```
      PROGRAM COVFIT16
C $Id: covfit16.for 292 2009-10-22 08:54:45Z cct $
C TEST AND FITTING OF COVARIANCE FUNCTIONS. ORIGINALLY PROGRAMMED BY
C C.C.TSCHERNING, DEP. OF GEODETIC SCIENCE, THE OHIO STATE UNIVERSITY,
C AND GEODAETISK INSTITUT, DANMARK, JUNE 1975.
C MODIFIED SEPT 1985 BY C.C.TSCHERNING (ADAPTED TO RC FORTRAN).
C ADDITIONS 1986 BY P.KNUDSEN FOR COVARIANCE FUNCTION FITTING.
C LATEST MODIFICATION: SEP 01, 2011 BY CCT.
C (C) COPYRIGHT BY C.C.TSCHERNING AND P.KNUDSEN, 1975, 1985, 1987, 1988,
C     1989, 1990, 1991, 1996, 2009.
C REFERENCES:
C (A) TSCHERNING,C.C.: COVARIANCE EXPRESSIONS FOR SECOND AND LOWER ORDER
C     DERIVATIVES OF THE ANOMALOUS POTENTIAL, REPORTS OF THE DEP. OF
C     GEODETIC SCIENCE NO. 225,1975.
C (C) KRARUP, T. AND C.C.TSCHERNING: EVALUATION OF ISOTROPIC COVARIANCE
C     FUNCTIONS OF TORSION BALANCE OBSERVATIONS. BULLETIN GEODESIQUE,
C     VOL. 58, NO. 2, PP. 180-192, 1984.
C (D) KNUDSEN, PER: ESTIMATION AND MODELLING OF THE LOCAL EMPIRICAL
C     COVARIANCE FUNCTION USING GRAVITY AND SATELLITE ALTIMETER DATA.
C     BULLETIN GEODESIQUE, VOL. 61, PP. 145-160, 1987.
C (E) Sanso, F. and W.-D. Schuh: Finite Covariance Functions. Bulletin
C     Geodesique, Vol. 61, pp. 331-347, 1987.
C (F) Tscherning, C.C.: Prediction of spherical harmonic coefficients
C     using Least-Squares Collocation. J. of Geodesy, 2000.
C
      IMPLICIT NONE
      INTEGER MAXSM,ISTEP,MAXSTI,MAXO,NSAT,NCTA,MXPAR,NIPT,NIPCAT,MAXOD
C
      PARAMETER (MAXSM=2200,ISTEP=1,MAXSTI=2200)
C CHANGE 2009-10-19 from 360 to 2200.
      PARAMETER (MAXO=5600,NSAT=5600,NCTA=1600,MXPAR=2500,NIPT=1500,
     *NIPCAT=100002,MAXOD=50400)
C MAXSM IS THE MAXIMAL NUMBER OF EMPIRICAL OR ERROR DEGREE-VARIANCES.
C ISTEP IS THE STEP BETWEEN EACH SUMMATION INDEX USED FOR MODE 7.
C MAXXTI IS THE MAXIMAL NUMBER OF STEPS FOR MODE 7.
C
      INTEGER NWAR,
     *KCI,NC1,NC2,NFU,KEYH,NINTH,NTABH,NHE,IOBS2,NSTART,KSAT,
     *NDX1,NDX2,I4,NDP,IPACAT,NDQ,IT,K3,K4,NUM,INN,ITCOUN,
     *IGP,NBOLD,IA1,IKP,IU1,IC11,IMAX1,IMAX1R,INV,ITE1,
     *ITIME0,KK,ITIME,INUMR,IP1,K21,K2,IU,IITE1,IITE,IIP1,IIP,IIE,
     *IIE1,K2P3,IT1,ITE,IP,IC1,IA,IB,NNX,NTABX,IFQ,ISATP,ISAT,
     *IHQ,IHP,INDEX,NR,NI,ICZERO,J2,K8,INZOLD,IEM,K21X,INL,K17,NAI,
     *K15,ICSYSL,K11,NO1,K9,K7,NO,IOBS1,IANG,IH,MP,IPAR,IFP,
     *KFQ,JR,NOBLK,IXX,K13,K19,NCZERO,NLA,INO,IB1,ISO,IPX,IS,
     *JJDEG,IIDEG,K1,IPTYPE,K23,I3,IPA,KFP,NPARM,NPARM1,MAXPAR,
     *II,NMAX,MAXB,IX,NCXLAS,MODE,IJ,N2,KTYPE,KI,IK,IK1,I,IMAX,ISTART,
     *ICHAR,MAXC1,MODEL,IMIN,MODEL1,IMA100,NT,NT1,NTMAX,J,KPP,KQQ,NLAT,NLON,
     *NMIN,NTMAX1,NLON,NCOV,IDEG,MIN,MV,MT,MODEFI,MAXFI,K,
     *NMIN,IFT,KX,KY,IQ,MG,KP,KQ,NFILTE,NSTEP,NSTEPE,NSTEQ,NSTEQE,IEHD,
     *KPSI,M,NLAT1,NLON1,MODERR,IKQ,IDAT,ISET,IC3,IC6,NDAT,NDSET,NITMAX,
     *MLK,KLM,ND2,MM,KPZ,KQZ,NPROD,NDATO1,NA,NITER,IIX,NFOURI,NITS,
     *NIT,III,IOLD,IDDY
C
      REAL*8 GM,RLOMAX,RLAMAX,RLOMIN,RLAMIN,B,HQ,RLAT,
     *SINLAT,COSLAT,RLONG,SINLON,COSLON,WOBS,SINLOP,
     *COSLOP,BSIZE,BSIZEN,BSIZEE,COSLAP,SINLAP,RLONGP,RP,CAZP,SAZP,
     *CCI,CCR,SIGMA0,SIGMA,HCMAX,CCV,D,OBS,OLDR,SLOQ,CFX,
     *RE,BIPC,CRHT,PREDP,HP,RLATP,BIP,HQOLD,C11,CTA,CTTF,CTSF,
     *SZ,AZ,HTA,TMAX,SIZEI,COVX,CIX,SLOP,D2,CLOP,CLOQ,GMC,PI,DXX,HCZERO,
     *VARI,DGPM2,SCALE,SCALE2,DRAPP,OLDT,RADSEC,CFA,SIGMAP,HPOLD,
     *D5,D0,D1,D3,D4,PRETAP,HMAX,SUMSIG,R,VARDG2,S,DR,RB2,AAI,SM,CVV,VAR,
     *DEGRAD,SROT,VG,CVV0,A0,DLON,DLAT,DEGRAD,VAR,
     *VZERO,DT,SEC,ALFA,RT,FAK1,COV45,FOUCOF,
     *RLCH,T2,CA,SA,SIGMAF,T1,RDD,SCFACT,RLATPP,FILTER,COSLA,SINLA,
     *STEPE,COSSTE,SINSTE,STEPN,COSSTN,SINSTN,SHIFTS,COST2P,SINt2P,
     *BSIZEA,SAZQ,CAZQ,STEQE,COSSQE,SINSQE,STEQN,COSSQN,SINSQN,
     *COST2Q,SINt2Q,AZIM,COZERO,FINDR,PSI1,TEST2,C3,TEST1,C2,OSI,C0,
     *CR,CA1,SA1,SA2,CA2,CA3,SA3,THAUR,HEIAR,AZIMR,HEIANG,PSI,
```



```
     *C1,THAU,DPSI,TEST3,RLONQ,RLATQ,CQ,SQ,CD,SD,t,RV,COV,U,RLONGQ,
     *COSLAQ,SINLAQ,COSLOQ,SINLOQ,COVXX,COMEAN,CC1,WEIGHT,HHQ,HHP,TAU,
     *TAU0,RM,COVCG,COX,CC2,FI1,FI2,DFI,ALA1,ALA2,DLA,CC,RPROD,ERR,ERRY,
     *VGOLD,RB2OLD,SOLD,CCI10,TAU1,TAU2,TAU3,DX,DEV2,AX,RTA,FAUCOF,
     *SCFRDD,COV14,ERRX,DEV21,DEV,COV25,DXD3,DXD2,DXD1,DEV1,DEVOLD,A,
     *FAK3,FAK2,FAK,DEV2A,ERRK,BX,HQ1,HPX,RDDY
      LOGICAL LSUM,LOCAL,LA,LPOLE,LAST1,LAST2,LAST3,LNEW,LTEST,
     *LZERO,LT,LF,LONECO,LNKSIP,LNETAP,LDEFVP,LWR,LPUNCH,LDEN,
     *LDENP,LDENQ,LREDUC,LBACK,LTABH,LTABHX,LINVAR,LSUMC,
     *LFILE,LSAT,LINTER,LTESTS,LX,LNX,LMULTF,LSPOUT,LITERA
     *,LOCOLD,LFINI,LFOURI,LLCOEE,LOBSST,LGRIDO,LFINOU,LMEAP,LMEAQ,
     *LEQANG,LMEAP1,LFILTE,LEQANQ,LMEAQ1,LCZERO,L,LN
      CHARACTER*128 DNAME
      CHARACTER*128 PNAME,FINAME
      COMMON /CMCOV/CCI(24),CR(56),SIGMA0(MAXSM),SIGMA(MAXSM),HMAX,
     *CCV(2,2),D(36),KI(37),NC1,N2,LOCAL,LSUM,LFINI
      COMMON /DDX/K7(17),K9(17),K11(17),K13(17),K15(17),K17(17),
     *K19(17),K21X(17),K23(17),K8(17),C11(17),J2(2),I3(2),I4(2),
     *LN(7),L(7)
C     COMMON /DDX/K7(15),K9(15),K11(15),K13(15),K15(15),K17(15),K19(15),
C    *K21(15),K23(15),K8(15),C11(15),J2(2),I3(2),I4(2),LN(7),L(7)
C     COMMON /PR/DRAPP(181),DGPM2(201)
C THE ARRAYS CONTAINS ERROR DEGREE-VARIANCES FOR RAPP 1981 AND
C GPM2 SOLUTIONS.
      COMMON /DDY/RDDY(5),IDDY(12),ND2
C     COMMON /PR11/SINLOP,COSLOP,BSIZE(42),BSIZEN,BSIZEE,
C    *COSLAP,SINLAP,RLONGP,RP,
C    *HPP,RLATP,PRETAP,PREDP,LONECO,LNKSIP,LNETAP,LDEFVP,
C    *NI,NR,INDEX(42),IKP
      COMMON/PR/BX(MAXO),HQ1(MAXO),RLAT(MAXO),
     *SINLAT(MAXO),COSLAT(MAXO),RLONG(MAXO),
     *SINLON(MAXO),COSLON(MAXO),WOBS(MAXO),
C    *COSAZ(NSAT),SINAZ(NSAT),SINLOP,COSLOP,BSIZE(42),BSIZEN,BSIZEE,
     *SINLOP,COSLOP,BSIZE(42),BSIZEN,BSIZEE,
     *COSLAP,SINLAP,RLONGP,RP,CAZP,SAZP,
     *HPX,RLATP,PRETAP,PREDP,HCZERO,ICZERO,NCZERO,
     *NI,NR,INDEX(42),IKP,ISAT(42),ISATP,NOBLK,
     *LONECO,LNKSIP,LNETAP,LDEFVP,LOBSST
      COMMON /TABELC/CTA(1600,16,2),CTTF(800),CTSF(20),SZ(30),AX(18),
     *MAXB(20),IX(8),IIX(17),IOLD
      COMMON /CTABH/DRAPP(181),DGPM2(201),RTA(818),HTA(5),TMAX,SIZEI,
     *NFU(5),KEYH(5,5),NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFA,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
      COMMON /CMEAN/STEPN,COSSTN,SINSTN,STEPE,COSSTE,SINSTE,
     *COST2P,SINT2P,FILTER(11),NFILTE
      COMMON /CMEAQ/STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE,
     *COST2Q,SINT2Q
C     COMMON /CLPARM/SCFACT,RDD,KP1,KQ1,LPARMP,LPARMQ
      COMMON /CLPARM/SCFRDD(42),SCFACT,RDD,FOUCOF(0:21),NFOURI,LFOURI,
     *LLCOEE(42)
      COMMON /PDEGV/SIGMAP(2200),SLOP,SLOQ,CLOP,CLOQ,
     *IIDEG,JJDEG,LSPOUT
      COMMON /CCOZERO/SIGMAF(MAXSM),T1,T2,MAXFI
C THE COMMON AREAS ARE USED FOR THE TRANSFER OF DATA TO AND FROM THE
C SUBROUTINES COVAX, CTABEL AND COVCG.
C     DIMENSION AA(18), COV(181,7),KP(180),KQ(180),HP(7),HQ(7)
      DIMENSION COV(181,7),KP(180),KQ(180),HP(7),HQ(7)
     *,A(6),B(3),ERRY(180),DX(3),ERRX(3),TAU(3),TAU0(3),
     *KX(3),KY(3),IP(3),IQ(3),LA(3),SM(MAXSM),DNAME(2),SROT(3,3),
     *DEV2A(MAXSTI,6),NITS(MAXSTI)
C RE IS THE MEAN RADIUS OF THE EARTH AND GM IS THE PRODUCT OF THE GRAVI-
C TATIONAL CONSTANT AND THE MASS OF THE EARTH.
C
      PI=D4*DATAN(D1)
      DEGRAD=PI/180.0D0
      LWR=LF
      LSAT=LF
```

```
      LDEN=LF
      LSPOUT=LT
      WRITE(6,10)
   10 FORMAT(/' FITTING OR TABULATION OF COVARIANCE FUNCTIONS,',
     *' VERS.  15 OKT 2008.',//,
     *' THE KINDS AND CORRESPONDING UNITS ARE AS FOLLOWS: (E=EOTVOS'
     *,'):',/,
     *' (1) THE HEIGHT ANOMALY (METERS), (2) THE NEGATIVE RADIAL DER-',/
     *,' IVATIVE (THE GRAVITY DISTURBANCE), (3) THE GRAVITY',/,
     *' ANOMALY (MGAL), (4) THE RADIAL DERIVATIVE OF (3) (E), (5) THE',/
     *' SECOND ORDER RADIAL DERIVATIVE (E), (6),(7) THE LATITUDE AND',/,
     *' THE LONGITUDE COMPONENTS OF THE DEFLECTIONS OF THE VERTICAL',/,
     *' (ARCSECONDS), (8),(9) THE DERIVATIVES OF (3) IN NORTHERN AND',/,
     *' EASTERN DIRECTION, RESPECTIVELY (E), (10),(11) THE DERIVATIVE',/
     *,' OF (2) IN THE SAME DIRECTIONS (E), (12) - (15) THE SECOND',/
     *,' ORDER DERIVATIVES WITH RESPECT TO LATITUDE, IN NORTHERN',/,
     *' AND EASTERN DIRECTION * 2, WITH RESPECT TO LONGITUDE, AND',/,
     *' IN EASTERN MINUS NORTHERN DIRECTIONS, RESPECTIVELY (E).',/,
     *' (0) OR (16) DENSITY ANOMALIES IN G/CM**3*10.'/,
     *' (17) COEFFICIENT OF THE F.NORM. SPHERICAL HARMONIC Y(I,J),',/
     *' 100 ADDED INDICATES 1 OR 2-D MEAN & NEGATIVE ROTATED. '//)
C
C THE COMPUTATIONS AND INPUT FLOW IS GUIDED BY THE VALUE OF A
C MODE-PARAMETER:
C  MODE=1: COMPUTATIONS BY COVAX,
C  MODE=2: COMPUTATIONS BY COVCG IF WITHIN RANGE OF TABEL,
C  MODE=3: DIFFERENCES BETWEEN RESULT OF COVAX AND COVCG.
C  MODE=4: FITTING OF EMPIRICAL COVARIANCE FUNCTIONS.
C  MODE=5: TABULATION OF FINITE COVARIANCE FUNCTION, CF. REF (E).
C  MODE=6: TABULATION OF  FINITE OR DIFFERENCE BETWEEN ANALYTIC AND FINITE
C          COVARIANCE FUNCTIONS.
C  MODE=7: LIKE MODE 4, BUT THE VALUE OF SUMMATION LIMIT IS CHANGED
C          FROM IMAX TO NMIN IN STEPS OF ISTEP (=5).
C
C THE INPUT SEQUENCE IS AS FOLLOWS:
C (1) INPUT OF MODE AND OTHER PARAMETERS, INCLUDING LAST1, TRUE IF
C     NO MORE INPUT FOR MODE 1 - 3.
C (1B) FOR MODE=7 ONLY: NMIN.
C (2) INPUT OF DEGREE-VARIANCE TYPE,
C (3) INPUT OF CONSTANTS USED TO SPECIFY THE DEGREE-VARIANCE MODEL
C     AND FOR MODE 2 OR 3 TABULATION SPECIFICATIONS.
C IF MODE=4 OR 7, JUMP TO (6)
C (4) INPUT SPECIFICATION OF TABLE, INCLUDING LAST2, TRUE IF LAST
C     SPECIFICATION.
C (5) INPUT OF CONSTANTS SPECIFYING TYPE OF FUNCTIONALS AND HEIGHTS,
C     AND LOGICAL VARIABLE LAST3, TRUE IF NO MORE INPUT OF TYPE (5).
C IF LAST3 IS FALSE GO TO (5) ELSE IF LAST2 IS FALSE GO TO (4) ELSE
C IF LAST1 IS FALSE GO TO (1) ELSE STOP.
C (6) INPUT NUMBER OF ITERATIONS TO BE EXECUTED AND OTHER CONSTANTS,
C (7) INPUT OF NUMBER OF TABLES AND KIND OF TABLES.
C     CONTINGENTLY INPUT OF PARAMETERS OF FINITE COV.FCT., MEAN VALUE
C     COV.FCT. OR ROTATED COV.FCT.
C (8) INPUT EMPIRICAL COVARIANCE TABLES,
C
C
C ************************ INPUT (0) **************************
C
C INPUT OF LINTER, TRUE IF INTERACTIVE JOB.
      WRITE(*,*)' INTERACTIVE ? (T/F) ? '
      READ(*,*)LINTER
C
C ************************ INPUT (1) **************************
C INPUT OF:
C MODE,
C LTEST, TRUE WHEN TEST OUTPUT IS NEEDED,
C LSUM, TRUE WHEN A FINITE LEGENDRE
C   SERIES (MAXIMAL DEGREE N2 < 2000) MUST BE USED FOR THE
C   EVALUATION OF COVARIANCES IN ALTITUDES GREATHER THAN HMAX,
C LAST1, TRUE WHEN THERE IS NO MORE INPUT.
```



```
C LPUNCH, TRUE WHEN OUTPUT OF COVARIANCES IS NEEDED ON A FILE 7.
  100 IF (LINTER) WRITE(*,31)
   31 FORMAT(
     *' INPUT MODE (1-7), TEST? (T/F), FUNCTION GIVEN AS LEGENDRE SUM?'/
     *' LAST COMPUTATION ? , OUTPUT TABLE TO FILE ? (T/F)')
      READ(5,*)MODE,LTEST,LSUM,LAST1,LPUNCH
      WRITE(6,49)MODE
      LFINI=MODE.EQ.5.OR.MODE.EQ.6
      ICHAR=1
      IF(.NOT.LPUNCH) GOTO 311
C
C ---------------------- INPUT (1A) --------------------------------
C
      IF (LINTER) WRITE(*,*)' INPUT NAME OF FILE TO HOLD TABLE'
      READ(5,'(A)')DNAME(1)
      WRITE(6,402) (DNAME(IJ),IJ=1,ICHAR)
      OPEN(7,FILE=DNAME(1),STATUS='UNKNOWN')
  402 FORMAT(' OUTPUT OF COVARIANCES ON FILE: ',2A128//)
C
  311 CONTINUE
   49 FORMAT(/'  COMPUTATION MODE=',I3)
      IF (MODE.EQ.2.OR.MODE.EQ.3) THEN
C ---------------------- INPUT (1B) --------------------------------
       WRITE(*,*)
     * ' TABULATION  IN FIXED HEIGHT TO BE USED/TESTED ? (T/F)'
       READ(*,*)LTABHX
       WRITE(*,*)LTABHX
      END IF
      IF (MODE.EQ.3) WRITE(6,12)
   12 FORMAT(' VALUES IN TABLES ARE DIFFERENCES BETWEEN COVAX',
     *' AND COVCG OR TABH.')
      IF (MODE.EQ.6) WRITE(6,11)
   11 FORMAT(' VALUES IN TABLES ARE DIFFERENCES BETWEEN COVAX',
     *' AND  A FINITE COVARIANCE FUNCTION.')
C
C ---------------------- INPUT (1C) --------------------------------
      IF (MODE.EQ.7) THEN
       WRITE(*,*)' MODE7: ITERATIVE FITTING '
       LITERA=LT
       MODE=4
       WRITE(*,*)' INPUT LOWER LIMIT FOR ITERATION '
       READ(*,*)NMIN
       WRITE(*,*)' ITERATION STOPS AT NMIN ', NMIN
      ELSE
       LITERA=LF
      END IF
C
C IF LSUM IS TRUE INPUT OF N2, HMAX (IN METERS).
      N2=2
      IF (LSUM) THEN
C ---------------------- INPUT (1D) --------------------------------
       IF (LINTER)
     * WRITE(*,*)' INPUT SUMMATION LIMIT AND HEIGHT (M)'
       READ(5,*)N2,HMAX
       IF (N2.GT.2200) N2 = 2200
       WRITE(6,6)HMAX,N2
    6  FORMAT(' WHEN THE HEIGHT OF ONE OF THE POINTS OF EVALUATION IS ABO
     * VE ',E14.7,' METERS',/,' WILL THE COVARIANCES BE EVALUATED BY MEAN
     * S OF A LEGENDRE SERIES HAVING ',I5,' TERMS.')
      END IF
      N2 = N2+1
      SUMSIG = D0
      MAXC1 = 1
C IS IS A POINTER USED IN GEOCOL IN ORDER TO HAVE TWO SETS OF
C COVARIANCE PARAMETERS AVAILABLE SIMULTANEOUSLY. THIS FEATURE IS
C NOT USED HERE.
      IS = 0
C
C ******************* INPUT (2)*****************************
```

```
C
C INPUT OF THE INTEGER KTYPE DETERMINING TYPE OF DEGREE-VARIANCE
C MODEL USED FOR DEGREE-VARIANCES OF DEGREE GREATHER THAN IMAX
C (SEE BELOW). KTYPE MAY BE EQUAL TO 1, 2, OR 3, CORRESPONDING
C TO THE DEGREE-VARIANCE MODEL NUMBERS OF REF(A).
      IF (LINTER)WRITE(*,*)' INPUT TYPE OF COV. FCT. MODEL (1,2,3)'
      READ(5,*)KTYPE
 1102 FORMAT(I2)
      IF (KTYPE.LE.0 .OR. KTYPE.GE.4) THEN
       WRITE(*,*)' KTYPE OUT OF RANGE ',KTYPE
       STOP
      END IF
      KI(5)=KTYPE
      IK=0
      IK1=0
      IF (KTYPE.LT.2) GO TO 1036
      IF (LINTER)WRITE(*,*)' INPUT INTEGER(S) IN NUMERATOR'
      IF (KTYPE .EQ. 2) READ(5,*)IK
      IF (KTYPE .EQ. 3) READ(5,*)IK,IK1
      IF (IK.LT.0.OR.IK1.LT.0) THEN
       WRITE(*,*)' WARNING, NEGATIVE VALUES ',IK,IK1
       STOP
      END IF
C
 1036 KI(3)=IK
      KI(4)=IK1
      WRITE(6,1141)
 1141 FORMAT(/' THE MODEL ANOMALY DEGREE-VARIANCES ARE EQUAL TO'/,
     *' A*(I-1)')
      GO TO (1038,1039,1037),KTYPE
 1038 WRITE(6,143)
  143 FORMAT('+',8X,'/(I-2).')
      GO TO 1000
 1039 WRITE(6,144)IK
  144 FORMAT('+',8X,'/((I-2)*(I+',I4,')).')
      GO TO 1000
 1037 WRITE(6,142)IK,IK1
  142 FORMAT('+',8X,'/((I-2)*(I-',I4,')*(I-',I4,')).')
C
 1000 DO 1035 I = 1, MAXSM
 1035 SIGMA(I) = D0
C
      SUMSIG = D0
      MAXC1 = 1
C
C ************************ INPUT (3) ****************************
C
C INPUT OF CONSTANTS USED FOR THE SPECIFICATION OF THE DEGREE-VAR-
C IANCE MODEL:
C R      - RATIO BETWEEN THE BJERHAMMAR-SPHERE RADIUS AND THE
C          MEAN RADIUS OF THE EARTH (RE), IF POSITIVE. IF NEGATIVE IT
C          IS THE DEPTH TO THE BJERHAMMAR SPHERE IN KM.
C VARDG2 - VARIANCE OF GRAVITY ANOMALIES AT ZERO ALTITUDE. IF A NEGA-
C          TIVE VALUE IS USED, VARDG2 WILL BE COMPUTED, USING THE
C          (POSITIVE) VALUE AS THE SCALING CONSTANT A.
C IMAX   - MAXIMAL DEGREE FOR EMPIRICAL DEGREE-VARIANCES.
C LZERO  - TRUE IF ALL EMPIRICAL DEGREE-VARIANCES ARE ZERO.
C THIS IS THE FOLLOWED BY FURTHER DETAILS:
C (B) SPECIFICATION OF MODEL FOR THE VARIANCES.
C (C) IF MODE  =2 OR 3, THEN TABLE SPECIFICATIONS.
      IF (LINTER) THEN
       WRITE(*,32)
   32  FORMAT(
     * ' INPUT -DEPTH TO BJ.SPH.(KM), GRAVITY VARIANCE (MGAL**2),'/
     * ' MAX DEGREE OF ERROR DEGREE-VARIANCES (EDGV)'/,
     * ' EDGV ALL ZERO ? (T/F), ')
      END IF
      READ(5,*)R,VARDG2,IMAX,LZERO
      IF (R.GT.D1) GO TO 999
```



```
      LINVAR=VARDG2.LT.D0
      IMAX1=IMAX+1
      ISTART=IMAX
C
      IF (R.GT.D0) S=RE*(R-D1)
C CONVERSION FROM KM TO M.
      IF (R.LT.D0) S=R*1.0E3
      CCI(10)=S
      DR=S
      IF (R.LT.D0) R=(RE+S)/RE
      RB2=(S+RE)**2
C CHANGED BACK TO ORIGINAL 1991.06.12 BY CCT.
      IF (LINVAR) AAI=-VARDG2*1.0D-10*RB2
      IF (.NOT.LINVAR) AAI=RB2*1.0E-8
      CCI(8) = AAI
      LOCAL=LT
      NC1=IMAX1
      NC2=3
      CALL COVAX(SM,IS)
C CORRECTION 195.07.19 BY CCT. AND ADDITION 2000-09-17.
