We would like to add some concluding remarks. All the routines will reside permanently on disk. That means that one has to be careful in the choice of names of other subprograms which may appear in a main program. Even if the user calls only one of these routines, he should avoid any of the names of the other routines. For example, if only NUT1 would be called, the names A13 and A46 are still forbidden since they are assigned to programs internally called by NUT1.

Our routines were constructed in a very general way. This, of course, could mean that they are not in all applications the most effective solutions for a special problem, as far as computing time is concerned.

We have done extensive testing of all of the programs. At present, they seem to be very well debugged. However, since more than 200 essentially different constants are distributed over more than 300 different places within this "system", with different precision, it will be understood that we cannot entirely exclude the possibility of errors. We would be grateful if the users would inform us about errors.

References: (I) NRAO Computer Division Internal Report No. 2
(II) NRAO Computer Division Internal Report No. 3