      LSUMC=LSUM
      LSUM=LF
      CVV=VAR(SM,IS,3,S,AAI,D0,IMAX1,LF,1.0D0,0.0D0,LF,SROT)
      IF (LTEST) WRITE(*,*)' AAI, CVV ',AAI,CVV
      LSUM=LSUMC
C
      LOCAL = LZERO
C THE DEGREE-VARIANCES HAVE BEEN INITIALIZED TO ZERO IN THE BLOCK DATA MODULE.
      IF (LZERO) WRITE(6,1112)IMAX
 1112 FORMAT( I4,' ERROR DEGREE-VARIANCES EQUAL TO ZERO')
      IF (LOCAL) GO TO 1040
C
C --------------- INPUT (3A) ---------------------------------
C INPUT OF MODEL NUMBER, FIRST DEGREE TO BE USED , SCALE FACTOR,
C AND A LOGICAL VARIABLE (LWR) TRUE WHEN A LISTING OF THE DEGREE
C VARIANCES IS NEEDED.
      IF (LINTER) WRITE(*,*)
     *' INPUT ERROR DEG.VAR. MODEL NO., IMIN, SCALE FACTOR AND LIST?'
      READ(5,*)MODEL,IMIN,VG,LWR
C MODEL .LE. 0 IS A MODEL INPUT FROM A FILE, PNAME. IF LT 0, A FURTHER
C LINEAR SCALING IS POSSIBLE.
C MODEL 1 IS A MODEL FOR THE ERROR IN RAPP'S 1978 SET
C MODEL 2 IS THE ERROR DEGREE-VARIANCES FOR RAPP'S 1981 SET,
C MODEL 3 IS THE ERROR DEGREE-VARIANCES FOR WENZELS GPM2 SET.
C MODEL 4 IS A LINEAR MODEL IN THE DEGREE, SO THAT FOR VG=1
C     THE ERROR DEGREE VARIANCE IS EQUAL TO 1.0 AT DEGREE 100.
C MODEL 5 IS A SIMILAR, BUT QUADRATIC MODEL.
C FOR MODEL 2 AND 3 THE INITIALIZATION TAKES PLACE IN THE
C BLOCK DATA MODULE.
      WRITE(6,116)MODEL,IMIN,IMAX,VG
  116 FORMAT(' MODEL ',I3,' USED FROM DEGREE ',I5,' TO ',I5,
     *' WITH SCALE FACTOR= ',F9.6)
C ADDITION 1999.02.06 BY CCT.
      LMULTF=(MODEL.LT.0)
      IF (LMULTF) THEN
       MODEL=0
      END IF
C
      MODEL1=MODEL+1
      DO 1043 I = 2, IMAX
      SIGMA(I+1) = D0
      IF (I.LE.IMIN.OR.MODEL.EQ.0) GO TO 1043
      GO TO (1051,1052,1053,1054,1055),MODEL
 1051 SIGMA(I+1) = (2*I+1)*(VG*9.81)**2
      GO TO 1043
 1052 SIGMA(I+1) = VG*DRAPP(I+1)
      GO TO 1043
 1053 SIGMA(I+1) = VG*DGPM2(I+1)
      GO TO 1043
```

```
 1054 SIGMA(I+1) = I*1.0D-2*VG
      GO TO 1043
 1055 SIGMA(I+1) = I**2*1.0D-4*VG
 1043 CONTINUE
      IF (MODEL.NE.0) GO TO 1063
C
C ------------------------------ INPUT (3B) ------------------------
C
C INPUT NAME OF FILE HOLDING DEGREE-VARIANCES (IN MGAL**2).
C NOTE THAT THE DEGREE-VARIANCE OF ORDER I IS STORED IN SIGMA(I+1).
      IF (LINTER) WRITE(*,*)' INPUT NAME OF FILE HOLDING DEG.VAR.'
      READ(*,'(a)')PNAME
  163 FORMAT(A128)
      WRITE(6,*) ' DEGREE-VARIANCES INPUT FROM FILE ',PNAME,IMIN,IMAX
      OPEN(19,FILE=PNAME)
      READ(19,*)(SIGMA(I+1),I=IMIN,IMAX)
      REWIND(19)
C CHANGE 1999.02.06 BY CCT.
      IF (LMULTF) THEN
       WRITE(*,*)' MULTIPLICATIVE FACTOR USED '
       DO 1064 I=IMIN,IMAX
 1064  SIGMA(I+1)=SIGMA(I+1)*VG
      ELSE
C ------------------------------- INPUT (3C) ---------------------
       WRITE(*,*)' INPUT VALUE FOR I=IMIN '
       READ(*,*)VZERO
       WRITE(*,1071)VZERO
 1071  FORMAT(' LINEAR FACTOR =' ,F8.4,' USED, !! WARNING ! ')
       IF (VZERO.LT.D0) THEN
        WRITE(*,*)' VZERO < 0 '
        STOP
       END IF
       DO 1065 I=IMIN,IMAX
 1065  SIGMA(I+1)=SIGMA(I+1)*(VZERO+(VG-VZERO)*I/(IMAX-1))
      END IF
C
 1063 IF(LWR) THEN
       WRITE(6,1111)IMAX
 1111  FORMAT(I4,' EMPIRICAL ANOMALY DEGREE-VARIANCES FOR DEGREE',
     * ' > 1,'/,' (< 100) IN UNITS OF MGAL**2 (SCALED) : ')
       IMA100=100
       IF (IMAX1.LT.100) IMA100 = IMAX1
       WRITE(6,1198) (SIGMA(I), I = 3, IMA100)
      END IF
 1198 FORMAT(8F8.4)
C
      DO I = 3, IMAX1
       SIGMA0(IS+I)=SIGMA(I)
       SUMSIG = SUMSIG + SIGMA(I)
      END DO
C
      WRITE(*,7469)SUMSIG
 7469 FORMAT(' SUM OF DEGREE-VARIANCES (MGAL) = ',f12.5)
 1040 IF (IMAX1+IS.GE.MAXSM) THEN
       WRITE(6,1108)
 1108  FORMAT(' SUBSCRIPTS OF ARRAY SIGMA EXCEEDS ARRAY LIMIT, STOP.')
       STOP
      END IF
C
      IF (.NOT.LINVAR) AAI=(VARDG2-SUMSIG)*RB2*1.0E-8/CVV
      CVV0=CVV
      CCI(8)=AAI
      CALL COVAX(SM,IS)
      CALL COVBX(SM,LF,IS)
      CALL COVCX(SM,CVV,IS,LF)
      IF (.NOT.LINVAR.AND.ABS(CVV-VARDG2).GT.0.1)
     *WRITE(6,7464)CVV,VARDG2
      IF (LINVAR) VARDG2=CVV
 7464 FORMAT(' ** WARNING ** CVV,VARGD2= ',2E15.8)
```



```
C
C THE DEG.VAR. OF THE COVARIANCE FUNCTION OF THE ANOMALOUS POTENTIAL
C ARE STORED IN THE FIRST PART OF SIGMA (SUBSCRIPT 1 TO IMAX1R) FOR
C COLLOCATION I AND IN THE LAST PART (SUBSCRIPT IS=IMAX1R+3 TO
C IS+IMAX1) FOR COLLOCATION II.
C
 1110 FORMAT(/' RATIO R/RE                =   ',F9.6,/
     *' DEPTH TO BJERHAMMAR SPHERE (R-RE)     = ',F10.2,' M'/
     *' VARIANCE OF POINT GRAVITY ANOMALIES    = ',F10.2,' MGAL**2'/
     *' THE FACTOR A, DIVIDED BY RE**2 IS      = ',F10.2,' MGAL**2')
      A0 = AAI*1.0E10/RE**2
      IF (LTEST) WRITE(*,*)' AAI, A0 ',AAI,A0
      IF (AAI.LT.D0) THEN
       WRITE(*,*)' AAI < 0 ',AAI
       STOP
      END IF
      WRITE(6,1110)R,DR,VARDG2,A0
      IF (MODE.EQ.4) GO TO 991
C
C --------------- INPUT (3D) ---------------------------------
C
      IF (MODE.EQ.1.OR.MODE.GE.5) GO TO 201
C
      IF (LTABHX) GO TO 200
C
C INPUT OF PARAMETERS DEFINING THE TABLE USED FOR FAST COMPUTATION
C OF COVARIANCES. MAXB(1) NUMBER OF STEPS IN HEIGHT, SZ(1) MINIMUM
C HEIGHT (M), (GENERALLY ZERO), SZ(2) MAXIMAL HEIGHT (M),
C MAXB(2) NUMBER OF INTERVALS WITH EQUIDISTANT STEPSIZE IN PSI. SZ(3)
C IS ZERO AS STARTING INTERVAL END-POINT, SZ(I+3) IS RIGHTMOST END-
C POINT OF I'TH INTERVAL. MAXB(I+6) IS NUMBER OF EQUIDISTANT INTER-
C VALS IN I'TH INTERVAL.
C PSI IS THE SPHERICAL DISTANCE.
C A TABLE WILL BE GENERATED BY CTABEL, WHICH MAKES A FAST COMPUTATION
C OF COVARIANCES OF TYPE (1,1), (1,6), (1,7), (3,3), (3,6), (3,7)
C POSSIBLE.
      IF (LINTER)
     *WRITE(*,*)' INPUT NO. STEPS IN HEIGHT, HMIN, HMAX, ID STEPS'
      READ(5,*)MAXB(1),SZ(1),SZ(2),MAXB(2)
      NT=MAXB(2)
C NT+5 MUST BE LESS THAN 31, (DIMENSION OF SZ).
C  50 FORMAT(I4,2F10.3,I4)
      IF (LINTER) WRITE(*,*)' INPUT INTERVAL END POINTS'
      READ(5,*)(SZ(I+3),I=1,NT)
C 51 FORMAT(6F10.3)
      IF (LINTER)
     *WRITE(*,*)' INPUT NUMBER OF SUB-INTERVALS IN EACH INTERVAL'
      READ(5,*)(MAXB(I+6),I=1,NT)
   56 FORMAT(6I10)
      SZ(3)=D0
      NT1=1
      NTMAX1=798
      NTMAX=1600
C NTMAX IS CURRENT MAXIMUM OF POINTS IN TABEL, I.E. NTMAX*8 IS
C THE DIMENSION OF CT. NTMAX1 IS THE UPPER LIMIT FOR NUMBER OF
C ENTRIES RELATED TO SPHERICAL DISTANCE (PSI), I.E. THE DIMENSION
C OF CTTF (COMMON BLOCK TABELC).
C
      DO I=1,NT
       NT1=NT1+MAXB(I+6)
      END DO
      IF (NT1.GT.NTMAX) THEN
       WRITE(6,53)NT1
   53  FORMAT(' *** TOO LARGE TABEL REQUIRED *** NT1=',I5)
       STOP
      END IF
      WRITE(6,52)MAXB(1),SZ(1),SZ(2)
   52 FORMAT(/' TABEL OF COVARIANCES GENERATED USING',/,
     *' NS=',I4,', HMIN=',F10.3,', HMAX=',F10.3,
```

```
     */,' MAX-PSI N-INTERVALS.')
      DO 198 I=1,NT
  198 WRITE(6,55)SZ(I+3),MAXB(I+6)
   55 FORMAT(1X,F10.3,I10)
C
      IX(4)=0
C IX(4) POINTS AT THE ZERO'TH SUBSCRIPT OF SIGMA, WHICH HERE IS 0.
      CALL CTABEL(0,LTEST)
C
C     IF (LTEST) CALL TEBICU
C SEE TEBICU FOR INSTRUCTIONS CONCERNING INPUT DATA TO THE SUB-
C ROUTINE WHICH TESTS BSFC AND BILDEC.
C
      GO TO 201
C
C ------------------ INPUT (3E) ------------------------------
C INPUT OF CONSTANTS SPECIFYING TABLES USED WITH FIXED HEIGHTS
C AND FUNCTIONALS. NHE IS NUMBER OF HEIGHTS (MAX 5), NINTH THE
C NUMBER OF FUNCTIONALS AND SIZEI THE INTERVAL LENGTH IN ARCSECONDS.
C
  200 IF (LINTER) WRITE(*,*)' INPUT NHE, NINTH, SIZEI'
      READ(5,*)NHE,NINTH,SIZEI
C  41 FORMAT(2I4,F8.1)
      IF (LINTER) WRITE(6,42)NINTH,SIZEI
   42 FORMAT(' NUMBER OF INTERVALS=',I4,', SIZE =',F9.1,' ARCSEC.',
     */,' HEIGHT (M)  FUNCTIONAL TYPES')
      IF (NINTH.GT.499) WRITE(6,43)
   43 FORMAT(' *** WARNING ***  NINTH DECREASED TO 499')
      IF (NINTH.GT.499) NINTH=499
      DO 44 I=1,NHE
      IF (LINTER) WRITE(*,*)' INPUT H AND NUMBER OF FUNCTIONALS'
      READ(5,*)HTA(I),NFU(I)
C INPUT OF HEIGHT IN METERS AND NUMBER OF FIXED FUNCTIONALS.
C  45 FORMAT(F9.1,I3)
      IF (LINTER) WRITE(*,*)' INPUT FUNCTIONAL TYPES'
      READ(5,*)(KEYH(J,I),J=1,NFU(I))
C INPUT OF FUNCTIONAL TYPES.
      WRITE(6,47)HTA(I),(KEYH(J,I),J=1,NFU(I))
   47 FORMAT(F10.1,5I4)
   44 CONTINUE
C  46 FORMAT(5I3)
      CALL INTABH(SM,0,LTEST)
C
  201 KPP=0
      KQQ=0
      LDEN=LF
C
C ********************** INPUT (4) ******************************
C
  102 IF (LINTER) WRITE(*,*)
     *' COVARIANCES IN GRID OR PROFILE & LAST ? (T/F)'
      READ(*,*)LGRIDO,LAST2
      IF (.NOT.LGRIDO) GO TO 892
C
C ********************** INPUT (4a) ******************************
C
       IF (LINTER)
     * WRITE(*,*)' INPUT GRID LABEL '
       READ(*,*)RLAMIN,RLAMAX,RLOMIN,RLOMAX,DLAT,DLON
       NLAT=((RLAMAX-RLAMIN)/DLAT+1)
       NLON=((RLOMAX-RLOMIN)/DLON+1)
       NCOV=((RLAMAX-RLAMIN)/DLAT+1)*((RLOMAX-RLOMIN)/DLON+1)
c      write(*,*)' NLAT,NLON,NCOV= ',NLAT,NLON,NCOV
      GO TO 907

C ********************** INPUT (4b) ******************************
C
C INPUT OF QUANTITIES SPECIFYING THE OUTPUT TABLE. THE TABLE WILL CON-
C TAIN MT COLUMNS OF COVARIANCES OF KINDS KP, KQ (TO BE INPUT SUBSEQUE-
```



```
C NTLY) COMPUTED IN POINTS P AND Q. EACH COLUMN WILL CONTAIN NCOV VALUES
C CORRESPONDING TO Q MOVING IN AN AZIMUTH (ALFA) IN STEPS OF LENGTH DT
C (MINUTES). THE SPECIFICATION CONSIST OF NCOV, DT, ALFA IN DEG.,MIN.,
C SEC AND A LOGICAL, LAST2, WHICH IS TRUE WHEN THIS IS THE LAST SPECI-
C FICATION FOR THE DEGREE-VARIANCE MODEL UNDER CONSIDERATION.
  892 IF (LINTER) THEN
       WRITE(*,*)
     * ' INPUT NUMBER OF VALUES, DISTANCE (DEC: DEG:), AZIMUTH'
       WRITE(*,*) ' (DD MM SS.S) AND T IF LAST INPUT '
      END IF
      READ(5,*)NCOV,DT,IDEG,MIN,SEC,LAST2
C  14 FORMAT(I5,F7.2,I5,I3,F6.2,L2)
      CALL RAD(IDEG,MIN,SEC,ALFA,1)
      RT = DT*DEGRAD
      LPOLE =  ABS(ALFA).LT.1.0E-6.OR. ABS(ALFA-PI).LT.1.0E-6
      IF (LPOLE) GO TO 103
C
      CA = COS(ALFA)
      SA = SIN(ALFA)
  103 IF (LTEST) WRITE(6,15)NCOV,DT,IDEG,MIN,SEC,ALFA
   15 FORMAT(/' NCOV,DT,DEG,MIN,SEC,ALFA=',I4,F6.2,I5,I3,F6.2,E14.6)
C
      MV = 0
      MT = 1
C
      IF (NCOV.GT.181)  THEN
       NCOV = 181
       WRITE(6,37)
   37  FORMAT(' NCOV TOO BIG, FIXED TO 181.')
      END IF
C
C -------------------------- INPUT (4C) -------------------------
C
      IF (LFINI) THEN
C FINITE COVARIANCE FUNCTION IN USE. CHANGED 2004-02-28.
       WRITE(*,*)' FINITE COVARIANCE FUNCTION IN USE ! '
       WRITE(*,*)' INPUT FINITE COVARIANCE FUNCTION MODEL NO '
       WRITE(*,*)' (1) SANSO/SCHU (2) MOREAUX  '
       READ(*,*)MODEFI
       WRITE(*,*)' MODEL ',MODEFI
C
       LFINOU=MODE.EQ.5
       IF (LFINOU) THEN
        WRITE(*,*)' OUTPUT FINITE COV.FCT. '
       ELSE
        WRITE(*,*)' OUTPUT DIFFERENCE FINITE - ANALYTIC COV.FCT. '
       END IF
C
       IF (MODEFI.EQ.2) THEN
        WRITE(*,*)' INPUT NAME OF FILE WITH LEGENDRE COEFF. '
        READ(*,*)FINAME
        WRITE(*,*)' LEGENDRE COEFF. INPUT FROM FILE ',FINAME
        OPEN(77,FILE=FINAME)
        WRITE(*,*)' INPUT MAXIMAL DEGREE '
        READ(*,*)MAXFI
        IF (MAXFI.GT.MAXSM) THEN
         MAXFI=MAXSM
         WRITE(*,*)' MAX SET TO ',MAXSM
        END IF
        WRITE(*,*)' MAXIMAL DEGREE USED ',MAXFI
        READ(77,*)(SIGMAF(K),K=1,MAXFI+1)
        WRITE(*,*)' LEGENDRE C. ASSOCIATED WITH HEIGHT '
        READ(*,*)RLCH
        WRITE(*,*)' AND FUNCTIONAL TYPE (1, 2, 3, 5 ONLY ) '
        READ(*,*)IFT
        WRITE(*,*)' HEIGHT AND TYPE ',RLCH,IFT
        IF (IFT.EQ.1) T2=1.0D0
        IF (IFT.EQ.2.OR.IFT.EQ.3) T2=2.0D0
        IF (IFT.EQ.3) T2=3.0D0
```

```
       T1=D1
       RDD=((RE+RLCH)/RE)**2
       SCFACT=D1
       CLOSE(77)
      END IF
     END IF
C
  104 MT = MT+1
     IF (MV.NE.0.AND.MV.NE.3) GO TO 112
C
C ************************* INPUT (5) ***************************
C
C INPUT OF INTEGERS KP AND KQ SIGNIFYING THE KIND OF QUANTITIES BETWEEN
C WHICH WE WANT TO COMPUTE THE COVARIANCES. THE VALUES OF KP,KQ MUST BE
C EQUAL TO THE EQUATION NUMBERS OF REF(A), WHICH DEFINES THE QUANTITIES
C (1) - (9), (12) AND (14). THE VALUES 10, 11, 13, 15 CORRESPOND TO
C THE QUANTITIES GIVEN IN REF (C), EQ. (3) - (6). THE VALUES 0 AND
C 16 CORRESPOND TO DENSITY ANOMALIES.
C THE VALUE 17 CORRESPONDS TO A SPHERICAL HARMONIC COEFFICIENT.
C ON THE SAME LINE INPUT OF THE HEIGHTS OF P AND Q AS WELL,(IN
C METERS), (0.0 FOR KIND 17),
C AND OF A LOGICAL VARIABLE LAST3, TRUE WHEN THIS KIND OF COVA-
C RIANCES ARE THE LAST ONES TO BE COMPUTED WITH THE CHOOSED FORM OF THE
C TABLE. THREE SETS OF VALUES MAY BE OUTPUT ON ONE LINE , CF. FORMAT
C STATEMENT 15
  907 IF (LINTER ) THEN
       WRITE(*,*)' 3 TIMES (SOME MAY BE DUMMY): TYPE1,TYPE2'
       WRITE(*,*)
     * ' HEIGHT1, HEIGHT2, LAST (T/F), WITH TYPE=1..25, HEIGHT IN M'
       WRITE(*,*)
     * ' NEGATIVE TYPE, INDICATES USE OF ROTATED SYSTEM.'
      END IF
      READ(5,*,END=112)(KX(K),KY(K),IP(K),IQ(K),LA(K),K = 1, 3)
C  16 FORMAT(3(2I3,2I8,L2))
      MV = 0
  112 MV = MV+1
      MG=1
      KP(MT) = KX(MV)
      KQ(MT) = KY(MV)
      HP(MT) = IP(MV)
      HQ(MT) = IQ(MV)
      LAST3 = LA(MV)
      LMEAP = KP(MT).GT.100
      LMEAQ = KQ(MT).GT.100
      IF (MODE.GT.5.AND.MODEFI.EQ.2) RDD=RE**2/((RE+HP(MT))*(RE+HQ(MT)))
      IF (LGRIDO) THEN
       WRITE(*,908)RLAMIN,RLAMAX,RLOMIN,RLOMAX,DLAT,DLON,
     * HP(MT),HQ(MT),KP(MT),KQ(MT)
 908   FORMAT(' GRID LABEL ',/,4F10.4,2F9.5,2F9.1,2I3)
       IF (LPUNCH) WRITE(7,902)RLAMIN,RLAMAX,RLOMIN,RLOMAX,DLAT,DLON,
     * HP(MT),HQ(MT),KP(MT),KQ(MT)
 902   FORMAT(4F10.4,2F9.5,2F9.1,2I3)
      END IF
c
      IF (LMEAP) THEN
C --------------- INPUT (5A) ---------------------------------
       IF (LINTER)WRITE(6,1117)
 1117  FORMAT(' INPUT PARAMETERS DEFINING TYPE AND SIZE OF MEAN VALUE',/
     * ' LEQANG- TRUE IF EQUAL ANGULAR OR 1-D BLOCK',/
     * ' BLOCK SIDE LENGTH IN LAT. & LONG. (MIN) OR (LENGTH .0 IF 1D)',/
     * ' LATITUDE OF TOTAL AREA MEAN')
       READ(5,*)LEQANG,BSIZEN,BSIZEE,RLATPP
C LMEAP1 IS TRUE IF MEAN VALUES ARE 1D, ALONG A SATELLITE OR
C AIRCRAFT TRACK, FOR EXAMPLE BUT IS NOT MEANINGFULL FOR A GRID.
       LMEAP1=ABS(BSIZEE).LT.1.0D-8
       IF (LMEAP1) THEN
        LFILTE=LF
C TRACK-AZIMUTH IS FORCED TO 0.0.
        SAZP=D0
```



```
        CAZP=D1
        CALL MEAN1(FILTER,NFILTE,SAZP,CAZP,LFILTE,LF,LINTER)
       END IF
C
       IF (LEQANG) WRITE(6,232)BSIZEN,BSIZEE
  232  FORMAT(/' THE FOLLOWING QUANTITIES ARE MEAN VALUES, WITH',/,
     * ' BLOCKSIZE=',F10.2,' * ',F10.2,' MINUTES')
       IF ((.NOT.LEQANG).AND.(.NOT.LMEAP1))
     * WRITE(6,233)BSIZEN
  233  FORMAT(/' THE FOLLOWING QUANTITIES ARE EQUAL-AREA MEAN',
     * ' VALUES, WITH BLOCK-SIZE=',F10.2,' MINUTES')
       IF (LMEAP1) WRITE(6,236)BSIZEN
  236  FORMAT(' THE QUANTITIES ARE 1-D MEANS OVER A ',F9.4,
     * ' ARCMIN TRACK SEGMENT ')
       RLATPP=RLATPP*3600/RADSEC
       COSLA=COS(RLATPP)
       SINLA=SIN(RLATPP)
       BSIZEN=BSIZEN*60/RADSEC
       IF (.NOT.LEQANG)BSIZEE=D0
       BSIZEE=BSIZEE*60/RADSEC
       IF (LMEAP1) THEN
        NSTEP=NFILTE
       ELSE
        NSTEP=5
       END IF
       NSTEPE=1
       STEPE=D0
       COSSTE=D1
       SINSTE=D0
C THE CALL TO ICMEAN GIVES STEPSIZE AND COS,SIN.
       CALL ICMEAN(BSIZEN,STEPN,NSTEP,COSSTN,SINSTN,D1,D0,LT,LMEAP1)
       SHIFTS=STEPN*(NSTEP-1)/2
       COST2P=COS(SHIFTS)
       SINT2P=SIN(SHIFTS)
       IF (LMEAP1) THEN
C LMEAP1 INTRODUCED 1992.10.07 BY CCT.
        BSIZEN=-BSIZEN
C WE USE THE SWITCH OF SIGN AS AN INDICATOR OF 1D MEAN.
       ELSE
        NSTEPE=5
        IF (LEQANG) THEN
         CALL ICMEAN(BSIZEE,STEPE,NSTEPE,COSSTE,SINSTE,D1,D0,LT,LF)
        ELSE
         BSIZEA= COS(RLATPP)*BSIZEE
         CALL ICMEAN(BSIZEA,STEPE,NSTEPE,COSSTE,SINSTE,D1,D0,LF,LF)
        END IF
       END IF
C
      ELSE
       NSTEP=1
       NSTEPE=1
       COST2P=D1
       SINT2P=D0
      END IF
c
      IF (LMEAQ) THEN
C --------------- INPUT (5B) ---------------------------------
       IF (LINTER)WRITE(6,1117)
       READ(5,*)LEQANQ,BSIZEN,BSIZEE,RLATP
C LMEAQ1 IS TRUE IF MEAN VALUES ARE 1D, ALONG A SATELLITE OR
C AIRCRAFT TRACK, FOR EXAMPLE BUT IS NOT MEANINGFULL FOR A GRID.
       LMEAQ1=ABS(BSIZEE).LT.1.0D-8
       IF (LMEAQ1.AND.(.NOT.LMEAP1)) THEN
        LFILTE=LF
        CALL MEAN1(FILTER,NFILTE,SAZQ,CAZQ,LFILTE,LF,LINTER)
C TRACK-AZIMUTH IS FORCED TO 0.0.
        SAZQ=D0
        CAZQ=D1
       END IF
```

```
C
       IF (LEQANQ) WRITE(6,232)BSIZEN,BSIZEE
       IF ((.NOT.LEQANQ).AND.(.NOT.LMEAQ1))
     * WRITE(6,233)BSIZEN
       IF (LMEAQ1) WRITE(6,236)BSIZEN
       RLATP=RLATP*3600/RADSEC
       COSLA=COS(RLATP)
       SINLA=SIN(RLATP)
       BSIZEN=BSIZEN*60/RADSEC
       IF (.NOT.LEQANQ)BSIZEE=D0
       BSIZEE=BSIZEE*60/RADSEC
       IF (LMEAQ1) THEN
        NSTEQ=NFILTE
       ELSE
        NSTEQ=5
       END IF
       NSTEQE=1
       STEQE=D0
       COSSQE=D1
       SINSQE=D0
C THE CALL TO ICMEAN GIVES STEPSIZE AND COS,SIN.
       CALL ICMEAN(BSIZEN,STEQN,NSTEQ,COSSQN,SINSQN,D1,D0,LT,LMEAQ1)
       SHIFTS=STEQN*(NSTEQ-1)/2
       COST2Q=COS(SHIFTS)
       SINT2Q=SIN(SHIFTS)
C CHANGE 2004-02-26.
       IF (LMEAQ1) THEN
C LMEAN1 INTRODUCED 1992.10.07 BY CCT.
        BSIZEN=-BSIZEN
C WE USE THE SWITCH OF SIGN AS AN INDICATOR OF 1D MEAN.
       ELSE
        NSTEQE=5
        IF (LEQANQ) THEN
         CALL ICMEAN(BSIZEN,STEQE,NSTEQE,COSSQE,SINSQE,D1,D0,LT,LF)
        ELSE
         BSIZEA= COS(RLATP)*BSIZEE
         CALL ICMEAN(BSIZEA,STEQE,NSTEQE,COSSQE,SINSQE,D1,D0,LF,LF)
        END IF
       END IF
C
      ELSE
       NSTEQ=1
       NSTEQE=1
       COST2Q=D1
       SINT2Q=D0
      END IF
c
      LSAT=KP(MT).LT.0.OR.KQ(MT).LE.0
      IF (LSAT) WRITE(6,*)' ROTATED FRAME USED.'
      LTESTS=LSAT.AND.LTEST
C
      LNEW = KP(MT).NE.KPP.OR.KQ(MT).NE.KQQ
      IF (KP(MT).GT.100) THEN
       KPP = KP(MT)-100
      ELSE
       KPP = KP(MT)
      END IF
      IF (KQ(MT).GT.100) THEN
       KQQ = KQ(MT)-100
      ELSE
       KQQ = KQ(MT)
      END IF
      KI(6) = ABS(KPP)
      KI(7) = ABS(KQQ)
      LDENP=KP(MT).EQ.0
      LDENQ=KQ(MT).EQ.0
      IF (.NOT.(LDENP.OR.LDENQ).OR.LDEN) GO TO 302
      LDEN=LT
C
```



```
C ------------------------ INPUT (5C) --------------------------------
C
C IF COVARIANCE INVOLVING DENSITY ANOMALY IS TO BE EVALUATED, INPUT
C OF POWER OF RADIAL WEIGHT-FACTOR, IEHD. IF IT IS ZERO, HARMONIC
C DENSITY ANOAMLIES ARE USED.
      IF (LINTER) WRITE(*,*)' INPUT DENSITY FUNCTIONAL SPEC.'
      READ(5,1102)IEHD
      KI(32)=IEHD-1
      CCI(15)=1.5+IEHD/2.0
      CCI(13)=10.0/(PI*6.67E-8*RE**(3-IEHD))
      IF (LINTER) WRITE(6,303)IEHD
  303 FORMAT(/,' WEIGHT FUNCTION ON DENSITY IS R**(',I2,')',/)
C
  302 IF (LDENP) HP(MT)=RE*HP(MT)/(RE-HP(MT))
      IF (LDENQ) HQ(MT)=RE*HQ(MT)/(RE-HQ(MT))
C
C ------------------------ INPUT (5D) --------------------------------
C
C INPUT OF DIRECTION VECTOR ANGLES (AZIMUTH, HEIGHT ANGLE, DEGREES).
C
      IF (LSAT) THEN
C CHANGE 2003-04-01, CORRECTION 2004-02-28.
       IF (ABS(KP(MT)).EQ.1.OR.ABS(KP(MT)).EQ.3) THEN
c NO ROTATION NEEDED:
        SROT(1,1)=D1
        SROT(2,2)=D1
        SROT(3,3)=D1
        SROT(1,2)=D0
        SROT(2,1)=D0
        SROT(3,2)=D0
        SROT(2,3)=D0
        SROT(1,3)=D0
        SROT(3,1)=D0
       ELSE
C WE NEED THE ROTATION ANGLES IN DEGREES: AZIMUTH, HEIGHT ANGLE AND
C ROLL. THEY ARE REGARDED AS ROTATIONS ABOUT THE 3, 2 AND 1 AXES,
C RESPECTIVELY. AZIM: FROM NORTH TOWARD EAST, HEIANG ANTI-CLOCKWISE,
C AND THAU CLOCKWISE (GENERALLY ZERO).
        IF (LINTER)
     *  WRITE(*,*)' INPUT SYSTEM ROTATION ANGLES (DEG.), AZ,BET,TAU '
        READ(5,*)AZIM,HEIANG,THAU
        WRITE(6,312)AZIM,HEIANG,THAU
  312   FORMAT(' ROTATION ANGLES IN DEG.',3F9.4)
        DEGRAD=PI/180.0D0
        AZIMR=PI/D2-AZIM*DEGRAD
        HEIAR=-HEIANG*DEGRAD
        THAUR=THAU*DEGRAD
        SA3=SIN(AZIMR)
        CA3=COS(AZIMR)
        SA2=SIN(HEIAR)
        CA2=COS(HEIAR)
        SA1=SIN(THAUR)
        CA1=COS(THAUR)
        SROT(1,1)=CA2*CA3
        SROT(2,1)=CA1*SA3+SA1*SA2*CA3
        SROT(3,1)=-SA1*SA3+CA1*SA2*CA3
        SROT(1,2)=-CA2*SA3
        SROT(2,2)=CA1*CA3-SA1*SA2*SA3
        SROT(3,2)=-SA1*CA3-CA1*SA2*SA3
        SROT(1,3)=-SA2
        SROT(2,3)=SA1*CA2
        SROT(3,3)=CA1*CA2
       END IF
       IF (LTEST) WRITE(*,901)SROT
  901  FORMAT(' SROT',6F10.7)
      END IF
c
C ------------------------ INPUT (5E) --------------------------------
C
```

```
      IF (KP(MT).EQ.17.OR.KQ(MT).EQ.17) then
       WRITE(*,*)' INPUT DEGREE AND ORDER OF SP.HARM. '
       READ(*,*)IIDEG,JJDEG
       WRITE(*,921)IIDEG,JJDEG
 921   FORMAT(' IIDEG, JJDEG= ',2I3)
      END IF
C
C COMPUTATION OF CONSTANTS NEEDED FOR THE COVARIANCE COMPUTATION, WHICH
C ARE INDEPENDENT OF T AND THE HEIGHTS BY THE CALL OF COVBX.
      IF (LNEW) CALL COVBX(SM,LSAT,0)
C
      IF (LTEST.AND.LNEW)
     *WRITE(6,17)(CCI(K),K=1,7),(KI(K),K=6,25),(SIGMA(K),K=1,NC1)
   17 FORMAT(/' CI:',7E11.4,/,' KI:',20I3,/,' SI:',5E11.4,/,59(4X,5E11.4/
     *))
C
      IF (MODE.GE.5.AND.MODEFI.EQ.1) THEN
C SANSO/SCHUH FINITE COVARIANCE FUNCTION IN USE.
C WE MUST DETERMINE COVARIANCE FUNCTION PARAMETERS
C WE NOW DETERMINE THE VALUE OF THE CORRELATION DISTANCE.
       CR(1)=D1
       CR(2) = HP(MT)
       CR(3) = HQ(MT)
       CR(4) = D0
       CR(5) = D0
       CR(6) = D1
       CR(7) = D1
       CR(8) = D0
       CR(9) = D1
       CR(10) = GM/(RE+HP(MT))**2
       CR(10) = GM/(RE+HP(MT))**2
       CR(11) = GM/(RE+HQ(MT))**2
       CALL COVCX(SM,CVV,0,.FALSE.)
       C0=CVV
       PSI=D0
       C1=C0
       DPSI=5.0D-4
       TEST3=D1
 3120  PSI= PSI+DPSI
       C2=c1
       CR(1)=COS(PSI)
       CR(4) = SIN(PSI)
       CR(5) = D0
       CR(6) = CR(1)
       CR(7) = D1
       CR(8) = D0
       CR(9) = D1
       CALL COVCX(SM,CVV,0,LF)
       C1=CVV
       TEST1=TEST3
       TEST3=C1/C0
       IF (TEST3.GT.1.0D0.OR.TEST3.LT.D0)
     * WRITE(*,*)' WARNING ',C0,C2,C1
       IF (TEST3.GT.0.5D0) GO TO 3120
       DPSI=DPSI/D2
       PSI=PSI-DPSI
       KPSI=0
 3121  CR(1)=COS(PSI)
       CALL COVCX(SM,CVV,0,LF)
       C3=CVV
       KPSI=KPSI+1
       TEST2=C3/C0
       DPSI=DPSI/D2
       IF (TEST2.GT.0.5D0.AND.TEST3.LT.0.5D0) THEN
        TEST1=TEST2
        PSI=PSI+DPSI
       ELSE
        TEST3=TEST2
        PSI=PSI-DPSI
```



```
       END IF
       IF (KPSI.LT.115.AND.ABS(0.5D0-TEST2).GT.1.0D-8) GO TO 3121
       PSI1=PSI
       RDD= FINDR(PSI1,1)
       WRITE(*,3122)C0,PSI1/DEGRAD,RDD,KPSI
 3122  FORMAT(' C0= ',F12.4,' CORREL.DIST.',F10.5,' R ',F10.5,
     * ' ITER= ',I5)
       SCFACT=C0/COZERO(D0,RDD,1)
      ELSE
       IF (MODE.GE.5.AND.MODEFI.EQ.2) THEN
C CHECK IF DATA KIND IS PERMITTED TYPE (NOT YET IMPLEMENTED).
       END IF
      END IF
C
      DO 120 M = 1, NCOV
C
      RLATP=D0
      RLONGP=D0
      COSLAP = D1
      SINLAP = D0
      COSLOP= D1
      SINLOP= D0
c
      IF (LGRIDO) THEN
       NLAT1=(M-1)/NLON
       NLON1=MOD((M-1),NLON)
       RLONQ=RLOMIN+NLON1*DLON
       RLATQ=RLAMAX-NLAT1*DLAT
c      write(*,*)' lat,lon,nlat1,nlon1 ',Rlatq,rlonq,nlat1,nlon1
       RLATQ=RLATQ*DEGRAD
       RLONQ=RLONQ*DEGRAD
       CQ=COS(RLATQ)
       SQ=SIN(RLATQ)
       CD=COS(RLONQ)
       SD=SIN(RLONQ)
       T =(CQ*COSLAP*CD)
       RV =ACOS(T)
c      WRITE(*,*)' RV= ',RV/DEGRAD
      ELSE
       IF (MT.EQ.2) COV(M,1) = (M-1)*RT
       RV = (M-1)*RT
       T = COS(RV)
C RV IS EQUAL TO THE SPHERICAL DISTANCE BETWEEN P AND Q IN UNITS OF RAD-
C IANS.
       U = SIN(RV)
       IF (.NOT.LPOLE) THEN
        SQ = U*CA
        CQ = SQRT(D1-SQ *SQ)
        SD = U*SA/CQ
        CD = T/CQ
       ELSE
C
        SD = D0
        CD = D1
        IF (RV.GT.PI/D2) CD = -D1
        CQ = T
        I = 1
        IF (ALFA.LT.D0) I = -1
        SQ = U*I
       END IF
      END IF
C
      IF (LMEAP) THEN
       IF (.NOT.LMEAP1) RLATP=RLATP+STEPN*D2
       COSLAP =  COS(RLATP)
       SINLAP =  SIN(RLATP)
       IF (LMEAP1) THEN
        CALL PAZIM(RLATP,RLONGP,COSLAP,SINLAP,COSLOP,SINLOP,
     *  -CAZP,-SAZP,COST2P,SINT2P,LTEST)
```

```
       ELSE
         IF (.NOT.LEQANG)CALL
     *   ICMEAN(BSIZEN,STEPE,NSTEPE,COSSTE,SINSTE,COSLAP,SINLAP,LF,LF)
         RLONGP=RLONGP-STEPE*D2
       END IF
       COSLOP = COS(RLONGP)
       SINLOP = SIN(RLONGP)
      END IF
C
      RLATQ=ATAN2(SQ,CQ)
CHANGE 1999-02-16.
C     RLONGQ=D0
      RLONGQ=ATAN2(SD,CD)
      COSLAQ = CQ
      SINLAQ = SQ
      COSLOQ= CD
      SINLOQ= SD
      IF (LMEAQ) THEN
       IF (.NOT.LMEAQ1) RLATQ=RLATQ+STEPN*D2
       COSLAQ =  COS(RLATQ)
       SINLAQ =  SIN(RLATQ)
       IF (LMEAQ1) THEN
        CALL PAZIM(RLATQ,RLONGQ,COSLAQ,SINLAQ,COSLOQ,SINLOQ,
     *  -CAZP,-SAZP,COST2Q,SINT2Q,LTEST)
       ELSE
        IF (.NOT.LEQANQ)CALL
     *  ICMEAN(BSIZEN,STEQE,NSTEQE,COSSQE,SINSQE,COSLAQ,SINLAQ,LF,LF)
        RLONGQ=RLONGQ-STEPE*D2
       END IF
       COSLOQ = COS(RLONGQ)
       SINLOQ = SIN(RLONGQ)
       CQ=COSLAQ
       SQ=SINLAQ
      END IF
C
C Change 1999-02-15 to transfer longitude info. to covcx.
      CLOP=COSLOP
      SLOP=SINLOP
      CLOQ=COSLOQ
      SLOQ=SINLOQ
C
C TRANSFER OF COORDINATE INFORMATION TO THE SUBROUTINE ACCORDING TO
C THE SPECIFICATIONS GIVEN IN THE SUBROUTINE.
      CR(1) = T
      CR(2) = HP(MT)
      CR(3) = HQ(MT)
C TWO ERRORS HERE 1999-02-16.
C SUSPECTED ERROR 2001-11-02 !
      CR(4) = SINLAP
c     CR(4) = SINLOP
      CR(5) = SQ
      CR(6) = COSLAP
c     CR(6) = COSLOP
      CR(7) = CQ
      CR(8) = SD
      CR(9) = CD
      CR(10) = GM/(RE+HP(MT))**2
      CR(11) = GM/(RE+HQ(MT))**2
      IF (LTEST)WRITE(6,18)SQ,CQ,SD,CD
   18 FORMAT(/' SQ,CQ,SD,CD=',6E12.5)
C
      LTABH=.FALSE.
      IF (MODE.NE.2.AND.MODE.NE.5) THEN
       IF (LMEAP.OR.LMEAQ) THEN
        COVXX = COMEAN(SM,0,0,COSLAP,SINLAP,COSLOP,SINLOP,
     *  COSLAQ,SINLAQ,COSLOQ,SINLOQ,NSTEP,NSTEQ,LF,LCZERO,LF)
       ELSE
        CALL COVCX(SM,COVXX,0,LSAT)
       END IF
```



```
       IF (LGRIDO) THEN
        COV(1,MG)=COVXX
        MG=MG+1
       ELSE
        COV(M,MT)=COVXX
       END IF
C
      END IF
C
      IF (LSAT) THEN
       CALL COVROT(SROT,SROT)
       IF (LTEST) WRITE(*,906)COVX
  906  FORMAT(3E15.7)
       KPZ=ABS(KPP)
       KQZ=ABS(KQQ)
       COV(M,MT)=
     * COVX(KSAT(KPZ,1),KSAT(KPZ,2),KSAT(KQZ,1),KSAT(KQZ,2))
       IF (KPZ.EQ.15.AND.KQZ.NE.15)
     * COV(M,MT)=COV(M,MT)-COVX(2,2,KSAT(KQZ,1),KSAT(KQZ,2))
       IF (KPZ.NE.15.AND.KQZ.EQ.15)
     * COV(M,MT)=COV(M,MT)-COVX(KSAT(KPZ,1),KSAT(KPZ,2),2,2)
       IF (KPZ.EQ.15.AND.KQZ.EQ.15)
     * COV(M,MT)=COV(M,MT)-COVX(1,1,2,2)-COVX(2,2,1,1)+COVX(2,2,2,2)
       COV(M,MT)=COV(M,MT)*CFA
      END IF
C
      IF (MODE.NE.1.AND.(.NOT.LTABH).AND.MODE.NE.5)
     *COX=COVCG(SM,0,LTEST)
      IF (MODE.EQ.5.OR.MODE.EQ.6) THEN
       PSI = ACOS(T)
       COX=SCFACT*COZERO(PSI,RDD,MODEFI)
      END IF
      LTABH=LTABHX
      IF (MODE.NE.1.AND.LTABHX) THEN
       IF (LMEAP.OR.LMEAQ) THEN
        COX = COMEAN(SM,0,0,COSLAP,SINLAP,COSLOP,SINLOP,
     *  COSLAQ,SINLAQ,COSLOQ,SINLOQ,NSTEP,NSTEQ,LTABH,LCZERO,LF)
       ELSE
        CALL COVCX(SM,COX,0,LSAT)
       END IF
      END IF
      IF (MODE.EQ.2.OR.MODE.EQ.5.OR.(MODE.EQ.6.AND.LFINOU))
     * COV(M,MT)=COX
      IF (MODE.EQ.3.OR.(MODE.EQ.6.AND.(.NOT.LFINOU)))
     *COV(M,MT)=COV(M,MT)-COX
C
      IF (LGRIDO.AND.MG.EQ.7) THEN
       IF (LPUNCH) WRITE(7,889) (COV(1,MM),MM=1,6)
       WRITE(*,889)(COV(1,MM),MM=1,6)
       MG=1
      END IF
C
      IF (LTEST) THEN
       KK = KI(8)+1
       WRITE(6,19)((CR(I*8+K+3),K=1,8),I=1,KK),
     * (CR(I+50),I=1,ND2)
   19  FORMAT(' CR:',8E11.4,/,5(4X,8E11.4,/))
      END IF
  120 CONTINUE
C
      IF (LGRIDO.AND.MG.NE.1) THEN
       WRITE(*,889)(COV(1,MM),MM=1,(MG-1))
       IF (LPUNCH) WRITE(7,889) (COV(1,MM),MM=1,(MG-1))
  889  FORMAT(' ',6E12.5)
       MG=1
      END IF
      IF (LGRIDO) GO TO 887
C
      IF ((.NOT.(LAST3.OR.MT.EQ.7)).OR.LGRIDO) GO TO 104
```

```
C
C OUTPUT OF A TABLE OF COVARIANCES.
C     WRITE(7,30)
      WRITE(6,30)
   30 FORMAT(' ')
      WRITE(6,20)IDEG,MIN,SEC
C     WRITE(7,20)IDEG,MIN,SEC
   20 FORMAT( ' TABLE OF COVARIANCES:',/,
     *' BETWEEN QUANTITIES OF KIND KP AND KQ, EVALUATED IN P,Q.',/,
     *' HAVING SPHERICAL DISTANCE PSI, HEIGHTS HP, HQ',/,
     *' AND AN AZIMUTH OF',I5,' D',I3,' M',F6.2,' SEC FROM P TO Q.')
      WRITE(6,30)
      WRITE(6,21)(KP(I),I=2,MT)
   21 FORMAT('   KP= ',6(I6,5X))
      WRITE(6,22)(KQ(I),I=2,MT)
   22 FORMAT('   KQ= ',6(I6,5X))
      WRITE(6,23)(HP(I),I=2,MT)
   23 FORMAT('   HP= ',6(1X,F10.1))
      WRITE(6,24)(HQ(I),I=2,MT)
   24 FORMAT('   HQ= ',6(1X,F10.1))
      WRITE(6,26)
   26 FORMAT(' PSI')
C
      DO K = 1, NCOV
       RM = COV(K,1)
       RM = RM/DEGRAD
       WRITE(6,25)RM,(COV(K,I), I = 2, MT)
   25  FORMAT(F10.5,F12.6,6F11.5)
      END DO
C
      IF(.NOT.LPUNCH) GOTO 888
      DO 301 KLM=2,MT
      DO 301 MLK=1,NCOV
       RM=COV(MLK,1)/DEGRAD
       WRITE(7,401) RM,COV(MLK,KLM)
  301 CONTINUE
  401 FORMAT(' ',F9.4,2F11.3)
C
  888 CONTINUE
C
      MT = 1
 887  IF (.NOT.LAST3) GO TO 104
C
      IF (.NOT.LAST2) GO TO 102
C
      IF (.NOT.LAST1) GO TO 100
      IF (MODE.NE.1) WRITE(6,54)KI(37),KI(36),KI(35)
   54 FORMAT(/' COVCG+BSFC OR TABH+ISPCOV CALLED',I6,' TIMES',/,
     *' COVCG OR TABH ALONE     CALLED',I6,' TIMES',/,
     *' COVCX           CALLED',I6,' TIMES.',/)
      IF(.NOT.LPUNCH) GOTO 999
      CLOSE(7)
C
      GO TO 999
  991 CONTINUE
C
C************************ INPUT 6 **********************************
C
C  MODE = 4: FITTING A COVARIANCE FUNCTION (REF. D).
C
C
C  INPUT OF ITERATION SPECIFICATIONS:
C
C   NUMBER OF ITERATIONS, TAU(1)-TAU(3) USED IN CX MATRIX.
C   (SEE EQ. (4.1) IN REF. D)
C
C   TAU(1) - TAU(3) ARE RELATIVE APRIORI STANDARD DEVIATIONS OF EACH
C   ADJUSTMENT. THEY ACT ON THE ESTIMATION OF AA, A, AND (RB-RE)
c   RESPECTIVELY.
```



```
C
      IF (LINTER) WRITE(*,*)
     *' INPUT NUMBER OF ITERATIONS AND 3 APRIORI WEIGHT FACTORS'
      READ(5,*) NITMAX,TAU(1),TAU(2),TAU(3)
C5531 FORMAT(I4,3F6.2)
C
      TAU0(1)=TAU(1)
      TAU0(2)=TAU(2)
      TAU0(3)=TAU(3)
C
      WRITE(6,5529) NITMAX,TAU(1),TAU(2),TAU(3)
 5529 FORMAT(/' NUMBER OF ITERATIONS:',I5,'.',//,
     *' THE VALUE OF TAU(J) USED IN THE CX MATRIX: CX(J,J)=TAU(J)**2',
     */,' AA:',F6.2,', A:',F6.2,', RB-RE:',F6.2,'.',/)
      WRITE(6,5527)
 5527 FORMAT(/,' (AA IS EQUAL TO THE CONSTANT SMALL A)',/)
C
C************************ INPUT 7 **********************************
C
C  INPUT OF EMPIRICAL COVARIANCE VALUES.
C
C   NUMBER OF DATA SETS.
C
      IF (LINTER) WRITE(*,*)' INPUT NO. OF DATASETS'
      READ(5,*) NDSET
C5532 FORMAT(I3)
C
      WRITE(6,5534) NDSET
 5534 FORMAT(/' NUMBER OF DATA SETS:',I3,'.')
C
C************************ INPUT 7A **********************************
C
C   FOR EACH DATA SET:
C   NUMBER OF VALUES, KIND KP AND KQ, HEIGHTS HP AND HQ.
C   MODERR AND A RELATIVE WEIGHT USED AS ERRY=ERRY/WEIGHT.
C
C     IF MODERR=0 THE ERRORS ARE READ FROM INPUT WITH THE VALUES.
C     IF MODERR=1 THE ERRORS ARE CALCULATED FROM THE VARIANCES AND
C       THE AREA SPECIFICATIONS (EQ. (3.1) IN REF. D).
C   LFILE, TRUE IF COVARIANCES ARE ON A FILE.
C
      NDAT=0
      IC3=3
      IC6=6
      DO 708 ISET=1,NDSET
C
      IF (LINTER)
     *WRITE(*,*)' INPUT NO. OF VALUES, DATA TYPE 1, DATA TYPE 2,',
     *' HEIGHT1, HEIGHT2, ERROR-MODE, WEIGHT, DATA ON FILE?(T/F)',
     *' WHERE ERROR-MODE=1: DATA DISTRIBUTION DEPENDENT',
     *' -  -  - =2: COVARIANCE-VARIANCE DEPENDENT'
      READ(5,*) IDAT,IKP,IKQ,HHP,HHQ,MODERR,WEIGHT,LFILE
C5533 FORMAT(3I3,2F8.3,/,I3,F8.3)
C
      LMEAP = IKP.GT.100
      LMEAQ = IKQ.GT.100
c
      IF (LMEAP) THEN
      IKP=IKP-100
C ---------------- INPUT (7B) ---------------------------------
      IF (LINTER)WRITE(6,1117)
      READ(5,*)LEQANG,BSIZEN,BSIZEE,RLATP
C LMEAN1 IS TRUE IF MEAN VALUES ARE 1D, ALONG A SATELLITE OR
C AIRCRAFT TRACK, FOR EXAMPLE BUT IS NOT MEANINGFULL FOR A GRID.
      LMEAP1=(.NOT.LEQANG).AND.ABS(BSIZEE).LT.1.0D-8
      IF (LMEAP1) THEN
      LFILTE=LF
C TRACK-AZIMUTH IS FORCED TO 0.0.
      SAZP=D0
```

```
      CAZP=D1
      CALL MEAN1(FILTER,NFILTE,SAZP,CAZP,LFILTE,LF,LINTER)
      END IF
C
      IF (LEQANG) WRITE(6,232)BSIZEN,BSIZEE
      IF ((.NOT.LEQANG).AND.(.NOT.LMEAP1))
     *WRITE(6,233)BSIZEN
      IF (LMEAP1) WRITE(6,236)BSIZEN
      RLATP=RLATP*3600/RADSEC
      COSLAP=COS(RLATP)
      SINLAP=SIN(RLATP)
      BSIZEN=BSIZEN*60/RADSEC
      IF (.NOT.LEQANG)BSIZEE=D0
      BSIZEE=BSIZEE*60/RADSEC
      IF (LMEAP1) THEN
      NSTEP=NFILTE
      ELSE
      NSTEP=5
      END IF
      NSTEPE=1
      STEPE=D0
      COSSTE=D1
      SINSTE=D0
C THE CALL TO ICMEAN GIVES STEPSIZE AND COS,SIN.
      CALL ICMEAN(BSIZEN,STEPN,NSTEP,COSSTN,SINSTN,D1,D0,LT,LMEAP1)
      SHIFTS=STEPN*(NSTEP-1)/2
      COST2P=COS(SHIFTS)
      SINT2P=SIN(SHIFTS)
      IF (LMEAP1) THEN
C LMEAN1 INTRODUCED 1992.10.07 BY CCT.
      BSIZEN=-BSIZEN
C WE USE THE SWITCH OF SIGN AS AN INDICATOR OF 1D MEAN.
      ELSE
      NSTEPE=5
      IF (.NOT.LEQANG) THEN
      CALL ICMEAN(BSIZEN,STEPE,NSTEPE,COSSTE,SINSTE,D1,D0,LT,LF)
      ELSE
      BSIZEA= COS(RLATP)*BSIZEE
      CALL ICMEAN(BSIZEA,STEPE,NSTEPE,COSSTE,SINSTE,D1,D0,LT,LF)
      END IF
      END IF
C
      END IF
c
      IF (LMEAQ) THEN
      IKQ=IKQ-100
C -------------- INPUT (7C) ---------------------------------
      IF (LINTER)WRITE(6,1117)
      READ(5,*)LEQANQ,BSIZEN,BSIZEE,RLATP
C LMEAN1 IS TRUE IF MEAN VALUES ARE 1D, ALONG A SATELLITE OR
C AIRCRAFT TRACK, FOR EXAMPLE BUT IS NOT MEANINGFULL FOR A GRID.
      LMEAQ1=(.NOT.LEQANQ).AND.ABS(BSIZEE).LT.1.0D-8
      IF (LMEAQ1.AND.(.NOT.LMEAP1)) THEN
      LFILTE=LF
      CALL MEAN1(FILTER,NFILTE,SAZQ,CAZQ,LFILTE,LF,LINTER)
C TRACK-AZIMUTH IS FORCED TO 0.0.
      SAZQ=D0
      CAZQ=D1
      END IF
C
      IF (LEQANQ) WRITE(6,232)BSIZEN,BSIZEE
      IF ((.NOT.LEQANQ).AND.(.NOT.LMEAQ1))
     *WRITE(6,233)BSIZEN
      IF (LMEAQ1) WRITE(6,236)BSIZEN
      RLATP=RLATP*3600/RADSEC
      COSLAP=COS(RLATP)
      SINLAP=SIN(RLATP)
      BSIZEN=BSIZEN*60/RADSEC
      IF (.NOT.LEQANQ)BSIZEE=D0
```



```
      BSIZEE=BSIZEE*60/RADSEC
      IF (LMEAQ1) THEN
      NSTEQ=NFILTE
      ELSE
      NSTEQ=5
      END IF
      NSTEQE=1
      STEQE=D0
      COSSQE=D1
      SINSQE=D0
C THE CALL TO ICMEAN GIVES STEPSIZE AND COS,SIN.
      CALL ICMEAN(BSIZEN,STEQN,NSTEQ,COSSQN,SINSQN,D1,D0,LT,LMEAQ1)
      SHIFTS=STEQN*(NSTEQ-1)/2
      COST2Q=COS(SHIFTS)
      SINT2Q=SIN(SHIFTS)
      IF (LMEAP1) THEN
C LMEAP1 INTRODUCED 1992.10.07 BY CCT.
      BSIZEN=-BSIZEN
C WE USE THE SWITCH OF SIGN AS AN INDICATOR OF 1D MEAN.
      ELSE
      NSTEPE=5
      IF (.NOT.LEQANQ) THEN
      CALL ICMEAN(BSIZEN,STEQE,NSTEQE,COSSQE,SINSQE,D1,D0,LT,LF)
      ELSE
      BSIZEA= COS(RLATP)*BSIZEE
      CALL ICMEAN(BSIZEA,STEQE,NSTEQE,COSSQE,SINSQE,D1,D0,LT,LF)
      END IF
      END IF
C
      END IF
c
C********************** INPUT (7D) **********************************
C
C     IF MODERR=1 ENTER C1, C2, FI1, FI2, DFI, LA1, LA2, AND DLA.
C
      IF(MODERR.EQ.1) THEN
      IF (LINTER)
     *WRITE(*,*)' INPUT VAR1, VAR2, FI1,2 DFI, LON1,2, DLO'
      READ(5,*) CC1,CC2,FI1,FI2,DFI,ALA1,ALA2,DLA
C5539 FORMAT(2F10.4,/,3F10.4,/,3F10.4)
      END IF
C
C CHANGE OF FORMAT 2004-06-22.
      WRITE(6,5535) ISET,IDAT,IKP,IKQ,HHP,HHQ,MODERR
 5535 FORMAT(/' DATA SET NO.',I3,':',
     ./,      '   NUMBER OF VALUES:',I4,',',
     ./,      '   BETWEEN KIND',I9,' AND',I10,','
     ,/,      '   AT HEIGHTS:',F10.1,' AND',F10.1,'.',
     .//,     '   MODE FOR ERROR:',I3,','
     .)
      IF(MODERR.EQ.0) WRITE(6,5541)
 5541 FORMAT('    READ FROM INPUT.')
      IF(MODERR.EQ.0) GO TO 709
      WRITE(6,5542) CC1,CC2,FI1,FI2,DFI,ALA1,ALA2,DLA
 5542 FORMAT(/'   ERRORS ARE CALCULATED USING:',
     ./,      '   THE VARIANCES OF THE TWO KINDS:',2F10.4,',',
     ./,      '   AND AREA SPECIFICATIONS:',
     ./,      '   FI1, FI2, AND DFI: ',3F10.4,',',
     ./,      '   LA1, LA2, AND DLA: ',3F10.4,'.')
C
      NPROD=((FI2-FI1)/DFI)*((ALA2-ALA1)/DLA)
      CC=SQRT(CC1*CC2)
      RPROD=NPROD
      ERR=CC/SQRT(RPROD)
C
      WRITE(6,5536) CC,NPROD,ERR
 5536 FORMAT(/'   GEOMETRIC MEAN OF VARIANCES:',F10.4,'.'
     .,/,     '   EXPECTED NUMBER OF PRODUCTS:',I10,'.'
     .,/,     '   VAR/SQRT(N)         =',F10.4,'.')
```

```
C
  709 WRITE(6,5543) WEIGHT
 5543 FORMAT(/'   RELATIVE WEIGHT FACTOR:',F8.3,'.',/)
C
      IF (LFILE) THEN
C INPUT OF NAME OF FILE HOLDNG COVARIANCE FUNCTION TABLE.
      IF (LINTER) WRITE(*,*)' INPUT NAME OF FILE HOLDING TABLE'
      READ(5,'(A)')DNAME(1)
      WRITE(6,409) (DNAME(IJ),IJ=1,ICHAR)
      OPEN(9,FILE=DNAME(1),STATUS='OLD')
  409 FORMAT(' INPUT OF COVARIANCES FROM FILE: ',2A48//)
      END IF
C
      LDENP=IKP.EQ.0.OR.IKP.EQ.16
      LDENQ=IKQ.EQ.0.OR.IKQ.EQ.16
      IF (.NOT.(LDENP.OR.LDENQ).OR.LDEN) GO TO 5590
C
C ********************** INPUT (7E) ******************************
C
C IF ONE OF THE TABELS ARE A DENSITY COVARIANCE-FUNCTION, PARAME-
C TERS DEFINING A QUASI-HARMONIC DENSITY MUST BE INPUT LIKE IN
C INPUT (5A).
      IF (LINTER) WRITE(*,*)' INPUT DENSITY SPEC.'
      READ(5,*)IEHD
      KI(32)=IEHD-1
      CCI(13)=10.0/(PI*6.67E-8*RE**(3-IEHD))
      WRITE(6,303)IEHD
      IF (LDENP) HHP=RE*HHP/(RE-HHP)
      IF (LDENQ) HHQ=RE*HHQ/(RE-HHQ)
      LDEN=.TRUE.
C
 5590 NDATO1=NDAT+1
      NDAT=NDAT+IDAT
      DO I=NDATO1,NDAT
C
C************************ INPUT (8) ********************************
C
C IF MODERR=0:
C      INPUT OF PSI(DEG), C(PSI), ERR.
C IF MODERR=1:
C      INPUT OF PSI(DEG), C(PSI), NA.
C
       IF (LFILE) THEN
        IF(MODERR.EQ.0) READ(9,1706)COV(I,1),COV(I,4),ERRY(I)
        IF(MODERR.EQ.1) READ(9,*) COV(I,1),COV(I,4),NA
       ELSE
        IF(MODERR.EQ.0) READ(5,1706)COV(I,1),COV(I,4),ERRY(I)
        IF(MODERR.EQ.1) READ(5,*) COV(I,1),COV(I,4),NA
       END IF
       COV(I,1)=COV(I,1)*0.0174533
       COV(I,2)=HHP
       COV(I,3)=HHQ
       KP(I)=IKP
       KQ(I)=IKQ
       IF(MODERR.EQ.0) GO TO 706
       ERRK=NPROD
C THE EXPECTED NUMBER OF PRODUCTS, ERRK, IS CALCULATED EXEPT FOR
C THE MULTIPLICATION BY DELTA-PSI. THIS IS DONE LATER IN THE DO-705.
C ( SEE EQ. (3.1) IN REF. D).
       IF(I.GT.NDATO1) ERRK=NPROD*360.0/DFI/DLA*COV(I,1)*57.29578
       ERRY(I)=(ERR*ERRK)/NA
  706  ERRY(I)=ERRY(I)/WEIGHT
      END DO
      IF(MODERR.EQ.0) GO TO 708
      DO 705 I=(NDATO1+1),(NDAT-1)
      DPSI=(COV(I+1,1)-COV(I-1,1))/2.0
      ERRY(I)=ERRY(I)*DPSI
  705 CONTINUE
      ERRY(NDAT)=ERRY(NDAT)*DPSI
```



```
  708 CONTINUE
C
 1706 FORMAT(F8.3,F10.4,10X,F10.1)
C1707 FORMAT(F8.3,F10.4,I10)
      WRITE(6,1708)
 1708 FORMAT(1H1)
C
      NITER=0
      VGOLD=VG
      RB2OLD=RB2
      SOLD=S
      CCI10=CCI(10)
      TAU1=TAU(1)
      TAU2=TAU(2)
      TAU3=TAU(3)
      LOCOLD=LOCAL
C
 1709 IF (LITERA) WRITE(*,*)' SUMMATION LIMIT = ',IMAX

      NITER=NITER+1
      IF (NITER.GT.1) THEN
       DO I=1,MAXSM
        SIGMA(I)=D0
       END DO
       LOCAL=LT
       AAI=RB2OLD*1.0D-8
       IMAX1=IMAX+1
       NC1=IMAX1
       CCI(8)=AAI
       CCI(10)=CCI10
       S=CCI10
       VG=VGOLD
       CALL COVAX(SM,IS)
       CVV=VAR(SM,IS,3,SOLD,AAI,D0,IMAX1,LF,1.0D0,0.0D0,LF,SROT)
       LOCAL=LOCOLD
C
      DO 7609 I=1,(IMAX+IS)
      SIGMA0(I+IS+1)=0.0
      IF(I.LE.IMIN) GO TO 7609
      GO TO (7609,7603,7604,7605,7606,7607),MODEL1
 7603 SIGMA0(I+IS+1)=(2.0*I+1)*(VG*9.81)**2
      GO TO 7609
 7604 SIGMA0(I+IS+1)=VG*DRAPP(I+1)
      GO TO 7609
 7605 SIGMA0(I+IS+1)=VG*DGPM2(I+1)
      GO TO 7609
 7606 SIGMA0(I+IS+1)=I*1.0E-2*VG
      GO TO 7609
 7607 SIGMA0(I+IS+1)=I**2*1.0D-4*VG
 7609 CONTINUE
C
       IF (MODEL.NE.0) GO TO 7069
       REWIND(19)
       READ(19,*)(SIGMA(I+1),I=IMIN,IMAX)
       REWIND(19)
C
 7069  SUMSIG=D0
       IF (LMULTF) THEN
       DO 7068 I=IMIN,IMAX
 7068   SIGMA0(I+IS+1)=SIGMA(I+1)*VG
       ELSE
       DO 7070 I=IMIN,IMAX
C7070   SIGMA0(I+IS+1)=SIGMA(I+1)*VG*(VZERO+I/(IMAX-1))
 7070   SIGMA0(I+IS+1)=SIGMA(I+1)*(VZERO+(VG-VZERO)*I/(IMAX-1))
       END IF
       DO I=IMIN,IMAX
        IF (LMULTF) THEN
        SUMSIG=SUMSIG+SIGMA(I+IS+1)*VG
        ELSE
```

```
        SUMSIG=SUMSIG+SIGMA(I+1)*(VZERO+(VG-VZERO)*I/(IMAX-1))
        END IF
       END DO
       AAI=(VARDG2-SUMSIG)*RB2*1.0D-8/CVV
       IF (LT) WRITE(*,*)' AAI, CVV ',AAI,CVV
      END IF
      DX(1)=VGOLD
      CCI(8)=AAI
      DX(2)=CCI(8)
      DX(3)=CCI10
      TAU(1)=TAU1
      TAU(2)=TAU2
      TAU(3)=TAU3
      LNEW=LT
C
C***
C*** NITMAX ITERATIONS IN "DO-678" LOOP.
C***
      DEV2=1.0E30
C
      NIT=-1
  680 NIT=NIT+1
C
      WRITE(6,498)NIT
C
  928 CONTINUE
      WRITE(6,509)VG,CCI(8),CCI(10),(DX(J),J=1,3)
      IF (NITER.GT.0) THEN
       DEV2A(NITER,2)=VG
       DEV2A(NITER,3)=CCI(8)
       DEV2A(NITER,4)=CCI(10)
      END IF
C ADDITION 2009-02-04.
      IF (VG.LT.D0) THEN
       WRITE(*,620)VG
 620   FORMAT(' SCALE FACTOR AA ON ERROR-DEGREE VARIANCES NEGATIVE, ',
     * ' STOP, AA= ',D12.4)
       STOP
      END IF
C
C**  CALCULATION OF MODEL-VALUES (III=1), MODEL WITH A=A+1 (III=2),
C**  AND MODEL WITH RB=RB+1 KM (III=3) IN "DO-303"-LOOP.
C**
      DO 7303 III=1,3
      IF(III.EQ.2) VG=VG+1.0
      IF(III.EQ.3) CCI(10)=CCI(10)-1.0
      DO 609 I=1,(IMAX+IS)
      SIGMA0(I+IS+1)=0.0
      IF(I.LE.IMIN) GO TO 609
      GO TO (609,603,604,605,606,607),MODEL1
  603 SIGMA0(I+IS+1)=(2.0*I+1)*(VG*9.81)**2
      GO TO 609
  604 SIGMA0(I+IS+1)=VG*DRAPP(I+1)
      GO TO 609
  605 SIGMA0(I+IS+1)=VG*DGPM2(I+1)
      GO TO 609
  606 SIGMA0(I+IS+1)=I*1.0E-2*VG
      GO TO 609
  607 SIGMA0(I+IS+1)=I**2*1.0D-4*VG
  609 CONTINUE
C
      IF (MODEL.NE.0) GO TO 1069
      REWIND(19)
      READ(19,*)(SIGMA(I+1),I=IMIN,IMAX)
      REWIND(19)
C
      IF (LMULTF) THEN
       DO 1068 I=IMIN,IMAX
 1068  SIGMA0(I+IS+1)=SIGMA(I+1)*VG
```



```
      ELSE
       DO 1070 I=IMIN,IMAX
 1070  SIGMA0(I+IS+1)=SIGMA(I+1)*(VZERO+(VG-VZERO)*I/(IMAX-1))
      END IF
C
 1069 CALL COVAX(SM,IS)
      KPP=-1
      KQQ=-1
C
      DO 7120 I=1,NDAT
      LNEW= KP(I).NE.KPP.OR.KQ(I).NE.KQQ
      KI(6)=KP(I)
      KI(7)=KQ(I)
      IF(LNEW) CALL COVBX(SM,LF,0)
      KPP=KP(I)
      KQQ=KQ(I)
C
      RLATP=D0
      RLONGP=D0
      COSLAP = D1
      SINLAP = D0
      COSLOP= D1
      SINLOP= D0
      IF (LMEAP) THEN
       IF (.NOT.LMEAP1) RLATP=RLATP+STEPN*D2
       COSLAP =  COS(RLATP)
       SINLAP =  SIN(RLATP)
       IF (LMEAP1) THEN
       CALL PAZIM(RLATP,RLONGP,COSLAP,SINLAP,COSLOP,SINLOP,
     * -CAZP,-SAZP,COST2P,SINT2P,LTEST)
      ELSE
       IF (.NOT.LEQANG)CALL
     * ICMEAN(BSIZEN,STEPE,NSTEPE,COSSTE,SINSTE,COSLAP,SINLAP,LF,LF)
       RLONGP=RLONGP-STEPE*D2
      END IF
      COSLOP = COS(RLONGP)
      SINLOP = SIN(RLONGP)
      END IF
C
      T = COS(COV(I,1))
      U = SIN(COV(I,1))
      RLATQ=ATAN2(U,T)
      RLONGQ=D0
      COSLAQ = T
      SINLAQ = U
      COSLOQ= D1
      SINLOQ= D0
      IF (LMEAQ) THEN
       IF (.NOT.LMEAQ1) RLATQ=RLATQ+STEPN*D2
       COSLAQ =  COS(RLATQ)
       SINLAQ =  SIN(RLATQ)
       IF (LMEAQ1) THEN
        CALL PAZIM(RLATQ,RLONGQ,COSLAQ,SINLAQ,COSLOQ,SINLOQ,
     *  -CAZP,-SAZP,COST2Q,SINT2Q,LTEST)
       ELSE
        IF (.NOT.LEQANQ)CALL
     *  ICMEAN(BSIZEN,STEQE,NSTEQE,COSSQE,SINSQE,COSLAQ,SINLAQ,LF,LF)
        RLONGQ=RLONGQ-STEPE*D2
       END IF
       COSLOQ = COS(RLONGQ)
       SINLOQ = SIN(RLONGQ)
       CQ=COSLAQ
       SQ=SINLAQ
      END IF
C
C TRANSFER OF COORDINATE INFORMATION TO THE SUBROUTINE ACCORDING TO
C THE SPECIFICATIONS GIVEN IN THE SUBROUTINE.
      CR(1) = T
      CR(2) = COV(I,2)
```

```
      CR(3) = COV(I,3)
      CR(4) = SINLAP
C ERROR DETECTED 2001-11-02.
c     CR(4) = SINLOP
      CR(5) = U
      CR(6) = COSLAP
c     CR(6) = COSLOP
      CR(7) = T
C ERROR 2001-11-02
      CR(8) = SINLOP
      CR(9) = COSLOP
      CR(10) = GM/(RE+CR(2))**2
      CR(11) = GM/(RE+CR(3))**2
      IF (LTEST)WRITE(6,18)U,T,SINLOP,COSLOP,COSLAP,SINLAP
C
      IF (LMEAP.OR.LMEAQ) THEN
       COV(I,4+III) = COMEAN(SM,0,0,COSLAP,SINLAP,COSLOP,SINLOP,
     * COSLAQ,SINLAQ,COSLOQ,SINLOQ,NSTEP,NSTEQ,LF,LCZERO,LF)
      ELSE
       CALL COVCX(SM,COV(I,4+III),0,LF)
      end if
C
      IF (.NOT.LTEST) GO TO 7120
      KK = KI(8)+1
      WRITE(6,19)((CR(J*8+K+3),K=1,8),J=1,KK)
 7120 CONTINUE
C
      IF(III.EQ.2) VG=VG-1.0
      IF(III.EQ.3) CCI(10)=CCI(10)+1.0
 7303 CONTINUE
C
      FAK1=VG
      FAK2=CCI(8)
      FAK3=CCI(10)
      IF ( ABS(FAK1).LT.0.01) FAK1=0.01
C
      DO 627 I=1,3
      KX(I)=1
      A(I)=0.0
      A(I+3)=0.0
      B(I)=0.0
  627 CONTINUE
      DEVOLD=DEV2
      DEV1=0.0E0
      DEV2=0.0E0
      IF(NIT.EQ.0.OR.NIT.EQ.NITMAX)
     .WRITE(6,499)
  499 FORMAT(1H0,'  KP KQ',
     +'  PSI    HP    HQ    OBS    MODEL     ',
     +'ERR   DIF/ERR',/)
C
C*   CALCULATION OF THE PARTIAL DERIVATIVES (NUMERICALLY) IN DX(J),
C*   THE AT(C-1)A MATRIX AND AT(C-1)Y VECTOR.
C*   CALCULATION OF DEVIATIONS BETWEEN DATA AND MODEL.
C*   IT ALL IN "DO-629" LOOP.
C*
      DO 629 I=1,NDAT
      DXD1=(COV(I,6)-COV(I,5))*FAK1
      DXD2=(COV(I,5)-VG*(COV(I,6)-COV(I,5)))
      DXD3=-(COV(I,7)-COV(I,5))*FAK3
      A(1)=A(1)+(DXD1/ERRY(I))**2
      A(2)=A(2)+(DXD1/ERRY(I))*(DXD2/ERRY(I))
      A(3)=A(3)+(DXD2/ERRY(I))**2
      A(4)=A(4)+(DXD1/ERRY(I))*(DXD3/ERRY(I))
      A(5)=A(5)+(DXD2/ERRY(I))*(DXD3/ERRY(I))
      A(6)=A(6)+(DXD3/ERRY(I))**2
      B(1)=B(1)+(COV(I,4)-COV(I,5))/(ERRY(I)**2)*DXD1
      B(2)=B(2)+(COV(I,4)-COV(I,5))/(ERRY(I)**2)*DXD2
      B(3)=B(3)+(COV(I,4)-COV(I,5))/(ERRY(I)**2)*DXD3
```



```
      DEV1=DEV1+(COV(I,4)-COV(I,5))**2
      DEV2=DEV2+((COV(I,4)-COV(I,5))/ERRY(I))**2
      IF(.NOT.(NIT.EQ.0.OR.NIT.EQ.NITMAX)) GO TO 629
      COV45=(COV(I,4)-COV(I,5))/ERRY(I)
      WRITE(6,500)I,KP(I),KQ(I),COV(I,1)*57.29578,
     .COV(I,2),COV(I,3),COV(I,4),COV(I,5),
     .ERRY(I),COV45
  629 CONTINUE
C
      IF (DEV2.GT.1.0D-9) THEN
       DEV2=SQRT(DEV2/NDAT)
      ELSE
       WRITE(*,*)' WARNING: DEV = ',DEV
       DEV=D1
      END IF
      DEV21=1.0E0-DEV2/DEVOLD
      WRITE(6,508) DEV2, DEV21
C ADDED 2001-11-17.
      IF (NITER.LE.MAXSTI) THEN
       DEV2A(NITER,1)=DEV2
       NITS(NITER)=NIT
      END IF
C
      IF(DEV2.LE.DEVOLD) GO TO 628
C
      VG=VG-DX(1)
      CCI(8)=CCI(8)-DX(2)
      CCI(10)=CCI(10)-DX(3)
      DX(1)=0.0E0
      DX(2)=0.0E0
      DX(3)=0.0E0
      TAU(1)=TAU(1)/2.
      TAU(2)=TAU(2)/2.
      TAU(3)=TAU(3)/2.
      WRITE(6,1701)
      GO TO 928
 1701 FORMAT(' >> BAD STEP. LAST ADJUSTMENT IGNORED !!!',/)
  628 CONTINUE
C
C*   CALCULATION OF THE ADJUSTMENT OF THE MODEL-PARAMETERS.
C*
C ADDED 2001-11-12 BY CCT.
      IF (ABS(DX(1)).LT.1.0D-5.AND.ABS(DX(2)).LT.D1.AND.
     *ABS(DX(3)).LT.0.1.AND.DEV21.LT.1.0D-5.AND.DEV21.gt.D0) THEN
       WRITE(*,*)' ITERATION TERMINATED AT ',NIT
       write(*,*)' BB ',cci(10),dx(1),dx(2),dx(3)
       GO TO 679
      END IF
      IF(NIT.EQ.NITMAX) GO TO 678
C  ADD THE CX-1 MATRIX TO THE ATA MATRIX.
      A(1)=A(1)+1./(TAU(1)**2)
      A(3)=A(3)+1./(TAU(2)**2)
      A(6)=A(6)+1./(TAU(3)**2)
C
      LBACK=.TRUE.
      LREDUC=.TRUE.
      CALL PRONLL(A,KX,B,IC3,ERR,LREDUC,LBACK,IC6,IC3,IC3)
      DX(1)=B(1)*FAK1
      DX(2)=B(2)*FAK2
      DX(3)=B(3)*FAK3
C
      VG=VG+DX(1)
      CCI(8)=CCI(8)+DX(2)
      CCI(10)=CCI(10)+DX(3)
C
      IF(CCI(10).LT.D0) GO TO 678
      CCI(10)=(CCI(10)-DX(3))/2.
      DX(3)=-CCI(10)
      TAU(3)=TAU(3)/2.
```

```
  678 IF (NIT.LT.NITMAX) GO TO 680
C
  679 NIT=NITMAX
      A(1)=A(1)+D1/(TAU0(1)**2)
      A(3)=A(3)+D1/(TAU0(2)**2)
      A(6)=A(6)+D1/(TAU0(3)**2)
      WRITE(6,512)(TAU(J),J=1,3)
      WRITE(6,513)
C
      LREDUC=.TRUE.
      LBACK=.TRUE.
      DO 808 I=1,3
      DO 806 J=1,3
  806 B(J)=0.0E0
      B(I)=1.0E0
      CALL PRONLL(A,KX,B,IC3,ERR,LREDUC,LBACK,IC6,IC3,IC3)
      LREDUC=.FALSE.
      IF (ERR.GT.1.0D-9) THEN
       ERRX(I)=SQRT(ERR)
      ELSE
       WRITE(*,*)' WARNING ',ERR
      END IF
      WRITE(6,501) I,(B(J),J=1,3)
  808 CONTINUE
      ERRX(1)=ERRX(1)*ABS(FAK1)
      ERRX(2)=ERRX(2)*ABS(FAK2)
      ERRX(3)=ERRX(3)*ABS(FAK3)
      WRITE(6,511)(ERRX(J),J=1,3),(ERRX(J)*DEV2,J=1,3)
C
C  CALCULATION OF C(0) FOR GRAVITY ANOMALIES
C
      IF (MODEL.GT.0) THEN
       DO 809 I=1,(IMAX+IS)
        SIGMA0(I+IS+1)=0.0E0
        IF(I.LE.IMIN) GO TO 809
        GO TO (813,814,815,816,817), MODEL
  813    SIGMA0(I+IS+1)=(2.0E0*I+1)*(VG*9.81E0)**2
        GO TO 809
  814    SIGMA0(I+IS+1)=VG*DRAPP(I+1)
        GO TO 809
  815    SIGMA0(I+IS+1)=VG*DGPM2(I+1)
        GO TO 809
  816    SIGMA0(I+IS+1)=I*1.0D-2*VG
        GO TO 809
  817    SIGMA0(I+IS+1)=I**2*1.0D-4*VG
  809  CONTINUE
      ELSE
       READ(19,*)(SIGMA(I+1),I=IMIN,IMAX)
C
       DO I=IMIN,IMAX
        IF (LMULTF) THEN
         SIGMA0(I+IS+1)=SIGMA(I+1)*VG
        ELSE
c        SIGMA0(I+IS+1)=SIGMA(I+1)*VG*(VZERO+I/(IMAX-1))
         SIGMA0(I+IS+1)=SIGMA(I+1)*(VZERO+(VG-VZERO)*I/(IMAX-1))
        END IF
       END DO
      END IF
C
      CALL COVAX(SM,IS)
      KI(6)=3
      KI(7)=3
      CALL COVBX(SM,LSAT,0)
      CR(1)=1.0E0
      CR(2)=0.0E0
      CR(3)=0.0E0
      CR(4)=D0
      CR(5)=0.0E0
      CR(6)=D1
```



```
      CR(7)=D1
      CR(8)=D0
      CR(9)=D1
      CR(10)=GM/RE**2
      CR(11)=GM/RE**2
      CALL COVCX(SM,COV14,0,LF)
C
      WRITE(6,513) COV14
      IF (NITER.GT.0) DEV2A(NITER,5)=COV14
C
  498 FORMAT(1H0,' RESULT OF ITERATION NO.',I3,':',/,1X,26(1H-))
C FORMAT CHANGED 2004-06-22 AND 2011-08-01.
  500 FORMAT(1X,3I3,F8.3,2f9.1,3F11.5,F8.4)
  501 FORMAT(2X,I1,1H','TH ROW OF INVERSE MATRIX  ',3E14.4)
 508 FORMAT(' RMS VALUE OF DIFFERENCES/ERRORS:',F14.6,
   + /,  ' (RELATIVE CHANGE:                ',F14.6,').',/)
 509 FORMAT(
   *25X,'          AA                A             RB-RE',/,
   *' NEW VALUE:',15X,F14.6,F14.0,F14.3,/,
   *' LAST ADJ.:',15X,F14.6,F14.0,F14.3,/)
 511 FORMAT(1H0,' STD.DEV.      ',3E14.6,/,
   .     '  STD.DEV.*RMS ',3E14.6)
 512 FORMAT(1H0,' TAU(J) USED IN THE CX MATRIX',3(E14.2))
 513 FORMAT('0RESULTS IN VARIANCE OF GRAVITY ANOMALIES:',
   .F9.2,' MGAL**2.')
C
    IMAX=IMAX-ISTEP
    IF (LITERA.AND.IMAX.GE.NMIN) GO TO 1709
    IF (NITER.GT.0) THEN
     WRITE(*,515)
 515  FORMAT(/,
   * ' N        RATIO       AA       A          RE-RB     VARG     IT ')
     DO I=1,NITER
      WRITE(*,514)(ISTART-(I-1)*ISTEP),(DEV2A(I,K),K=1,5),NITS(I)
 514  FORMAT(I4,D12.4,F8.4,D12.4,2F9.2,I5)
     END DO
    END IF
C
    WRITE(6,*)' '
 999 IF (MODEL.EQ.0) CLOSE(19)
    STOP
    END
    BLOCK DATA
C PROGRAMMED BY C.C.TSCHERNING, GEODETIC INSTITUTE, 1974.
C UPDATED: 2008-10-18 BY CCT.
C THE SUBROUTINE INITIALIZES A NUMBER OF VARIABLES. IT MAY BE
C SUBSTITUTED BY A "BLOCK DATA" CALL ON OTHER COMPUTERS.
C ON ICL-COMPUTERS, IT MUST HAVE A NAME, AND BE DECLARED AS
C AN EXTERNAL.
C
    IMPLICIT NONE
    INTEGER MAXO,NSAT,NCTA,MXPAR,NIPT,NIPCAT,MAXOD
   *,KCI,NC1,NC2,NFU,KEYH,NINTH,NTABH,NHE,IOBS2,NSTART,KSAT,
   *NDX1,NDX2,I4,NDP,IPACAT,NDQ,IT,K3,K4,NUM,INN,ITCOUN,
   *IGP,NBOLD,NWAR,IA1,IKP,IU1,IC11,IMAX1,IMAX1R,INV,ITE1,
   *KK,INUMR,IP1,K21,K2,IU,IITE1,IITE,IIP1,IIP,IIE,
   *IIE1,K2P3,IT1,ITE,IP,IC1,IA,IB,NNX,NTABX,IFQ,ISATP,ISAT,
   *IHQ,IHP,INDEX,NR,NI,ICZERO,J2,K8,INZOLD,IEM,K21X,INL,K17,NAI,
   *K15,ICSYSL,K11,NO1,K9,K7,NO,IOBS1,IANG,IH,MP,IPAR,IFP,
   *KFQ,JR,NOBLK,IXX,K13,K19,NCZERO,NLA,INO,IB1,ISO,IPX,IS,
   *JJDEG,IIDEG,K1,IPTYPE,K23,I3,IPA,KFP,NPARM,NPARM1,MAXPAR,
   *II,NMAX,MAXB,IX,NCXLAS
C   *,ITIME,ITIME0
C
    REAL*8 GM,RLOMAX,RLAMAX,RLOMIN,RLAMIN,B,HQ,RLAT,
   *SINLAT,COSLAT,RLONG,SINLON,COSLON,WOBS,SINLOP,
   *COSLOP,BSIZE,BSIZEN,BSIZEE,COSLAP,SINLAP,RLONGP,RP,CAZP,SAZP,
   *CCI,CCR,SIGMA0,SIGMA,HCMAX,CCV,D,OBS,OLDR,SLOQ,CFX,
   *RE,BIPC,CRHT,PREDP,HP,RLATP,BIP,HQOLD,C11,CTA,CTTF,CTSF,
```

```
     *SZ,AZ,HTA,TMAX,SIZEI,COVX,CIX,SLOP,D2,CLOP,CLOQ,GMC,PI,DXX,HCZERO,
     *VARI,DGPM2,SCALE,SCALE2,DRAPP,OLDT,RADSEC,CFA,SIGMAP,HPOLD,
     *D5,D0,D1,D3,D4,PRETAP

      PARAMETER (MAXO=5600,NSAT=5600,NCTA=1600,MXPAR=2500,NIPT=1500,
     *NIPCAT=100002,MAXOD=50400)
C
      LOGICAL L,LN,LOPEN7,LONECO,LNKSIP,LNETAP,LDEFVP,LSTOP,LRESOL,
     *LC1,LC2,LCREF,LKM,LNEQ,LT,LNDAER,LPOSDA,LDEFF,LF,LGRID,LERNO,
     *LDENOL,LNEWD,LPUNCH,LOUTC,LNERNO,LK30,LK31,LIN4,LOPCOF,LCLU7,
     *LFIRST,LSUM,LOCAL,LWRSOL,LPOT,LMDD,LCOMP,LCOM,LWLONG,LPRED,
     *LPARAM,LTERRC,LPOTIN,LK2EQ4,LNUOUT,LTABLE,LTABLR,LNEQ8,LNEWSO,
     *LINT,LTERMA,LTERMO,LTERM,LCO1,LBIPOT,LBICOV,LBISOL,LINSOL,
     *HP9000,LOPEN4,LTABH,LTIME,LTCOV,LONEQ,LX,LNX,LTESTS,LOBSST
     *,LSPOUT,LTRAN,LFINI
C    *,LCOERR,LLCOER
C
      CHARACTER*128 OLDN,OLDCOV
      COMMON /CON1/OLDN(4)/CON2/GM,RLOMAX,RLAMAX,RLOMIN,RLAMIN,
     *ICSYSL,NO,NAI,NLA,INL,IEM,INZOLD,
     *LNDAER,LPOSDA,LDEFF,LERNO,LCOMP,LCOM,LWLONG,
     *LDENOL,LMDD,LIN4,LOPCOF,LCLU7,LOPEN7,LOPEN4,
     *LBIPOT,LBICOV,LBISOL,LINSOL,LTIME,LTCOV,LONEQ
C
      COMMON/PR/B(MAXO),HQ(MAXO),RLAT(MAXO),
     *SINLAT(MAXO),COSLAT(MAXO),RLONG(MAXO),
     *SINLON(MAXO),COSLON(MAXO),WOBS(MAXO),
C    *COSAZ(NSAT),SINAZ(NSAT),SINLOP,COSLOP,BSIZE(42),BSIZEN,BSIZEE,
     *SINLOP,COSLOP,BSIZE(42),BSIZEN,BSIZEE,
     *COSLAP,SINLAP,RLONGP,RP,CAZP,SAZP,
     *HP,RLATP,PRETAP,PREDP,HCZERO,ICZERO,NCZERO,
     *NI,NR,INDEX(42),IKP,ISAT(42),ISATP,NOBLK,
     *LONECO,LNKSIP,LNETAP,LDEFVP,LOBSST
C FOR A COMPLETE DESCRIPTION, SEE THE MAIN PROGRAM.
C
      COMMON /CMCOV/CCI(24),CCR(56),SIGMA0(2200),SIGMA(2200),HCMAX,
     *CCV(2,2),D(36),KCI(37),NC1,NC2,LOCAL,LSUM,LFINI
      COMMON /DDX/K7(17),K9(17),K11(17),K13(17),K15(17),K17(17),
     *K19(17),K21X(17),K23(17),K8(17),C11(17),J2(2),I3(2),I4(2),
     *LN(7),L(7)
C
      COMMON /TABELC/CTA(NCTA,16,2),CTTF(800),CTSF(20),SZ(30),AZ(18),
     *MAXB(20),IX(8),IXX(18)
      COMMON /CTABH/DRAPP(181),DGPM2(201),CRHT(818),HTA(5),TMAX,
     *SIZEI,NFU(5),KEYH(5,5),NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
      COMMON /CTABH1/HPOLD,HQOLD,IHP,IHQ,KFP,KFQ,IFP,IFQ,NTABX,NNX
C
      COMMON/DAT/LNEWD,LRESOL,LGRID
C
      COMMON/OUTC/INUMR(12),NO1,K2,K3,K2P3,K4,IU,K21,IU1,IANG,LPUNCH,
     *LTERMA,LTERMO,LTERM,
     *LOUTC,LTRAN,LNERNO,LK30,LK31,LWRSOL,LSTOP,LK2EQ4,LNUOUT
      COMMON /CHEAD/IA,IB,IH,IP,IT,IA1,IB1,IP1,IT1,IC1,IC11,K1,IOBS1,
     *IOBS2,ITE,ITE1,IITE,IITE1,IIP,IIP1,IIE,IIE1,INO,
     *LPOT,LKM,LTERRC,LPOTIN
      COMMON /CHEAD1/LC1,LC2,LCREF
      COMMON /COM2/DXX,NUM(70),VARI(32),SCALE,SCALE2,INN,INV
C
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GMC,ITCOUN,LF,LT
C
      COMMON /GPOTC1/OLDT,OLDR,CFA,IGP(12),LFIRST,HP9000
      COMMON /OBSER/OBS(22)
C
      COMMON/CPARM/IPTYPE(NIPT),IPACAT(3*NIPT),
     *NPARM,NPARM1,MAXPAR,MP,IPA,NCXLAS
C     COMMON/CCTIME/ITIME(NIPCAT),ITIME0(NIPT),LCOERR,LLCOER
C ONLY ACTIVE IN GEOCOL.
C
      COMMON /CCVCG/KK(24)
```



```
C
      COMMON /BIPAR/BIP(7),NMAX,II,IPAR(13),JR,ISO,LPARAM,LPRED,
     *LNEQ,LNEQ8,LNEWSO,LINT
      COMMON /BIPARC/OLDCOV(2),BIPC(4),NBOLD,IS,IPX,IMAX1,IMAX1R,LTABLE,
     *LTABLR,LCO1
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFX,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
      COMMON /PDEGV/SIGMAP(2200),SLOP,SLOQ,CLOP,CLOQ,
     *IIDEG,JJDEG,LSPOUT
C
C KSAT HOLDS THE MAPPING BETWEEN THE DATA CODES AND THE POSITIONS
c IN THE ARRAY COVCX HOLDING THE COVARIANCES. SEE SUBROUTINE COVCX.
      DATA KSAT/
     *1,3,3,3,3,2,1,2,1,2,1,2,1,1,1,1,1,
     *1,1,1,3,3,1,1,1,1,3,3,2,2,1,1,1,1/
      DATA KK,KFP,HPOLD,HQOLD/1,5,2,5,5,3,4,9*5,1,3,9,11,0,1,4,5,
     *-1,2*-1.0D5/,KCI(26),KCI(27),KCI(28),KCI(29),KCI(30),KCI(31),
     *KCI(32),KCI(33),CCI(14)/1,0,1,0,2,0,-1,1,0.5/
     *,KCI(35),KCI(36),KCI(37)/3*0/
C INITIALIZING FOR DENSITY CONTRAST COMPUTATION.
C
      DATA
     *K7/5*0,6*1,4*2,2*0/,K9/5*1,2,3,2,3,2,3,2,2,3,4,2*0
     */,K11/11*0,2,3,3,6,2*0/,
     *K13/11*1,2,3,3,6,2*0/,K15/0,1,-1,-1,1,0,0,-1,-1,2,2,6*0/,
     *K17/3*0,2,2,12*0/,K19/1,4*0,1,1,10*0/,K21X/0,1,1,2,2,1,1,10*2/,
     *K23/5*1,2,1,2,1,2,1,1,1,2,2,0,0/,
     *K8/0,1,1,2,2,0,0,4*1,6*0/,C11,
     *HCMAX/1.0D0,2*1.0D5,2*1.0D9,2*-206264.806D0,5*1.0D9,2.0D9,
     *2*1.0D9,2*1.0D0,1.0D9
     */,D,BIP,BIPC/47*0.0D0/,J2/3,2/,I3/6,3/,I4/4,2/
C
      DATA RE,GMC,D1,D2,D3,D4,D5,
     *D0,BSIZE,SIGMA,SIGMA0,OLDT,OLDR,PREDP,PRETAP,OBS,
     *RADSEC,PI/6371.0D3,3.98D14,1.0D0,2.0D0,3.0D0,4.0D0,
     *5.0D0,4469*0.0D0,206264.806D0,3.1415926535D0/,LT,LNEQ,LSPOUT,
     *LNDAER,LCO1,LNERNO,LWRSOL,LBIPOT,LBICOV,LBISOL,LINSOL,LTABH,
     *LDENOL,LPOSDA,LFIRST,LCREF,LC1,LC2,LDEFF,LMDD,LIN4,LOPCOF,
     *LF,LGRID,LERNO,LCOMP,LCOM,LWLONG,LPRED,LCLU7,LOPEN7,LRESOL
     *,LTIME,LTCOV,LONEQ
     *,LTERRC,LTABLE,LTABLR,LNEQ8,LOPEN4/6*.TRUE.,34*.FALSE./,RLAMAX,
     *RLOMAX,RLAMIN,RLOMIN,HCZERO,ICSYSL,NCZERO/4*0.0D0,-3.0D8,-2,-1/,
     *ITCOUN,IPAR,NBOLD,ITE,ITE1,INZOLD,
     *IX,NPARM,NO,NAI,NLA,IS,ISO,IGP,IT,IP,INDEX(1),INDEX(2),
     *IA,IA1,INL,IEM,II,JR,NPARM1/48*0,2*9,10,25,2*2,1/
     *,IXX,IPX,NWAR/0,0,1,0,1,0,1,2,3,0,1,2,3,4,5,6,0,-2,-1,0/
C THE FACTOR MUST BE 2*NIPT.
C CHANGE 2002-02-05. DIMENSION OF ITIME NOW NIPCAT.
C DEACTIVATED - ACTIVE IN GEOCOL.
C     DATA IPTYPE,ITIME0/NIPT*0,NIPT*0/ITIME/NIPCAT*0/
C    *LCOERR,LLCOER/2*.FALSE./
C
C INITIALIZING VARIABLES IN /PR/ TO HOLD ERROR DEGREE-VARIANCES FOR
C POTENTIAL COEFFICIENT SOLUTIONS RAPP 1981 AND WENZELS GPM2.
      DATA DRAPP/
     *0.0,0.0,0.000006,0.0002455,0.0007523,0.005981,
     *0.007083,0.0264,0.03313,0.07616,0.06944,
     *0.128,0.095,0.135,0.135,0.140,0.177,0.168,0.185,0.203,0.195,
     *0.194,0.222,0.226,0.254,0.277,0.289,0.263,0.301,0.298,0.286,
     *0.280,0.293,0.274,0.283,0.292,0.304,0.183,0.188,0.193,0.198,
     *0.203,0.208,0.213,0.218,0.223,0.228,0.234,0.238,0.243,0.250,
     *0.254,0.259,0.266,0.271,0.276,0.282,0.287,0.291,0.297,0.303,
     *0.307,0.314,0.319,0.323,0.327,0.337,0.341,0.346,0.355,0.357,
     *0.362,0.371,0.376,0.385,0.387,0.392,0.398,0.406,0.410,0.418,
     *0.427,0.440,0.441,0.444,0.448,0.465,0.470,0.471,0.477,0.482,
     *0.496,0.495,0.510,0.511,0.514,0.521,0.532,0.545,0.541,0.555,
     *0.547,0.574,0.595,0.584,0.595,0.603,0.613,0.618,0.634,0.642,
     *0.641,0.650,0.652,0.657,0.702,0.681,0.692,0.694,0.710,0.730,
     *0.711,0.719,0.741,0.768,0.776,0.762,0.802,0.760,0.790,0.811,
```

```
     *0.776,0.835,0.844,0.824,0.859,0.869,0.864,0.907,0.884,0.914,
     *0.899,0.887,0.942,0.956,0.943,0.935,0.951,0.981,0.972,0.986,
     *1.009,1.015,1.018,1.021,1.049,1.095,1.060,1.072,1.104,1.117,
     *1.118,1.143,1.134,1.228,1.280,1.184,1.164,1.241,1.234,1.233,
     *1.270,1.230,1.268,1.361,1.320,1.297,1.323,1.352,1.450,1.363/
C
      DATA DGPM2/0.0,0.00,0.000030,0.000139,0.000208,0.000262,
     *0.000307,0.000338,0.000386,0.000422,0.000457,
     *0.000,13*0.001,0.002,0.002,0.002,0.002,0.002,0.003,
     *0.003,0.003,0.003,0.004,0.004,0.004,0.005,0.005,0.005,0.006,
     *0.006,0.007,0.007,0.008,0.008,0.009,0.009,0.010,0.010,0.011,
     *0.012,0.012,0.013,0.014,0.015,0.015,0.016,0.017,0.018,0.019,
     *0.020,0.020,0.021,0.022,0.023,0.024,0.026,0.027,0.028,0.029,
     *0.030,0.031,0.033,0.034,0.035,0.037,0.038,0.039,0.041,0.042,
     *0.044,0.045,0.047,0.049,0.050,0.052,0.054,0.056,0.057,0.059,
     *0.061,0.063,0.065,0.067,0.069,0.071,0.074,0.076,0.078,0.080,
     *0.083,0.085,0.087,0.090,0.092,0.095,0.097,0.100,0.103,0.105,
     *0.108,0.111,0.114,0.117,0.120,0.123,0.126,0.129,0.132,0.136,
     *0.139,0.142,0.146,0.149,0.153,0.156,0.160,0.164,0.168,0.171,
     *0.175,0.179,0.183,0.187,0.192,0.196,0.200,0.205,0.209,0.213,
     *0.218,0.223,0.227,0.232,0.237,0.242,0.247,0.252,0.258,0.263,
     *0.268,0.274,0.279,0.285,0.291,0.297,0.302,0.308,0.315,0.321,
     *0.327,0.334,0.340,0.347,0.353,0.360,0.367,0.374,0.381,0.389,
     *0.396,0.404,0.411,0.419,0.427,0.435,0.444,0.457,0.465,0.601,
     *0.659,0.549,0.523,0.521,0.526,0.533,0.541,0.549,0.558,0.567,
     *0.577,0.587,0.597,0.607,0.618,0.629,0.640,0.652,0.663,0.675/
C
      END
      SUBROUTINE RAD(IDEG,MIN,SEC,RA,IANG)
C THE SUBROUTINE CONVERTS FOR IANG = 1,2,3,4 ANGLES IN (1) DEGREES, MI-
C NUTES, SECONDS, (2) DEGREES, MINUTES, (3) DEGREES AND (4) 400-DEGREES
C TO RADIANS.
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE:
      IMPLICIT NONE
      REAL*8 PHI,SEC,RA,SE
      INTEGER I,MIN,IDEG,J,IANG
C
      PHI = 3.1415926536D0
      I = 1
      IF (IDEG .LT. 0 .AND. IANG .LT. 3) I = -1
      GO TO (1,2,3,4,3),IANG
    1 J = 1
      IF (MIN.LT.0) J = -1
      SE =I*IDEG*3600+J*MIN*60+SEC
      I = J*I
      GO TO 5
    2 SE=I*IDEG*3600+SEC*60
      GO TO 5
    3 SE = SEC*3600
      GO TO 5
    4 SE = SEC*3240
    5 RA= I*SE/206264.806D0
      IF (RA.GT.PHI) RA = RA-PHI*2.0D0
      IF (RA.LT.-PHI) RA = RA+PHI*2.0D0
      RETURN
      END
      SUBROUTINE MEAN1(FILTER,NFILTE,SAZP,CAZP,LFILTE,LGRID,LINTER)
C PROGRAMMED 1992.12.11 BY CCT. LAST UPDATE:  1995.01.16 BY CCT.
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE THE FOLLOWING STATEMENT:
      implicit none
      REAL*8 FILTER,SAZP,CAZP,SFILT,AZP,DEGRAD
      LOGICAL LFILTE,LGRID,LINTER
      INTEGER NFILTE,I
C
      DIMENSION FILTER(11)
      IF (.NOT.LFILTE) THEN
C IF 1-D MEANS ARE USED, IT IS HERE POSSIBLE TO INPUT UP TO 5 WEIGHTS
C WITH SUM EQUAL TO NUMBER OF WEIGHTS (NFILTE). ONLY ONE SET OF WEIGHTS
C MUST BE USED.
```



```
C CHANGE 1992.11.26 BY C.C.TSCHERNING.
      IF (LGRID)
     *WRITE(6,*)' 1-D MEANS NOT TO BE USED WITHOUT CAUTION'
      IF (LINTER) WRITE(*,*)' INPUT NUMBER OF FILTER FACTORS '
      READ(*,*)NFILTE
      IF (LINTER) WRITE(*,*)' INPUT ',NFILTE,' FILTER FACTORS '
      READ(*,*)(FILTER(I),I=1,NFILTE)
      DO 2073, I=1,NFILTE
 2073 SFILT=SFILT+ABS(FILTER(I))
      IF (ABS(SFILT-NFILTE).GT.1.0D-5) WRITE(6,*)
     *' *** WARNING *** FILTER MUST SUM TO NFILTE.'
      LFILTE=.TRUE.
      END IF
      IF (LGRID)  THEN
      IF (LINTER)WRITE(6,*)' INPUT AZIMUTH IN DEGREES '
      READ(*,*) AZP
      DEGRAD=3.1415926535D0/180.0D0
      SAZP = SIN(AZP*DEGRAD)
      CAZP = COS(AZP*DEGRAD)
      END IF
C
      RETURN
      END
C -----------------------------------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                      C
C                 I F R A C            C
C                                      C
C  SUBROUTINE GIVING TRUE INTEGER PART OF REAL REAL    C
C                                      C
C  RF, JUNE 1983                       C
C                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER FUNCTION IFRAC(R)
C
      implicit none
      REAL*8 R
      IF (R.LT.0.0D0) GO TO 1
        IFRAC = R
      RETURN
 1      IFRAC = R - 0.999999999D0
      RETURN
      END
C ---------------------------------------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                      C
C                 I S P C O V          C
C                                      C
C  INITIALIZATION PROCEDURE FOR FAST 1-DIMENSIONAL EQUIDISTANT     C
C  SPLINE INTERPOLATION, WITH FREE BOUNDARY END CONDITIONS         C
C  REFERENCE: JOSEF STOER: EINFUHRUNG IN DIE NUMERISCHE MATHEMATIK C
C  I, SPRINGER 1972. MODIFIED FOR COVARIANCE INTERPOLATION.        C
C                                      C
C  PARAMETERS (REAL):                  C
C                                      C
C  Y  GIVEN VALUES, Y(1), ..., Y(N)                 C
C                                      C
C  R  SPLINE MOMENTS (1 ... N), TO BE USED BY FUNCTION 'SPLINE'   C
C                                      C
C  Q  WORK-ARRAY, DECLARED AT LEAST 1:N                 C
C                                      C
C  RENE FORSBERG, JULY 1983, MODIFIED BY C.C.TSCHERNING OCT 87.    C
C                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE ISPCOV(NTAB,NN)
C
      implicit none
```

```
      INTEGER NCTA,MAXB,IX,IIX,IOLD,NFU,KEYH,NINTH,NTABH,NHE,NSTART,
     *N,KY0,NTAB,NN,KSTART,KK,K,KY
      REAL*8 Y,CTTF,CTSF,SZ,AX,R,HTA,TMAX,SIZEI,Q,P
      PARAMETER (NCTA=51200)
      LOGICAL LTABH
C
      COMMON /TABELC/Y(NCTA),CTTF(800),CTSF(20),SZ(30),AX(18),
     *MAXB(20),IX(8),IIX(17),IOLD
      COMMON /CTABH/R(1200),HTA(5),TMAX,SIZEI,NFU(5),KEYH(5,5),
     *NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
      DIMENSION Q(1200)
C
      N=NINTH+2
      KY0=(NTAB+NN-1)*N
      KSTART=N*(NN-1)
      Q(1) = 0.0D0
      R(1+KSTART) = 0.0D0
      DO 11 KK = 2, N-1
        K=KK+KSTART
        KY=KK+KY0
        P = Q(K-1)/2+2
        Q(K) = -0.5E0/P
        R(K) = (3*(Y(KY+1)-2*Y(KY)+Y(KY-1)) - R(K-1)/2)/P
   11 CONTINUE
      R(N+KSTART) = 0.0D0
      DO 12 KK = N-1, 2, -1
        K=KK+KSTART
        R(K) = Q(K)*R(K+1)+R(K)
   12 CONTINUE
      RETURN
      END
C ---------------------------------------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                        S P L C O V                                   C
C                                                                      C
C  FAST ONE-DIMENSIONAL EQUIDISTANT SPLINE INTERPOLATION FUNCTION.      C
C                                                                      C
C  PARAMETERS:                                                         C
C                                                                      C
C  X   INTERPOLATION ARGUMENT (REAL), X = 1 FIRST DATA-POINT,          C
C      X = N LAST DATA-POINT. OUTSIDE THE RANGE LINEAR EXTRA-          C
C      POLATION IS USED.                                               C
C                                                                      C
C  Y   REAL*8 ARRAY, 1 .. N : DATA VALUES                              C
C                                                                      C
C  R   DO: SPLINE MOMENTS CALCULATED BY SUBROUTINE 'ISPCOV'            C
C                                                                      C
C  PROGRAMMER:                                                         C
C  RENE FORSBERG, JUNE 1983, MODIFIED OCT 87 BY C.C.TSCHERNING          C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      FUNCTION SPLCOV(X, NTAB, NN)
C
      implicit none
      REAL*8 X,Y,CTTF,CTSF,SZ,AX,R,HTA,TMAX,SIZEI,XX,SPLCOV
      INTEGER NTAB,NN,NINTH,NTABH,NHE,NSTART,MAXB,IX,IIX,IOLD,
     *NFU,KEYH,NCTA,N,KY0,KY1,KSTART,J,IFRAC
      LOGICAL LTABH
      PARAMETER (NCTA=51200)
C
      COMMON /TABELC/Y(NCTA),CTTF(800),CTSF(20),SZ(30),AX(18),
     *MAXB(20),IX(8),IIX(17),IOLD
      COMMON /CTABH/R(1200),HTA(5),TMAX,SIZEI,NFU(5),KEYH(5,5),
     *NINTH, NTABH(15,5,5),NHE,NSTART,LTABH
C
      N=NINTH+2
      KY0=(NTAB+NN-1)*N
```



```
      KY1=KY0+1
      KSTART=N*(NN-1)
      IF(X.GE.1.0D0) GO TO 1
        SPLCOV = Y(KY1) + (X-1)*(Y(2+KY0)-Y(KY1)-R(2+KSTART)/6)
      RETURN
    1 IF(X.LE.FLOAT(N)) GO TO 2
        SPLCOV = Y(N+KY0)+(X-N)*(Y(N+KY0)-Y(KY0+N-1)+R(KSTART+N-1)/6)
      RETURN
    2   J = IFRAC(X)
        XX = X - J
        SPLCOV = Y(J+KY0) +
     .    XX * ((Y(J+KY1)-Y(J+KY0)-R(J+KSTART)/3-R(J+1+KSTART)/6) +
     .      XX * (R(J+KSTART)/2 +
     .      XX * (R(J+KSTART+1)-R(J+KSTART))/6))
      RETURN
      END
      SUBROUTINE ICMEAN
     *(BSIZE,STEP,NSTEP,COSST,SINST,COSLAT,SINLAT,LEQANG,LMEA1)
C PROGRAMMED BY C.C.TSCHERNING, GEODETIC INSTITUTE, NOV 1985.
C THE SUBROUTINE INITIALIZES STEP VARIABLES FOR MEAN VALUE
C COMPUTATION. CHANGED 1996.10.08 BY CCT.
C LEQANG IS TRUE, WHEN WE DEAL WITH EQUAL-ANGULAR BLOCK AVERAGES.
C LMEA1 IS TRUE WHEN WE HAVE 1-D MEANS.
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE THE FOLLOWING STATEMENT:
      implicit none
      LOGICAL LEQANG,LMEA1,LTEST
      REAL*8 BSIZE,STEP,COSST,SINST,COSLAT,SINLAT,BSIZEA
      INTEGER NSTEP,NSTEP1
C
      LTEST=.FALSE.
      NSTEP1=NSTEP-1
      BSIZEA=ABS(BSIZE)
      IF (LEQANG) GO TO 10
      STEP=2*BSIZE/4.0
      BSIZEA=BSIZEA/(COSLAT* COS(STEP)+SINLAT* SIN(STEP))
C CORRECTION 1995.11.21 BY CCT.
   10 IF (LMEA1) THEN
C FOR 1-D MEANS, THE POINTS ARE SUPPOSED TO BE DISTRIBUTED EQUIDISTANTLY
C ON THE INTERVAL OF SIZE BSIZE. FOR 2-D MEANS THEY ARE DISTRIBUTED
C WITH NSTEP POINTS INSIDE THE INTERVAL.
      STEP=BSIZEA/NSTEP1
      ELSE
      STEP=BSIZEA/NSTEP
      END IF
      COSST= COS(ABS(STEP))
      SINST= SIN(ABS(STEP))
      IF (LTEST) WRITE(*,*)' ICMEAN: STEP= ',STEP
      RETURN
      END
      FUNCTION COMEAN(SM,IS,ISP,COSLAP,SINLAP,COSLOP,SINLOP,
     *COSLAQ,SINLAQ,COSLOQ,SINLOQ,NSTEPP,NSTEPQ,LTABLE,
     *LCZERO,LTCOV)
C PROGRAMMED NOV 1985 BY C.C.TSCHERNING, GEODETIC INSTITUTE.
C THE SUBROUTINE COMPUTES MEAN VALUES OF COVARIANCES.
C CHANGED 2002-09-12.
      implicit none
      LOGICAL LOCAL,LSUM,LTABLE,LMEAP1,LMEAQ1,LTEST,LCZERO,LTCOV,
     *LFOURI,LLCOEE,LPARMP,LPARMQ,LFINI
C
      REAL*8 STEPN,COSSTN,SINSTN,STEPE,COSSTE,SINSTE,
     *COST2P,SINT2P,FILTER,STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE,
     *COST2Q,SINT2Q,CCI,CCR,SI,HCMAX,CCV,DC,SCFRDD,SCFACT,RDD,
     *FOUCOF,SM,COVME,RADEG,RLAT,RJ,SINLAP,SINLAQ,COVM,COLAP,
     *COSLAP,SILAP,COLOP,COSLOP,SILOP,RLAY,COLOQ,COSLOQ,SILOQ,RLAX,
     *COSDLO,T,RLOX,PSI,COV,COLOQ1,COLAQ1,COLOP1,COLAP1,COMEAN,COVCG,
     *COZERO,RLONG,SINLOP,SINLOQ,COLAQ,SILAQ,RLOY,COSLAQ
      INTEGER KVI,KP,KQ,NFOURI,I,NSTEPE,NSTEQE,NSTEPP,NSTEPQ,MLAP,
     *MLOP,J,IS,ISP,MLAQ,MLOQ,NFILTE
C
```

```
      COMMON /CMEAN/STEPN,COSSTN,SINSTN,STEPE,COSSTE,SINSTE,
     *COST2P,SINT2P,FILTER(11),NFILTE
      COMMON /CMEAQ/STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE,
     *COST2Q,SINT2Q
      COMMON /CMCOV/CCI(24),CCR(56),SI(4400),HCMAX,CCV(4),
     *DC(36),KVI(39),LOCAL,LSUM,LFINI
      COMMON /CLPARM/SCFRDD(42),SCFACT,RDD,FOUCOF(0:21),NFOURI,LFOURI,
     *LLCOEE(42)
      COMMON /CLPAR1/KP,KQ,LPARMP,LPARMQ
      DIMENSION SM(2200),COVME(4)
C
      LTEST=LTCOV
      RADEG=180.0/3.1415926535D0
C CCI(20)=1 INDICATES THAT NOT-SO PRECISE EQUATIONS WILL BE USED IN
C COVCX. 2002.10-30.
      CCI(20)=1.0D0
      RLAT=0.0D0
      RJ = 0.0D0
C     STEQE=5.0d0
      LMEAP1=STEPE.LT.1.0D-8
      LMEAQ1=STEQE.LT.1.0D-8
      IF (LTEST) WRITE(*,*)' STEPE,STEQE ',STEPE,STEQE
      NSTEPE=NSTEPP
      NSTEQE=NSTEPQ
      IF (LMEAP1) NSTEPE=1
      IF (LMEAQ1) NSTEQE=1
      IF (LTEST) write(*,*)'STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE',
     *STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE
      IF ( ABS(SINLAP-SINLAQ).GT.1.0D-8.OR. ABS(SINLOP-SINLOQ)
     *.GT.1.0D-8.OR.NSTEPP.EQ.1) GO TO 2999
      COSSQN=COSSTN
      COSSQE=COSSTE
      SINSQN=SINSTN
      SINSQE=SINSTE
C
 2999 COVM=0.0D0
      IF (LTEST) write(*,*)' LMP,Q,SPEN,SQEN ',LMEAP1,LMEAQ1,
     *STEPE,STEPN,STEQE,STEQN
      DO 3000 I=1,4
 3000 COVME(I)=0.0D0
C
      COLAP=COSLAP
      SILAP=SINLAP
C
      DO 3043 MLAP=1,NSTEPP
      CCR(4)=SILAP
      CCR(6)=COLAP
      IF (MLAP.EQ.1.OR.(.NOT.LMEAP1)) THEN
      COLOP=COSLOP
      SILOP=SINLOP
      END IF
C     IF (MLAP.EQ.1.AND.LMEAP1) THEN
C     CALL PAZIM(RLAT,RLONG,COLAP,SILAP,COLOP,SILOP,
C    *-COSSTE,-SINSTE,COST2P,SINT2P,LTEST)
C     END IF
      IF (LTEST) RLAY=ATAN2(SILAP,COLAP)*RADEG
C
      DO 3044 MLOP=1,NSTEPE
      COLAQ=COSLAQ
      SILAQ=SINLAQ
      IF (LTEST) THEN
      RLOY=ATAN2(SILOP,COLOP)*RADEG
      write(*,*)' LAP,LOP',RLAY,RLOY
      END IF
C
      DO 3045 MLAQ=1,NSTEPQ
      IF (MLAQ.EQ.1.OR.(.NOT.LMEAQ1)) THEN
      COLOQ=COSLOQ
      SILOQ=SINLOQ
```



```
      END IF
C     IF (MLAQ.EQ.1.AND.LMEAQ1) THEN
C     CALL PAZIM(RLAT,RLONG,COLAQ,SILAQ,COLOQ,SILOQ,
C    *-COSSQE,-SINSQE,COST2Q,SINT2Q,LTEST)
C     END IF
      CCR(5)=SILAQ
      CCR(7)=COLAQ
      IF (LTEST) RLAX=ATAN2(SILAQ,COLAQ)*RADEG
C
      DO 3046 MLOQ=1,NSTEQE
      COSDLO=COLOP*COLOQ+SILOP*SILOQ
      T=SILAQ*SILAP+COLAP*COLAQ*COSDLO
      IF (T.GT.1.0D0) T=1.0D0
      CCR(9)=COSDLO
      CCR(8)=-SILOP*COLOQ+COLOP*SILOQ
      CCR(1)=T
      IF (LTEST) THEN
      RLOX=ATAN2(SILOQ,COLOQ)*RADEG
      write(*,*)' LAQ,LOQ, T',RLAX,RLOX,T
      END IF
      IF (LCZERO) THEN
C FINITE COVARIANCE FUNCTIONS INTRODUCED MAY, 1996 BY CCT.
      PSI=ACOS(T)
      COV=SCFACT*COZERO(PSI,RDD,1)
      CCV(1)=COV
      ELSE
      IF (LTABLE) THEN
      COV=COVCG(SM,ISP,.FALSE.)
      ELSE
      CALL COVCX(SM,COV,IS,.FALSE.)
      IF (LTEST) WRITE(*,*)' COV= ',COV
      END IF
      END IF
C CORRECTION FOR LATITUDE FACTOR MADE DEC. 1996.
      IF (.NOT.LMEAP1.AND.(.NOT.LMEAQ1)) THEN
      DO 3001 I=1,4
 3001 COVME(I)=COVME(I)+CCV(I)*COLAP*COLAQ
      COVM=COVM+COV*COLAP*COLAQ
      RJ=RJ+COLAP*COLAQ
      ELSE
      IF (LMEAP1.AND.LMEAQ1) THEN
      COVM=COVM+COV*FILTER(MLAQ)*FILTER(MLAP)
      ELSE
      IF (LMEAQ1.AND.(.NOT.LMEAP1)) THEN
      COVM=COVM+COV*FILTER(MLAQ)*COLAP
      RJ=RJ+COLAP
      END IF
      IF (LMEAP1.AND.(.NOT.LMEAQ1)) THEN
      COVM=COVM+COV*FILTER(MLAP)*COLAQ
      RJ=RJ+COLAQ
      END IF
      END IF
      END IF
C
      IF (.NOT.LMEAQ1) THEN
      COLOQ1=COLOQ
      COLOQ=COLOQ*COSSQE-SILOQ*SINSQE
      SILOQ=SILOQ*COSSQE+COLOQ1*SINSQE
      END IF
 3046 CONTINUE
C
      IF (LMEAQ1) THEN
      CALL PAZIM(RLAT,RLONG,COLAQ,SILAQ,COLOQ,SILOQ,
     *COSSQE,SINSQE,COSSQN,SINSQN,.FALSE.)
      ELSE
      COLAQ1=COLAQ
      COLAQ=COLAQ*COSSQN+SILAQ*SINSQN
      SILAQ=SILAQ*COSSQN-COLAQ1*SINSQN
      END IF
```

```
 3045 CONTINUE
C
      IF (.NOT.LMEAP1) THEN
      COLOP1=COLOP
      COLOP=COLOP*COSSTE−SILOP*SINSTE
      SILOP=SILOP*COSSTE+COLOP1*SINSTE
      END IF
 3044 CONTINUE
C
      IF (LMEAP1) THEN
      CALL PAZIM(RLAT,RLONG,COLAP,SILAP,COLOP,SILOP,
     *COSSTE,SINSTE,COSSTN,SINSTN,.FALSE.)
      ELSE
      COLAP1=COLAP
      COLAP=COLAP*COSSTN+SILAP*SINSTN
      SILAP=SILAP*COSSTN−COLAP1*SINSTN
      END IF
 3043 CONTINUE
C
      J=(NSTEPP*NSTEPQ*NSTEPE*NSTEQE)
      IF (LMEAP1.AND.LMEAQ1) RJ = J
      COMEAN=COVM/RJ
      IF (.NOT.LMEAP1.AND.(.NOT.LMEAQ1)) THEN
      DO 3003 I=1,4
 3003 CCV(I)=COVME(I)/RJ
      ELSE
      CCV(1)=COVM/RJ
      END IF
      IF (LTEST) WRITE(*,*)' COMEAN, J, RJ ',COMEAN,J,RJ
C
      RETURN
      END
      SUBROUTINE BILDEC(DS,DT,C,A)
COMMENT GI REG.NO. 81020, PROGRAMMED SEPT. 1981 BY C.C.TSCHERNING.
C
C REFERENCE: SUENKEL, HANS: A COVARIANCE APPROXIMATION PROCE−
C DURE, OSU REP. 286, 1979, P. 32.
C
C FUNCTION:
C THE VALUES OF A FUNCTION OF TWO VARIABLES, F(S, T) AND ITS DERI−
C VATIVES FS=DF/DS, FT=DF/DT, FTS=D(2)F/DSDT GIVEN IN THE FOUR
C CORNERS OF A RECTANGLE WITH SIDE LENGTH DS, DT ARE USED TO COM−
C PUTE THE COEFFICIENTS OF A POLYNOMIAL, WHICH MAY BE USED TO
C REPRESENT THE FUNCTION OVER THE RECTANGLE, SEE REF. SECTION 4.
C
C  DS  (CALL VALUE, REAL)   SIDE LENGTH IN S,
C  DT  ( −   − , − )   −     −     T,
C  C   ( −   − , DIMENSION) BOUNDS (1:16) HOLDS THE DERIVATIVES:
C    IF K IS RELATED TO THE 4 CORNERS IN THE SEQUENCE (0, 0),
C    (DS, 0), (0, DT), (DS, DT), THEN C(J)=F, C(J+1)=FS, C(J+4)=
C    FT, C(J+5)=FTS, AND J=CASE K OF (1, 3, 9, 11).
C  A  (RETURN VALUES, DIMENSION) THE COEFFICIENTS OF THE POLYNOMIAL
C     IN A(1) − A(16). A(17) = DS, A(18) = DT
C
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE:
      implicit none
      real*8 B,A,C,DS,DT,E
      INTEGER I,J
      DIMENSION B(16),A(18),C(16)
C
      A(17)= DS
      A(18)= DT
C
      I=0
      DO 100 J=1,4
      IF (J.EQ.3)I=8
      I=I+2
      C(I)= C(I)*DS
      C(I+3)= C(I+3)*DT
```



```
 100 C(I+4)= C(I+4)*DS*DT
C
      DO 101 I=1,4
      B(2*I−1)= 2.0*(C(I+8)−C(I))
      B(2*I)= 1.5*B(2*I−1)
      E= C(I+4)
      B(2*I+7)= C(I+12)+E
 101 B(2*I+8)= B(2*I+7)+E
C
      A(1)=C(1)
      A(2)=C(2)
      A(5)=C(5)
      A(6)=C(6)
      A(9)= B(2)−B(10)
      A(13)= B(9)−B(1)
      A(10)= B(4)−B(12)
      A(14)= B(11)−B(3)
      A(4)= C(4)+C(2)−2.0*(C(3)−C(1))
      A(3)= C(3)−C(2)−C(1)−A(4)
      A(8)= C(8)+C(6)−2.0*(C(7)−C(5))
      A(7)= C(7)−C(6)−C(5)−A(8)
      E= B(6)−B(14)−A(9)
      A(12)= B(8)−B(16)+A(10)−2.0*E
      A(11)= E−A(10)−A(12)
      E= B(5)−B(13)+A(13)
      A(16)= 2.0*E−B(7)+B(15)+A(14)
      A(15)= −E−A(14)−A(16)
      RETURN
      END
      FUNCTION POL(I,A,T)
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE:
      implicit none
      integer I,J,K
      REAL*8 A,T,P,POL
      DIMENSION A(18)
      P=A(I+12)
      K=2
      DO 10 J=1,3
      P=P*T+A(I+4*K)
  10 K=K−1
      POL=P
      RETURN
      END
      FUNCTION DPOL(I,A,T2,T32)
C PROGRAMMED JUNE 1985 BY C.C.TSCHERNING. COMPUTES FIRST ORDER
C DERIVATIVE OF POLYNOMIAL USED IN BSFC.
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE:
      IMPLICIT NONE
      REAL*8 A,T2,T32,DPOL
      INTEGER I
C
      DIMENSION A(18)
      DPOL=A(I)+T2*(A(I+4)+T32*A(I+8))
      RETURN
      END
      FUNCTION BSFC(A,IS,IT,S,T)
C
COMMENT GI REG.NO.81021, PROGRAMMED BY C.C.TSCHERNING, SEP. 1981.
C FORTRAN VERSION JUNE 1985.
C
C REF.: SUENKEL, HANS: A COVARIANCE APPROXIMATION PROCEDURE,
C OSU REP. NO. 286, 1979, P. 42.
C
C FUNCTION:
C CALCULATION OF ZERO TO SECOND ORDER DERIVATIVES OF A
C BICUBIC POLYNOMIAL.
C
C PARAMETERS:
C  IS   (CALL VALUE, INTEGER) ORDER OF DERIVATIVE IN S,
```

```
C  IT  ( −   − ,  − )  −       −      T,
C  S   ( −   − , REAL) NORMALIZED COORDINATE OF THE POINT,
C  T   ( −   − ,  − )  −       −      − ,
C  A   ( −   − , DIMENSION) COEFFICIENTS OF THE POLYNOMIAL, E.G.
C              AS PRODUCED BY BILDEC IN A(1) − A(16).
C              A(17), A(18) HOLDS GRID SIZE IN S AND T.
C
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE:
      implicit none
      integer IS,IT,IT1,IS1,I,I1
      REAL*8 S,T,A,T2,T32,DS,DT,B,B1,B2,BSFC,POL,DPOL
C
      DIMENSION A(18)
      T2= 2.0D0*T
      T32= T*1.5D0
      DS= A(17)
      DT= A(18)
C
      IT1=IT+1
      IS1=IS+1
      GO TO (10,11,12),IT1
   10 B= POL(4,A,T)
C
      GO TO (20,21,22),IS1
C
   20 I = 3
      DO 30 I1 = 1,3
      B = POL(I,A,T)+B*S
   30 I = I−1
      GO TO 99
C
   21 B= POL(3,A,T)+B*S*1.5E0
      B= (POL(2,A,T)+B*S*2.0D0)/DS
      GO TO 99
C
   22 B= (POL(3,A,T)+B*S*3.0D0)*2.0D0/(DS*DT)
      GO TO 99
C
   11 B= DPOL(8,A,T2,T32)
C
      GO TO (31,32,33),IS1
C
   31 I = 7
      DO 34 I1 = 1, 3
      B = DPOL(I,A,T2,T32)+B*S
   34 I = I−1
      B= B/DT
      GO TO 99
C
   32 B1= DPOL(7,A,T2,T32)+B*S*1.5E0
      B2= DPOL(6,A,T2,T32)+B1*S*2.0D0
      B= B2/(DT*DS)
      GO TO 99
C
   33 B= DPOL(7,A,T2,T32)+B*S*3.0D0
      B= B*2.0D0/(DS*DS*DT)
      GO TO 99
C
   12 B= A(13)+S*(A(14)+S*(A(15)+S*A(16)))
      B= A(9)+S*(A(10)+S*(A(11)+S*A(12)))+B*T*3.0D0
      B= B*2.0D0/(DS*DT)
C
   99 BSFC= B
      RETURN
      END
C −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
      SUBROUTINE CTABEL(IP,LTEST)
C
COMMENT GI REG.NO. 81024, PROGRAMMED BY C.C.TSCHERNING, SEP. 1981.
```



```
C UPDATED JAN. 1983 BY CCT, FORTRAN VERSION JUNE 1985, LATEST UPDATE
C NOV 1991 BY CCT.
C
C FUNCTION:
C THE PROCEDURE WILL GENERATE A TABLE OF COVARIANCE VALUES
C IN A GRID WITH (NT+1)*(NS+1) POINTS. EACH KNOT IS ASSOCIATED WITH
C A VALUE OF T1=1−COS(SPHERICAL DISTANCE) AND S1=1−RB2/(RP*RQ),
C WHERE RB2 IS THE SQUARE OF THE RADIUS OF THE BJERHAMMAR−SPHERE
C AND RP, RQ ARE THE RADIAL DISTANCES OF THE POINTS P, Q FROM THE
C ORIGIN. THE TABLE IS USED TO GENERATE THE COEFFICIENTS OF BICUBIC
C POLYNOMIAL WITH 16 COEFFICIENTS, WHICH ARE STORED IN CTA.
C
C IN ORDER TO MAKE IT POSSIBLE TO HAVE SEVERAL TABELS STORED IN THE SAME
C COMMON AREA (FOR STEPVISE COLLOCATION, FOR EXAMPLE), POINTERS ARE USED
C TO POINT AT THE ZERO'TH ELEMENTS IN THE TABELS: CT − IC, CTSF − IS,
C CTTF − IT, SS − IZ, MAXB − IM, SIGMA AND SIGMA0 − II. THE POINTER
C VARIABLES WITH 1 − 6 ADDED POINTS AT THE 1 − 6 ELEMENTS. THE VALUES OF
C THESE POINTERS ARE KEPT IN COMMON, AND CHANGED IF IOLD IS DIFFERENT
C FROM IP. POINTERS FOR A NEW TABLE ARE RETURNED IN IX(IP*4+K), K=1,2,3
C OR 4.
C
C PARAMETERS:
C CALL VALUES:
C CI, KI, SIGMA0, AS SPECIFIED IN COVBX, SEE GI REG.NO. 76083.
C MAXB (INTEGER DIMENSION, BOUNDS (IM1:AT LEAST 12) WITH MAXB(1)=NS, THE
C NUMBER OF POINTS IN S, MAXB(IM2) THE NUMBER OF EQUIDISTANT INTERVALS
C IN PSI, MAXB(IM6+I) THE NUMBER OF SUBINTERVALS IN THE I'TH INTERVAL.
C SS   (DIMENSION, BOUNDS (1:MIN 6). THE VALUES OF SS DESCRIBES IN
C _     A CONDENSED FORM THE GRID. WE MUST HAVE:
C _     SS(IZ1) = MINIMUM HEIGHT (M), SS(IZ2) MAXIMAL HEIGHT (M),
C _     SS(IZ3) = 0.0, SS(IZ3+I) THE RIGHT−MOST INTERVAL BOUNDARY OF THE
C _     I'TH INTERVAL. A TYPICAL EXAMPLE IS (WITH IS=IM=0):
C _     SS(1)=0.0, SS(2)=2000.0, SS(4)=60.0, SS(5)= 360.0, SS(6)=1200.0.
C _     MAXB(1)=4, MAXB(2)=3, MAXB(7)=3, MAXB(8)=3, MAXB(9)=10.
C
C RETURN VALUES, (IN COMMON TABELC):
C MAXB(IM3)= NT1= TOTAL NUMBER OF T1 INTERVALS +1.
C CTA  (DIMENSION, BOUND ((NS+1)*NT1),16,2) THE COEFFICIENTS.
C CTSF (DIMENSION, BOUND .GE. NS), S1 VALUES.
C CTTF (DIMENSION, BOUND .GE. MAXB(3)), T1 VALUES.
C IX   (DIMENSION, BOUND .GE.6), NEW POINTERS.
C
      implicit none
      INTEGER NCTA
      PARAMETER (NCTA=1600)
C SMA PARAMETER (NCTA=100)
C 386 PARAMETER (NCTA=150)
      LOGICAL LOCAL,LSUM,LTEST,LF,LT,LFINI
      REAL*8 CT,SM,RR,U,RS,CX,CTA,CTTF,CTSF,SS,A,CI,CR,SIGMA0,
     *SIGMA,HMAX,CV,D,D0,D1,D2,D3,D4,D5,RE,RG,PI,GM,R5,R9,DB,R,PMIN,
     *TE,P,DP,SN,S0,T0,S1,T1,S,R2,G,G2,COV,RB2,HMIN,DH,H
      INTEGER KK,MAXB,IX,IC,IT,IT1,IS,IS1,IZ,IZ1,IZ2,IZ3,IM,IM1,IM2,IM3,
     *IM4,IM5,IM6,II,IOLD,KI,N1,N,ITCOUN,IP4,IP,NS,NS1,NT,I,K2,
     *J,K,NT1,KA,K0,K1,I1,J1,I0,J0,M,NN,J3,I5,NS2,N2
C
      DIMENSION CT(3200),SM(2200),RR(200),U(200),RS(8),CX(18)
      COMMON /CCVCG/KK(24)
      COMMON /TABELC/CTA(NCTA,16,2),CTTF(800),CTSF(20),SS(30),A(18),
     *MAXB(20),IX(8),IC,IT,IT1,IS,IS1,IZ,IZ1,IZ2,IZ3,IM,IM1,IM2,IM3,
     *IM4,IM5,IM6,II,IOLD
      COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CV(2,2),D(36),KI(37),N1,N2,LOCAL,LSUM,LFINI
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RG,PI,GM,ITCOUN,LF,LT
C
      IP4=IP*4
      IC=IX(1+IP4)
      IT=IX(2+IP4)
      IT1=IT+1
      IS=IX(3+IP4)
```

```
      IS1=IS+1
      IZ=IP*3
      IZ1=IZ+1
      IZ2=IZ+2
      IZ3=IZ+3
      IM=IP*6
      IM1=IM+1
      IM2=IM+2
      IM3=IM+3
      IM4=IM+4
      IM5=IM+5
      IM6=IM+6
      II=IX(4+IP4)
      IOLD=IP
C
      MAXB(IM4)=-1
      MAXB(IM5)=-1
      MAXB(IM6)=-1
      KI(37)=0
      KI(35)=0
      KI(36)=0
C KI(35) - (37) ARE USED TO KEEP TRACK OF HOW MANY TIMES COVCG IS
C CALLED WITH (37) AND WITHOUT (36) HAVING TO CALL BILDEC, AND WHEN IT
C WAS NECESSARY TO CALL COVCX (35).
      R9=1.0D9
      R5=1.0D5
      NS=IABS(MAXB(IM1))
      NS1=NS+1
      NT= MAXB(IM2)
C
C DB IS THE DIFFERENCE BETWEEN THE EARTH MEAN RADIUS (RE) AND THE RADIUS
C OF THE BJERHAMMAR SPHERE.
      DB=-CI(10)
      RB2=CI(9)
      HMIN= SS(IZ1)
      DH=(SS(IZ2)-HMIN)/NS
      DO 10 I=1,NS1
      H=DH*(I-1)+HMIN
      R=RE+H
      RR(I)=R
   10 CTSF(I+IS)= (D2*RE-DB+H)*(DB+H)/(R*R)
C
      PMIN=D0
      TE=D0
      K2=1
      CTTF(IT1)=D0
      U(1)=D0
      DO 20 J=1,NT
      K=MAXB(J+IM6)
C CONVERSION OF SPHERICAL DISTANCE TO UNITS OF RADIANS.
      P=SS(J+IZ3)/RG
      DP=(P-PMIN)/K
      PMIN=P
C
      DO 21 I=1,K
      K2=K2+1
      TE=TE+DP
      CTTF(K2+IT)=D2* SIN(TE/D2)**2
   21 U(K2)= SIN(TE)
   20 CONTINUE
      NT1=K2
      MAXB(IM3)=NT1
C
      DO 25 KA=1,4
      KI(6)=6
      IF (KA.GT.2) KI(6)=8
      KI(7)= KA
      N= (-1)**(KA+1)
C
```



```
      CALL COVBX(SM,.FALSE.,II)
      CR(6)=D1
      CR(11)=D1
      CR(10)=D1
      CR(4)=D0
      CR(8)=D0
      K0=KA
      IF (KA.GT.2)K0=K0+2
      K1= K0+2
      DO 26 I=1,NS1
      CR(2)= RR(I)-RE
      CR(3)= CR(2)
      SN=N
      IF (KA.EQ.2.OR.KA.EQ.4) SN=N/(D1-CTSF(I))
C
      DO 26 J=1,NT1
      CR(9)= D1
      CR(7)= D1-CTTF(J)
      CR(1)= CR(7)
C CR(1) AND CR(7) HOLDS COSINE OF THE SPHERICAL DISTANCE,
C HERE EQUAL TO COSINE OF THE LATITUDE.
      CR(5)= U(J)
C
      CALL COVCX(SM,COV,II,.FALSE.)
C
      CT(K0+8*(NT1*(I-1)+J-1))= CR(52)*SN
      CT(K1+8*(NT1*(I-1)+J-1))= -CR(53)*SN
C
   26 CONTINUE
   25 CONTINUE
C
C SET POINTERS FOR NEW COVARIANCE FUNCTION TABLES:
      IP4=IP4+4
      IX(IP4+1)=NT1*NS1+IC
      IX(IP4+2)=NT1+IT
      IX(IP4+3)=NS1+IS
C
      NT=NT1-1
      DO 50 KA=1,2
      K=0
      IF (KA.EQ.2)K=4
C
      DO 51 I=1,NS
      DO 51 J=1,NT
      S0=CTSF(I+IS)
      T0=CTTF(J+IT)
      I1=I+1
      J1=J+1
C
      DO 53 I0=I,I1
      DO 53 J0=J,J1
      M=((J0-J)*2+I0-I)+1
      K0=KK(M+16)
C
      DO 54 N=1,4
      NN=N+K+8*((I0-1)*NT1+J0-1)
   54 CX(K0+KK(N+20))=CT(NN)
   53 CONTINUE
C
      S1=CTSF(I+IS1)-S0
      T1=CTTF(J+IT1)-T0
C
      CALL BILDEC(S1,T1,CX,A)
C
      J3=IC+(I-1)*NT1+J
      DO 55 I5=1,16
   55 CTA(J3,I5,KA)=A(I5)
C
   51 CONTINUE
```

```
   50 CONTINUE
C
      IF (.NOT.LTEST) GO TO 99
C
      WRITE(6,100)(CTSF(I+IS),I=1,NS1)
  100 FORMAT('0 1-S=',8F7.4,/,6X,8F7.4)
      WRITE(6,101)(CTTF(I+IT),I=1,K2)
  101 FORMAT('0 1-T=',5(1X,F10.9),10(/,6X,5(1X,F10.9)))
      WRITE(6,102)
  102 FORMAT(' (1,1) (1,2) (1,6) (2,6) (3,3) (3,4)',
     *' (3,8) (4,8) ')
C
      NS2=NS1
      IF (MAXB(IM1).LT.0) NS2=1
      K=1
      DO 30 I=1,NS2
      DO 30 J=1,K2
      S= D1-CTSF(I+IS)
      R= RR(I)
      R2= R*R
      G= R2/GM
      G2= G*G
C
      RS(1)=CT(K)*G2
      RS(2)=CT(K+1)*S*R9*G/R2
      RS(3)=CT(K+2)*RG*G2/R*U(J)
      RS(4)=CT(K+3)*S*R9*RG*G/(R2*R)*U(J)
      RS(5)=CT(K+4)*R5*R5/R2
      RS(6)=CT(K+5)*S*R5*R9/(R2*R)
      RS(7)=CT(K+6)*R9*R5/(R2*R)*U(J)
      RS(8)=CT(K+7)*S*R9*R9/(R2*R2)*U(J)
      K=K+8
      WRITE(6,105)RS
  105 FORMAT(1X,8F9.3)
   30 CONTINUE
   99 RETURN
      END
      FUNCTION COVCG(SM,IP,LTEST)
C
COMMENT GI REG.NO.81025, PROGRAMMED BY C.C.TSCHERNING, SEPT. 81.
C FORTRAN VERSION JUNE 1985, LATES UPDATE NOV 1991 BY CCT, WHERE
C LINEAR INTERPOLATION IS USED INSTEAD OF BICUBIC FOR THE
C CALCULATION OF CROS-COVARIANCE GRAVITY - DEFLECTIONS.
C
C FUNCTION:
C THE PROCEDURE WILL FIND IN CTA (1) THE COEFFICIENTS OF A BICUBIC
C POLYNOMIUM REPRESENTING A COVARIANCE FUNCTION IN A RECTANGLE
C WITH CORNER COORDINATES (CTSF(I), CTTF(J)) (LOWER, LEFTMOST)
C (CTSF(I+1), CTTF(J+1)) (UPPER, RIGHTMOST), IN A COORDINATE
C SYSTEM WITH S1 AS ABSCISSA AND T1 AS ORDINATE, (2) COMPUTE
C THE COVARIANCE FOR A POINT WITHIN THE RECTANGLE.
C
C HERE S1=1-S, S=RB**2/(RP*RQ), T1=1-T=1-COS(SPHERICAL DIST.),
C WHERE RB IS THE RADIUS OF THE BJERHAMMAR-SPHERE, RP, RQ ARE
C THE RADIAL DISTANCES OF TWO POINTS P AND Q, SEE THE PROCEDURE
C COVCX.
C
C THE COEFFICIENTS ARE STORED IN THE ARRAY A. IN MAXB ARE
C CURRENT VALUES OF I AND J STORED, TOGETHER WITH AN INTEGER K
C INDICATING WHICH TYPE OF COVARIANCE FUNCTION IS STORED (COVA-
C RIANCE(T, T) FOR K=0 AND COV(DELTA G, DELTA G) FOR K=3).
C THIS MAKES THE RECOLLECTION OF THE ELEMENTS OF A UNNECESSARY
C IF THE PROCEDURE IS CALLED IN ORDER TO COMPUTE VALUES WITHIN
C THE SAME RECTANGLE ANOTHER TIME.
C
C IF THE COVARIANCES NEEDED FOR THE INTERPOLATION HAVE NOT BEEN
C TABULATED (IN THE ARRAY CT), THEN COVCX WILL BE CALLED.
C
C PARAMETERS:
```



```
C SEE COVCX, COVBX, CTABLE FOR KI, CI, SIGMA, CR, CT AND D.
C A  (DIMENSION, BOUNDS (18)), HOLDS THE COEFFICIENTS (1-16),
C _   AND SIDE LENGTHS IN A(17), A(18).
C
C EXTNAL PROCEDURES USED: COVCX, BSFC.
C
      IMPLICIT NONE
      INTEGER NCTA
      PARAMETER (NCTA=1600)
C SMA PARAMETER (NCTA=100)
C 386 PARAMETER (NCTA=150)
      LOGICAL LFAST,LTEST,LT,LSUM,LOCAL,LF,LFINI
      INTEGER KI,N1,N2,ITCOUN,KK,MAXB,IX,IC,IT,IT1,IS,IS1,IZ,IZ1,IZ2,
     *IZ3,IM,IM1,IM2,IM3,IM4,IM5,IM6,II,IOLD,IP,IP4,NS,NT,NT1,NS1,
     *J,JJ,JD,NDT,NDS,K,KA,I2,J0,I,KP,KQ,NT0,I0
C
      REAL*8 CI,CR,SIGMA0,SIGMA,HMAX,CVV,D,D0,D1,D2,D3,D4,D5,RE,RG,
     *PI,GM,CTA,CTTF,CTSF,SZ,A,SM,R5,DB,SP,SQ,CP,CQ,CD,T,T1,HP,HQ,
     *RP,RQ,S,S1,S0,T0,DS,DT,C,DCT,DCS,DCSTY,DD00,DD01,DD10,DD11,DCST,
     *SD,SS,DCTX,CS,SC,SCC,CSC,CF,COV,COVCG,BSFC
C
      COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CVV(2,2),D(36),KI(37),N1,N2,LOCAL,LSUM,LFINI
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RG,PI,GM,ITCOUN,LF,LT
      COMMON /CCVCG/KK(24)
C ELEMENTS OF KK ARE INITIALIZED BY BLOCK DATA ROUTINE.
      COMMON /TABELC/CTA(NCTA,16,2),CTTF(800),CTSF(20),SZ(30),A(18),
     *MAXB(20),IX(8),IC,IT,IT1,IS,IS1,IZ,IZ1,IZ2,IZ3,IM,IM1,IM2,IM3,
     *IM4,IM5,IM6,II,IOLD
      DIMENSION SM(2200)
C
      IF (IOLD.EQ.IP) GO TO 10
      IP4=IP*4
      IC=IX(1+IP4)
      IT=IX(2+IP4)
      IT1=IT+1
      IS=IX(3+IP4)
      IS1=IS+1
      IZ=IP*3
      IZ1=IZ+1
      IZ2=IZ+2
      IZ3=IZ+3
      IM=IP*6
      IM1=IM+1
      IM2=IM+2
      IM3=IM+3
      IM4=IM+4
      IM5=IM+5
      IM6=IM+6
      II=IX(4+IP4)
      IOLD=IP
      MAXB(IM6)=-1
C THIS ASSIGNMENT OF -1 IS DONE TO ASSURE THAT BILDEC IS CALLED WHEN A
C A NEW COVARIANCE FUNCTION TABEL IS USED.
C
   10 R5=1.0D5
      DB=-CI(10)
      NS=IABS(MAXB(IM1))
      NT=MAXB(IM2)
      NT1=MAXB(IM3)
      NS1=NS+1
C
      SP= CR(4)
      SQ= CR(5)
      CP= CR(6)
      CQ= CR(7)
C
      SS= SP*SQ
      CD=CR(9)
```

```
      T= CR(1)
      T1=D1-T
      HP= CR(2)
      HQ= CR(3)
C
      RP= RE+HP
      RQ= RE+HQ
      S= CI(9)/(RP*RQ)
      S1= D1-S
C
      IF (KI(6).EQ.0.OR.KI(7).EQ.0) GO TO 90
      KP=KK(KI(6))
      KQ=KK(KI(7))
      LFAST=(T1.LE.CTTF(NT1+IT)).AND.(S1.GE.CTSF(IS1)).AND.(S1.LE.
     *CTSF(NS1+IS)).AND.(KP.LT.5).AND.(KQ.LT.5).AND.KI(8).LT.2
      IF (.NOT.LFAST) GO TO 90
C
      IF ( ABS(HP-HQ) .GE. 1.0D-5) GO TO 11
      IF ( ABS(HP) .LT. 1.0D-5) S1=DB*(D2-DB/RE)/RE
      IF ( ABS(HP).GT.1.0D-5) S1= (HP+DB)*(2*RE+HP-DB)/RP**2
      GO TO 12
   11 S1= (RE*(2*DB+HP+HQ)+HP*HQ-DB*DB)/(RP*RQ)
   12 S= D1-S1
C
COMMENT FIND INDEX FOR LOWER LEFT CORNER
      I=0
   14 I=I+1
      IF (CTSF(I+IS1) .LT. S1) GO TO 14
      J=0
      IF (.NOT.LTEST) GO TO 19
   13 J=J+1
      IF (CTTF(J+IT1) .LT. T1) GO TO 13
   19 NT0=NT1-1
      JJ=J
      J=1
      IF (T1.LE.D0) GO TO 22
      IF (T1.LT.CTTF(NT0+IT)) GO TO 23
      J=NT0
      GO TO 22
   23 J=NT0/2+1
      JD=NT0
   24 JD=JD/2
      IF (JD.LT.1) JD=1
      IF (T1.GE.CTTF(J+IT)) GO TO 25
      J=J-JD
      GO TO 24
   25 IF (T1.LT.CTTF(J+IT1)) GO TO 22
      J=J+JD
      GO TO 24
   22 IF (J.NE.JJ.AND.LTEST) WRITE(6,98)J,JJ
   98 FORMAT(' J,JJ=',2I4)
      S0= CTSF(I+IS)
      T0= CTTF(J+IT)
C
      DS= S1-S0
      DT=T1-T0
C
      IF (LTEST) WRITE(6,100)I,J,T1,S1,DT,DS
  100 FORMAT(' I,J,T1,S1,DT,DS=',2I4,4F12.9)
      NDT=0
      IF (KP.GT.2) NDT=1
      IF (KQ.GT.2) NDT=NDT+1
      NDS=0
      IF (KP.EQ.2)NDS=1
      IF (KQ.EQ.2)NDS=NDS+1
      IF (NDS.NE.2) GO TO 15
      NDS= 0
      K= 3
      GO TO 16
```



```
   15 K=-1
C
   16 IF (I.EQ.MAXB(IM4).AND.J.EQ.MAXB(IM5).AND.K.EQ.MAXB(IM6)) GO TO 18
      MAXB(IM4)=I
      MAXB(IM5)=J
      MAXB(IM6)=K
      KA=1
      IF (K.EQ.3)KA=2
      I2=IC+(I-1)*NT1+J
      DO 20 J0=1,16
   20 A(J0)=CTA(I2,J0,KA)
C
      S1=CTSF(I+IS1)-S0
      T1=CTTF(J+IT1)-T0
      A(17)=S1
      A(18)=T1
      IF (LTEST) WRITE(6,102)(A(I0),I0=1,16)
  102 FORMAT(' A=',4E15.6,3(/,3X,4E15.5))
      KI(37)= KI(37)+1
      GO TO 21
C
   18 KI(36)= KI(36)+1
C
   21 S1=A(17)
      T1=A(18)
      DS=DS/S1
      DT= DT/T1
C
      C=D0
      DCT=D0
      IF (NDS.EQ.0) GO TO 29
      IF (NDT.GT.0) GO TO 28
C
C COVARIANCE GRAVITY AND HEIGHT ANOMALY.
      C=BSFC(A,0,0,DS,DT)
      DCS= S*BSFC(A, 1, 0, DS, DT)
      C= -(DCS+D2*C)
      CR(52)=C
      GO TO 31
C
C COVARIANCE GRAVITY ANOMALY WITH DEFLECTIONS.
   28 DCTX= BSFC(A, 0, 1, DS, DT)
      IF (LTEST) DCSTY= BSFC(A, 1, 1, DS, DT)
C CHANGE HERE MADE MAY 1988, USING LINEAR INTERPOLATION INSTEAD
C OF BICUBIC INTERPOLATION. THIS MAY NOT BE NECESSARY ON COMPUTERS
C USING DOUBLE PRECISION ARTIHMETIC. DDXY HOLDS THE CORNER VALUES
C OF THE MIXED FIRST ORDER DERIVATIVES IN THE SQUARE WITH SIDE-
C LENGTHS EQUAL TO 1.0 .
      DD00= A(6)/(A(17)*A(18))
      DD01= BSFC(A,1,1,D0,D1)
      DD10= BSFC(A,1,1,D1,D0)
      DD11= BSFC(A,1,1,D1,D1)
      DCST= (DD00*(D1-DS)+DD10*DS)*(D1-DT)+(DD01*(D1-DS)+DD11*DS)*DT
      IF (LTEST) WRITE(6,9765)DD00,DD01,DD10,DD11,DS,DT,DCST,DCSTY
 9765 FORMAT(' DD00011011 ',4E15.7,/,' DSDTSTSTY ',4E15.7)
      DCT= DCST*S+2*DCTX
      CR(53)=DCT
      GO TO 31
C
   29 IF (NDT.GT.0) GO TO 30
C AUTOCOVARIANCE OF GRAVITY OR OF HEIGHT ANOMALY.
      C=BSFC(A,0,0,DS,DT)
      CR(52)=C
      GO TO 31
C
C CROSS-COVARIANCE HEIGHT ANOMALY WITH DEFLECTIONS.
   30 DCT=-BSFC(A,0,1,DS,DT)
      CR(53)=DCT
   31 CONTINUE
```

```
C
      IF (NDT.EQ.0) GO TO 40
      SD= CR(8)
      CS= CP*SQ
      SC= SP*CQ
      SCC= SC*CD
      CSC= CS*CD
      D(3)=CQ*SD
      D(13)= -CP*SD
      D(2)= CS-SCC
      D(7)= SC-CSC
C
   40 IF (LTEST) WRITE(6,110)C,RP,RQ,CI(11),CR(10),CR(11)
  110 FORMAT(' C,RP,RQ,CI11,GP,GQ=',/,
     *E13.6,2F9.1,E13.6,2F9.6)
C
      CF=CI(11)/(RP**KI(22)*RQ**KI(23)*CR(10)**KI(20)
     **CR(11)**KI(21))
      CI(12)=CF
      C=C*CF
      DCT=DCT*CF
      IF (NDT.GT.0) GO TO 71
      CVV(1,1)=C
      GO TO 72
   71 IF (KP.GT.2) GO TO 73
C COVARIANCE WITH DEFLECTION IN Q AND HEIGHT OR GRAVITY ANOMALY IN P.
      CVV(1,1)= DCT*D(13)
      CVV(1,2)= DCT*D(7)
      GO TO 72
C COVARIANCE WITH DEFLECTION IN P AND HEIGHT OR GRAVITY ANOMALY IN Q.
   73 CVV(1,1)=DCT*D(3)
      CVV(2,1)=DCT*D(2)
   72 COVCG=CVV(KI(24),KI(25))
C
      GO TO 99
   90 CALL COVCX(SM,COV,II,.FALSE.)
      COVCG=COV
      KI(35)= KI(35)+1
C
   99 RETURN
      END
C ---------------------------------------------------------------
      SUBROUTINE INTABH(SM,IS,LTEST)
C PROGRAMMED SEPT 1987 BY C.C.TSCHERNING. LATEST CHANGE 4 DEC 87.
C
C THE SUBROUTINE INITIALIZES TABELS FOR COVARIANCE-FUNCTION
C INTERPOLATION FOR FIXED HEIGHTS AND FUNCTIONALS.
C
C CALL VALUES:
C SM, IS PARAMETERS USED BY COVBX AND COVCX. SM IS A DUMMY
C REAL ARRAY WITH AT LEAST 3 ELEMENTS AND IS IS THE SUBSCRIPT
C OF THE FIRST ELEMENT USED IN SIGMA0 AND SIGMA, WHICH HOLDS
C THE DEGREE-VARIANCE CORRECTIONS. (SEE COVAX).
C
C OTHER CALL VALUES ARE STORED IN COMMON BLOCK CTABH:
C HTA  - HOLD THE UP TO 5 FIXED HEIGHTS IN METERS,
C SIZEI- THE INTERVAL SIZE IN ARCSECONDS (AT CALL),
C NFU  - THE NUNBER OF FIXED FUNCTIONALS FOR EACH FIXED HEIGHT,
C KEYH - THE FUNCTIONAL TYPES (0 - 15), SEE COVAX,
C NINTH- NUMBER OF INTERVALS IN TABLE,
C NHE  - NUMBER OF FIXED HEIGHTS.
C
C RETURN VALUES:
C IN CHTA THE COVARIANCE TABLES,
C TMAX  - THE MAXIMAL VALUE OF T=COS(SPHERICAL DISTANCE)
C NTABH - THE LOGICAL TABLE NUMBER FOR EACH HEIGHT AND
C         AND FUNCTIONAL. THERE WILL NORMALLY BE SEVERAL
C         TABLES FOR EACH FUNCTIONAL, SUCH AS BOTH THE FIRST,
C         SECOND AND THIRD DERIVATIVE WITH RESPECT TO T.
```



```
C SIZEI - NOW IN UNITS OF RADIANS.
C
      IMPLICIT NONE
      INTEGER NCTA
      PARAMETER (NCTA=51200)
C SMA PARAMETER (NCTA= 3200)
C 386 PARAMETER (NCTA= 4800)
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE THE FOLLOWING:
      REAL*8 CCI,CR,SIGMA0,SIGMA,HMAX,CCV,D,
     *CHTA,CTTF,CTSF,SZ,AX,RC,HTA,TMAX,SIZEI,A,S,RB2,T,B,
     *D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GMC,SM,TT,COV,HP,HQ,CR50
C
      INTEGER KI,NC1,N2,MAXB,IX,IIX,IOLD,NFU,KEYH,NINTH,NTABH,
     *NHE,NSTART,KT,KT1,K,I1,JJD,N3,KK,KQ,KP,ND,NR,ND1,ND2,ITCOUN,
     *NTAB,NINTH1,NINTH2,I,JI,J,II,JJ,II1,NTT,IK,NN,IS
C
      LOGICAL LTABH,LOCAL,LSUM,LOLDFU,LTEST,LT,LF,LFINI
C
      COMMON /CMCOV/CCI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CCV(2,2),D(36),KI(37),NC1,N2,LOCAL,LSUM,LFINI
C CMCOV CONTAINS VARIABLES USED BY COVAX, COVBX AND COVCX.
      COMMON /TABELC/CHTA(NCTA),CTTF(800),CTSF(20),SZ(30),AX(18),
     *MAXB(20),IX(8),IIX(17),IOLD
C TABELC CONTAINS VARIABLES USED BY CTABLE AND COVCG.
      COMMON /CTABH/RC(1200),HTA(5),TMAX,SIZEI,NFU(5),KEYH(5,5),
     *NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
      COMMON /DDY/A,S,RB2,T,B,KT,KT1,K,I1,JJD,N3,KK,KQ,KP,ND,NR,ND1,
     *ND2
C DDY CONTAINS VARIABLES USED BY COVBX AND COVCX. HERE ONLY ND2
C EQUAL TO THE NUMBER OF DERIVATIVES WITH RESPECT TO T IS USED.
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GMC,ITCOUN,LF,LT
      DIMENSION SM(2200),TT(500)
C
      LTABH=LF
C CREATION OF TABLES. NTAB COUNTS NUMBER OF TABLES.
      NTAB=0
      SIZEI=SIZEI/RADSEC
      TMAX= COS(NINTH*SIZEI)
      NINTH1=NINTH+1
      NINTH2=NINTH1+1
      DO 210 I=1,NINTH1
  210 TT(I)= COS((I-1)*SIZEI)
C
      DO 300 I=1,NHE
      HP=HTA(I)
      CR(2)=HP
      JI=((I-1)*I)/2
C
      DO 301 J=1, I
      HQ=HTA(J)
      CR(3)=HQ
      DO 301 II=1,NFU(I)
      KI(6)=KEYH(II,I)
C IF THE FUNCTIONAL IS ONE OF A PAIR (LIKE A PAIR OF
C OF DEFLECTIONS) THE TABLE FOR ONLY ONE OF THE TWO IS MADE.
      IF (KI(6).GT.16) KI(6)=KI(6)-10
      IF (KI(6).EQ.13)KEYH(JJ,I)=15
      IF (KI(6).EQ.11)KEYH(JJ,I)=10
      IF (KI(6).EQ.7)KEYH(JJ,I)=6
C
      II1=II
      IF (I.NE.J) II1=NFU(J)
      DO 302 JJ=1,II1
      KI(7)=KEYH(JJ,J)
      IF (KI(7).GT.16) KI(7)=KI(7)-10
      CALL COVBX(SM,.FALSE.,IS)
C
C DEPENDING OF FUNCTIONAL TYPE, THE DERIVATIVES WITH RESPECT TO
C T FROM NSTART-2 TO ND2-2 MYST BE STORED, SEE COVCX.
```

```
      NSTART=2
      IF (ND1.EQ.1) GO TO 305
      LOLDFU=KP.EQ.12.OR.KP.EQ.14.OR.KQ.EQ.12.OR.KQ.EQ.14
      NSTART=3
      IF (LOLDFU) GO TO 305
      NSTART=4
      IF (KI(10).GT.1.AND.KI(11).GT.1.OR.ND2.EQ.3) NSTART=3
  305 CONTINUE
      NTABH(JI+J,II,JJ)=NTAB
      NTT=ND2-NSTART+1
C NTT IS TOTAL NUMBER OF DERIVATIVES WRT T, WHICH MUST BE STORED.
C
      DO 303 IK=1, NINTH1
      CR(1)=TT(IK)
      CALL COVCX(SM,COV,IS,.FALSE.)
      IF (LTEST)WRITE(6,11)IK,CR(1),COV,(CR(NN+50+NSTART-1),NN=1,NTT)
   11 FORMAT(' IK,T,COV=',I3,F10.7,F9.3,/,5E14.7)
C
C THE CALL OF COVCX GIVES ALL DERIVATIVES, BUT NOT ALL NEED TO
C BE STORED.
      DO 304 NN=1,NTT
      CR50=CR(50+NN+NSTART-1)
      IF (IK.EQ.2) CHTA((NTAB+NN-1)*NINTH2+1)=CR50
  304 CHTA((NTAB+NN-1)*NINTH2+IK+1)= CR50
  303 CONTINUE
      IF (LTEST) WRITE(6,50)NTAB,ND2,NSTART,II,JJ,KI(6),KI(7)
   50 FORMAT(' NTAB,ND2,NSTART,II,JJ,KP,KQ=',7I4)
      NTAB=NTAB+ND2-NSTART+1
  302 CONTINUE
  301 CONTINUE
  300 CONTINUE
      LTABH=.TRUE.
      RETURN
      END
C ----------------------------------------------------------------------
      SUBROUTINE TABH(C,LTA)
C PROGRAMMED BY C.C.TSCHERNING, SEPT 1987.
C UPDATE 2002-10-25 (IMPLICIT NONE).
C
C THE SUBROUTINE INTERPOLATES IN A COVARIANCE FUNCTION TABLE
C USING SPLINES, AFTER FIRST HAVING CHECKED WHETHER THE NECESSARY
C TABLES ARE THE ONES CREATED. OTHERWISE COVCX IS CALLED.
C
C CALL VALUES:
C THE TABLE VALUES ARE STORED IN CHTA OF COMMON TABELC.
C THE VARIABLES OF COMMON CTABH AS INITIALIZED BY INTABH.
C
C CALL AND RETURN VARIABLES:
C IF THE SAME FIXED FUNCTIONALS AND HEIGHT AS IN LAST CALL
C ARE USED, THEN TABEL VALUES IN RC ARE USED FOR THE
C THE SPLINE INTERPOLATION. OTHERWISE THEY ARE COMPUTED BY
C ISPCOV.
C THE VARIABLES IN CTABH1 KEEPS TRACK OF THE LAST HEIGHT AND
C FUNCTIONAL TYPES USED.
C
C RETURN VARIABLES:
C C   - ARRAY CONTAINING THE DERIVATIVES WITH RESPECT TO T.
C LTA - LOGICAL, TRUE IF IT WAS POSSIBLE TO USE THE TABLES.
C       OTHERWISE COVCX WILL COMPUTE THE NEEDED QUANTITIES.
C
      implicit none
      INTEGER NCTA,NFU,KEYH,NINTH,NTABH,NHE,NSTART,N,
     *NC1,N2,KI,MAXB,IX,IIX,IOLD,KT,KT1,K,II,JJ,N3,NINTH1,NINTH2,JI,
     *NR,ND1,ND2,KK,KQ,KP,ND,IHP,IHQ,KFP,KFQ,IFP,IFQ,NTAB,NN,ITCOUN
      PARAMETER (NCTA=51200)
C SMA PARAMETER (NCTA= 3200)
C 386 PARAMETER (NCTA= 4800)
      REAL*8 RC,HTA,TMAX,SIZEI,CCI,CR,SIGMAO,SIGMA,HMAX,
     *CCV,D,CHTA,CTTF,CTSF,SZ,AX,A,S,RB2,T,B,HPOLD,HQOLD,
```



```
     *D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GMC,C,HP,HQ,T1,PSI,X1,COVS,
     *SPLCOV
C
      LOGICAL LTA,LTABH,LOCAL,LSUM,LSAHP,LSAHQ,LSAFP,LSAFQ,LSWI,
     *LOLDFU,LTEST,LF,LT,LFINI
C
      COMMON /CTABH/RC(1200),HTA(5),TMAX,SIZEI,NFU(5),KEYH(5,5),
     *NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
C SEE INTABH FOR DESCRIPTION OF VARIABLES.
      COMMON /CMCOV/CCI(24),CR(56),SIGMAO(2200),SIGMA(2200),HMAX,
     *CCV(2,2),D(36),KI(37),NC1,N2,LOCAL,LSUM,LFINI
C SEE COVAX, COVBX AND COVCX FOR DESCRIPTION OF VARIABLES.
      COMMON /TABELC/CHTA(NCTA),CTTF(800),CTSF(20),SZ(30),AX(18),
     *MAXB(20),IX(8),IIX(17),IOLD
C CHTA CONTAINS THE TABLES.
      COMMON /DDY/A,S,RB2,T,B,KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,
     *NR,ND1,ND2
C VARIABLES USED IN COVBX AND COVCX.
      COMMON /CTABH1/HPOLD,HQOLD,IHP,IHQ,KFP,KFQ,IFP,IFQ,NTAB,NN
C VARIABLES KEEPS TRACK OF LAST HEIGHTS AND FUNCTIONALS USED.
      COMMON /DCON/ D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GMC,ITCOUN,LF,LT
C CONSTANTS INITIALIZED BY BLOCK DATA.
      DIMENSION C(6)
C
      LTEST=LF
      IF (T.LT.TMAX) GO TO 600
      NINTH1=NINTH+1
      NINTH2=NINTH1+1
      HP= CR(2)
      HQ= CR(3)
      KP=KI(6)
      IF (KP.EQ.7.OR.KP.EQ.11)KP=KP-1
      IF (KP.EQ.13)KP=15
      KQ=KI(7)
      IF (KQ.EQ.7.OR.KQ.EQ.11)KQ=KQ-1
      IF (KQ.EQ.13)KQ=15
      LSAHP =  ABS(HP-HPOLD).LT. 0.1
      LSAHQ =  ABS(HQ-HQOLD).LT. 0.1
      LSAFP=KP.EQ.KFP
      LSAFQ=KQ.EQ.KFQ
C
C CHECK, WHETHER WE STILL HAVE THE SAME HEIGHT AND FUNCTIONALS
C AS DURING THE LAST CALL.
      IF (LSAHP.AND.LSAHQ.AND.LSAFP.AND.LSAFQ) GO TO 500
C
C THEN FIND POINTERS TO ACTUAL HEIGHTS AND FUNCTIONALS.
      IF (LSAHP) GO TO 100
      IHP=0
  101 IHP=IHP+1
      IF (IHP.GT.NHE) GO TO 600
      IF ( ABS(HP-HTA(IHP)).GT.0.1) GO TO 101
      HPOLD=HP
C
  100 IF (LSAHQ) GO TO 102
      IHQ=0
  103 IHQ=IHQ+1
      IF (IHQ.GT.NHE) GO TO 600
      IF ( ABS(HQ-HTA(IHQ)).GT. 0.1) GO TO 103
      HQOLD=HQ
C
  102 LSWI=IHQ.GT.IHP
      IF (LSWI) JI=((IHQ-1)*IHQ)/2+IHP
      IF (.NOT.LSWI) JI= ((IHP-1)*IHP)/2+IHQ
C JI POINTS AT TABLE INDEXES.
C
      IFP=0
  104 IFP=IFP+1
      IF (IFP.GT.NFU(IHP)) GO TO 600
      IF (KP.NE.KEYH(IFP,IHP)) GO TO 104
```

```
      KFP=KP
      IFQ=0
  105 IFQ=IFQ+1
      IF (IFQ.GT.NFU(IHQ)) GO TO 600
      IF (KQ.NE.KEYH(IFQ,IHQ)) GO TO 105
      KFQ=KQ
C
      IF (LSWI.OR.IHP.EQ.IHQ.AND.IFP.LE.IFQ) NTAB=NTABH(JI,IFQ,IFP)
      IF (.NOT.(LSWI.OR.IHP.EQ.IHQ.AND.IFP.LE.IFQ)) NTAB=
     *                                        NTABH(JI,IFP,IFQ)
C
C FIND WHICH DERIVATIVES WRT T ARE STORED.
      NSTART=2
      IF (ND1.EQ.1) GO TO 110
      LOLDFU=KP.EQ.12.OR.KP.EQ.14.OR.KQ.EQ.12.OR.KQ.EQ.14
      NSTART=3
      IF (LOLDFU) GO TO 110
      NSTART=4
      IF (KI(10).GT.1.AND.KI(11).GT.1.OR.ND2.EQ.3) NSTART=3
  110 CONTINUE
C
      NN=ND2-NSTART+1
C CREATE TABLES NECESSARY FOR THE SPLINE INTERPOLATION.
      IF (LTEST)
     *WRITE(6,10)IHP,IHQ,IFP,IFQ,KFP,KFQ,NTAB,NSTART,ND2
   10 FORMAT(' IHP,IHQ,IFP,IFQ,KFP,KFQ,NTAB,NSTART,ND2=',9I3)
      DO 130 N=1,NN
  130 CALL ISPCOV(NTAB,N)
      KI(37)=KI(37)+1
      GO TO 502
C
  500 KI(36)=KI(36)+1
C
  502 T1= ABS(D1-T)
      IF (T1.LT.2.0D-10) PSI=0.0D0
      IF (T1.GE.2.0D-10) PSI=PI/2.0D0-ASIN(T)
      IF (PSI.LT.D0) PSI=D0
      X1=PSI/SIZEI+D2
C SPLINE INTERPOLATION OF ALL NEEDED DERIVATIVES WRT T.
      DO 501 N=1,NN
      COVS=SPLCOV(X1,NTAB,N)
  501 C(N+NSTART-1)=COVS
      IF (LTEST) WRITE(6,15)PSI*RADSEC,T,C(NSTART)*CCI(12)
   15 FORMAT(' PSI,T,COV=',F10.1,F13.10,E15.5)
      LTA=.TRUE.
      RETURN
C
  600 LTA=.FALSE.
      KI(35)=KI(35)+1
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE COVAX(SM,IS)
C ORIGINAL VERSION PROGRAMMED JULY 1975 BY C.C.TSCHERNING. LATEST
C MODIFICATION 1999-02-14.
C
C THIS SUBROUTINE PREPARES CONSTANTS USED FOR COVARIANCE FUNCTION EVALU-
C ATION, WHICH IS EXECUTED USING THE SUBROUTINES COVBX AND COVCX.
C
C THE COVARIANCE FUNCTION USED IS DEFINED ACCORDING TO A DEGREE-VARIANCE
C MODEL AND A SET OF EMPIRICAL (POTENTIAL) DEGREE-VARIANCES. THE DEGREE-
C VARIANCE MODEL IS SPECIFIED THROUGH THE VALUES OF KI(1)-KI(5),CI(8)-
C CI(10) AND THE PARAMETERS N1 AND LOCAL OCCURRING IN THE COMMON BLOCK
C /CMCOV/. EMPIRICAL ANOMALY DEGREE-VARIANCES WILL HAVE TO BE STORED IN
C SIGMA0 WHEN LOCAL IS FALSE, AND ARE USED FOR THE COMPUTATION OF RESI-
C DUAL POTENTIAL DEGREE-VARIANCES, (SEE REF(A), EQ.(16)).
C
C BY THE CALL OF COVAX, THE KIND OF COVARIANCE FUNCTION TO BE USED IS
C DETERMINED. THE VALUE OF KI(5) WILL DETERMINE THE DEGREE-VARI-
```



```
C ANCE MODEL (1,2 OR 3, CF.REF(A),EQ.(17)) THAT WILL BE USED. THE QUAN-
C TITIES K(2),K(3) MUST BE STORED IN KI(3),KI(4), AND BE EQUAL TO ZERO
C WHEN NOT USED (EG.,KI(3),KI(4) BOTH ZERO WHEN KI(5)=1). THE QUANTITY
C A(I) MUST BE STORED IN CI(8) IN UNITS OF (M/SEC)**4, AND THE SQUARE OF
C THE RATIO BETWEEN THE RADIUS OF THE BJERHAMMAR-SPHERE (RB) AND THE
C MEAN RADIUS OF THE EARTH (RE) MUST BE STORED IN CI(10).
C
C THERE ARE THEN THREE POSSIBILITIES:
C (1) ONE OF THE DEGREE-VARIANCE MODELS IS USED WITHOUT MODIFICATIONS.
C     THE SUMMATION LIMIT P OF REF.(A),EQ.(20) IS THEN FIXED TO 3.
C     BECAUSE THIS IS EQUIVALENT TO REQUIRING THE FIRST 3 DEGREE-VARIAN-
C     AREA /CMCOV/ MUST BE EQUAL TO 3 AND .TRUE., RESPECTIVELY.
C     CES TO BE ZERO, THE VARIABLES N1 AND LOCAL STORED IN THE COMMON
C (2) A NUMBER (N1) OF THE ANOMALY DEGREE-VARIANCES (DEGREE ZERO TO
C     N1-1) ARE PUT EQUAL TO EMPIRICAL DETERMINED QUANTITIES. THE ANO-
C     MALY DEGREE-VARIANCE OF DEGREE K WILL HAVE TO BE STORED IN
C     SIGMA0(IS+K+1) IN UNITS OF MGAL**2 WHEN CALLING COVAX. LOCAL MUST
C     BE EQUAL TO FALSE. COVAX WILL CONVERT THE ANOMALY DEGRE5-VARIANCES
C     INTO POTENTIAL DEGREE-VARIANCES. THE POINTER IS MUST BE POSITIVE.
C (3) THE N1 FIRST DEGREE-VARIANCES (DEGREE 0 - N1-1) ARE EQUAL TO ZERO.
C     THIS MEANS, THAT THE VALUES OF A (N1-1)-ORDER LOCAL COVARIANCE
C     FUNCTION WILL BE COMPUTED. LOCAL MUST HAVE THE VALUE .TRUE..
C IN ALL CASES N1 MUST BE LESS THAN 300.
C
C THE COVARIANCES WILL GENERALLY BE COMPUTED BY CLOSED EXPRESSIONS, BUT
C THEY MAY IN CERTAIN CASES BE USELESS IN BIG ALTITUDES OF NUMERICAL
C REASONS, CF. REF(A), SECTION 4. IN THEESE CASES MUST THE LOGICAL VARI-
C ABLE LSUM BE TRUE AND THE VARIABLE HMAX MUST HAVE ASSIGNED A VALUE
C EQUAL TO THE CRITICAL ALTITUDE. WHEN LSUM IS TRUE AND THE HEIGHT OF
C P OR Q IS GREATHER THAN HMAX, WILL THE SERIES REF(A), EQ.(16), ABBRE-
C VIATED TO DEGREE N2-1 BE USED FOR THE COMPUTATION OF THE COVARIANCES.
C THE VALUES OF LSUM, N2 AND HMAX WILL (IN THE SAME WAY AS FOR THE PARA-
C METERS SPECIFYING THE DEGREE-VARIANCE MODEL) BE TRANSFERRED TO COVAX
C THROUGH THE COMMON AREA /CMCOV/, BUT AN ARRAY SM IS TRANSFERRED AS A
C PARAMETER IN THE CALL IN ORDER TO ENABLE VARIABLE DIMENSIONING (SPECI-
C FIED BY THE VARIABLE N2 IN /CMCOV/).
C
C THE CALL OF COVAX WILL ALSO INITIALIZE CERTAIN VARIABLES USED IN
C SUBSEQUENT COMPUTATIONS.
C
C REFERENCES:
C (A) TSCHERNING,C.C.: COVARIANCE EXPRESSIONS FOR SECOND AND LOWER ORDER
C     DERIVATIVES OF THE ANOMALOUS POTENTIAL, REPORTS OF THE DEP. OF
C     GEODETIC SCIENCE NO. 225,1976.
C (B) TSCHERNING,C.C. AND R.H.RAPP: CLOSED COVARIANCE EXPRESSIONS
C     FOR GRAVITY ANOMALIES, GEOID UNDULATIONS, AND DEFLECTIONS OF
C     THE VERTICAL IMPLIED BY ANOMALY DEGREE-VARIANCE MODELS. DEP-
C     ARTMENT OF GEODETIC SCIENCE, THE OHIO STATE UNIVERSITY,
C     REPORT NO. 208, 1974.
C (C) KRARUP, T. AND C.C.TSCHERNING: EVALUATION OF ISOTROPIC COVARIANCE
C     FUNCTIONS OF TORSION BALANCE OBSERVATIONS. BULLETIN GEOD-
C     DESIQUE, VOL. 58, NO. 2, PP. 180-192, 1984.
C (D) TSCHERNING,C.C.: IMPLEMENTATION OF ALGOL-PROCEDURES FOR COV-
C     ARIANCE COMPUTATION ON THE RC 4000-COMPUTER. THE DANISH
C     GEODETIC INSTITUTE INTERNAL REPORT NO. 12, 1976.
C (H) TSCHERNING, C.C.: PREDICTION OF SPHERICAL HARMONIC
C     COEFFICIENTS USING LEAST-SQUARES COLLOCATION. SEPT. 1999.
C (I) Tscherning, C.C.: Computation of covariances of derivatives of the
C     anomalous gravity potential in a rotated reference frame.
C     Manuscripta Geodaetica, Vol. 18, no. 3, pp. 115-123, 1993.
C
      implicit none
C
      REAL*8 CI,CR,SIGMA0,SIGMA,HMAX,CV,D,D0,D1,D2,D3,D4,D5,RE,
     *RADSEC,PI,GM,A,S,RB2,T,B,SIGMAP,slop,sloq,clop,cloq,RB,RE2,SM
      INTEGER KI,N1,N2,ITCOUN,KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,
     *ND2,IIDEG,JJDEG,IS
      LOGICAL LOCAL,LSUM,LF,LT,LSPOUT,LFINI
C
```

```
      COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CV(2,2),D(36),KI(37),N1,N2,LOCAL,LSUM,LFINI
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
      COMMON /DDY/A,S,RB2,T,B,KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,ND2
      COMMON /PDEGV/SIGMAP(2200),slop,sloq,clop,cloq,IIDEG,JJDEG,LSPOUT
C THE COMMON BLOCK CONTAINS THE VALUES OF PARAMETERS USED FOR THE COM-
C PUTATIONS AND RETURN VALUES OF FUNCTIONS AND CONSTANTS, WHICH HAVE
C BEEN USED IN THE COMPUTATIONS.
C PARAMETERS USED FOR THE COMPUTATIONS:
C   CI(8) = THE CONSTANT A(I) OF REF.(A), EQ.(17) IN UNITS OF (M/SEC)**4
C   CI(10) THE SQUARE OF THE RATIO BETWEEN THE BJERHAMMAR-SPHERE RADIUS
C   (RB) AND THE MEAN RADIUS OF THE EARTH (RE), OR IF NEGATIVE RB-RE,
C   (CHANGE MADE 3 JULY 1985).
C   SIGMA0(IS+1)-SIGMA0(IS+N1) MUST CONTAIN THE EMPIRICAL ANOMALY
C   DEGREE VARIANCES IN UNITS OF MGAL**2.
C   KI(3) = K(2) OF DEG.VAR. MODEL 2 OR 3,
C   KI(4) = K(3) OF DEG.VAR. MODEL 3, CF. REF.(A), EQ.(17).
C   KI(5) = THE DEG.VAR. MODEL NUMBER, (EQUAL TO 1, 2 OR 3),
C   N1 = THE NUMBER OF EMPIRICAL DEGREE-VARIANCES USED (LOCAL =.FALSE.)
C   OR (ORDER+1) OF THE LOCAL COVARIANCE FUNCTION USED (LOCAL=.TRUE.).
C   HMAX, N2, LSUM. HMAX IS THE HEIGHT ABOVE WHICH THE LEGENDRE SERIES
C   OF MAXIMAL DEGREE N2-1 WILL BE USED FOR THE COMPUTATION OF THE CO-
C   VARIANCES WHEN LSUM IS TRUE. N2 MUST BE GREATHER THAN 2 AS WELL AS
C   GREATHER THAN N1.
C RETURN VALUES:
C   CI(10) RB-RE, A NEGATIVE VALUE (MODIFICATION 3 JULY 1985).
C   CI(9) = RB**2.
C
      DIMENSION SM(2200)
C THE ARRAY SM IS USED TO STORE THE DEGREE-VARIANCES WHEN THE LOGICAL
C VARIABLE LSUM IS TRUE. IN CASE THE SUBSCRIPT LIMIT IS CHANGED IS IT
C NECESSARY TO CHANGE THE VALUE OF THE VARIABLE N2 ACCORDINGLY.
C
      KT = KI(5)
      KT1 = KT+1
      IF (KT.GE.3) GO TO 15
      DO 16 K = KT, 2
   16 KI(K+2) = D0
   15 KI(1) = -2
      KI(2) = -1
C
      IF ((KT.LT.3).OR.(KT.EQ.3.AND.KI(4).GT.KI(3))) GO TO 17
C ASSURING, THAT KI(4).GT.KI(3), BECAUSE THIS FACT IS USED IN SUB-
C SEQUENT COMPUTATIONS.
      K = KI(3)
      KI(3) = KI(4)
      KI(4) = K
   17 II = KI(3)
      JJ = KI(4)
      SM(1) = D0
      SM(2) = D0
C     N3 = N1
      A = CI(8)
      S = CI(10)
      IF (S.GT.D0) GO TO 40
C S IS HERE RB-RE, A NEGATIVE VALUE. (MODIFICATION 3 JULY 1985).
      RB=RE+S
      RB2=RB*RB
      RE2=RE*RE
      S=RB2/RE2
      GO TO 41
   40 RB2 = S*RE2
      CI(10)=RE*( SQRT(S)-D1)
   41 CI(9) = RB2
      RB2 = RB2*1.0D-10
      T = D0
C
      SIGMA0(IS+1) = D0
      SIGMA0(IS+2) = D0
```



```
      IF (LOCAL) THEN
      SIGMA0(IS+3) = D0
      ELSE
      SIGMA0(IS+3) = SIGMA0(IS+3)*RB2/S**4
      END IF
      DO 13 K = 4, N1
      GO TO (10,11,12),KT
   10 KK = 1
      GO TO 14
   11 KK = K+II-1
      GO TO 14
   12 KK = (K+II-1)*(K+JJ-1)
   14 IF (K.LE.N1) THEN
C CONVERSION FROM MGAL**2 TO M**2/SEC**2.
      IF (.NOT.LOCAL) T = SIGMA0(IS+K)*S**(-K-1)*RB2
      SIGMA0(IS+K) = (T-A*(K-2)/((K-3)*KK))/(K-2)**2
      END IF
   13 CONTINUE
      RETURN
      END
C ---------------------------------------------------------------------
      SUBROUTINE COVBX(SM,LSAT,IS)
C ORIGINAL VERSION PROGRAMMED JULY 1975 BY C.C.TSCHERNING AS A SUB-
C ENTRY OF COVAX. NEW VERSION CREATED SEP 1987 BY CCT.
C NEW VERSION JUNE 4, 1991. LAST UPDATE 2008-10-24 BY CCT.
C
C THE CALL OF COVBX WILL FIX CERTAIN CONSTANTS USED FOR THE COMPUTA-
C TIONS, WHICH ARE INDEPENDENT OF THE POINTS P AND Q. WHEN COVBX IS CAL-
C LED, THE KIND OF QUANTITIES BETWEEN WHICH THE COVARIANCE IS TO BE
C COMPUTED MUST BE SPECIFIED.  THIS IS DONE BY STORING IN KI(6) AND
C KI(7) INTEGERS EQUAL TO THE EQUATION NUMBERS OF REF.A, EQ.(1) - (9)
C (12) AND (14), AND 10, 11, 13, 15 CORRESPONDING TO REF.(C), EQ.
C (3) - (6). HOWEVER, THE QUANTITY OF KIND 2 IS NOW THE GRAVITY
C DISTURBANCE (CHANGED FROM THE SAME QUANTITY DIVIDED BY R).
C ADDED 1999.02.12 IS (17), FOR COEFFICIENTS OF SPHERICAL HARMONICS.
C
C REFERENCES (A) - (I): SEE COVAX.
C
      implicit none
      REAL*8 CI,CR,SIGMA0,SIGMA,HMAX,CV,D,D0,D1,D2,D3,D4,D5,RE,
     *RADSEC,PI,GM,A,S,RB2,T,B,SIGMAP,slop,sloq,clop,cloq,
     *COVX,CIX,CFA,SM,SIGMAX,RE2,SNN,BB0,RKP,REM,C11
      INTEGER KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,ND2,IIDEG,JJDEG,
     *KSAT,NDX1,NDX2,NDP,NDQ,NWAR,KI,N1,N2,ITCOUN,K7,K9,K11,K13,
     *K15,K17,K19,K21,K23,K8,J2,I3,I4,M,MK,IS,I,NDT,NDTOT,NDY,KU
C
      LOGICAL LOCAL,LSUM,LN,L,LF,LT,LSAT,LNX,LTEST,LTESTS,LSPOUT,LX,
     *LSPHAR,LFINI
C
      COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CV(2,2),D(36),KI(37),N1,N2,LOCAL,LSUM,LFINI
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
      COMMON /DDX/K7(17),K9(17),K11(17),K13(17),K15(17),K17(17),
     *K19(17),K21(17),K23(17),K8(17),C11(17),J2(2),I3(2),I4(2),
     *LN(7),L(7)
      COMMON /DDY/A,S,RB2,T,B,KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,ND2
      COMMON /PDEGV/SIGMAP(2200),slop,sloq,clop,cloq,IIDEG,JJDEG,LSPOUT
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFA,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
C THE COMMON BLOCK CONTAINS THE VALUES OF PARAMETERS USED FOR THE COM-
C PUTATIONS AND RETURN VALUES OF FUNCTIONS AND CONSTANTS, WHICH HAVE
C BEEN USED IN THE COMPUTATIONS.
C PARAMETERS USED FOR THE COMPUTATIONS:
C   CI(8) = THE CONSTANT A(I) OF REF.(A), EQ.(17) IN UNITS OF (M/SEC)**4
C   CI(10) THE SQUARE OF THE RATIO BETWEEN THE BJERHAMMAR-SPHERE RADIUS
C   (RB) AND THE MEAN RADIUS OF THE EARTH (RE), OR IF NEGATIVE RB-RE,
C   (CHANGE MADE 3 JULY 1985).
C   CI(13) USER DEFINED VALUE OF CI(11). CI(14), CI(15) USER DEFINED
C   VALUES OF CI(21) - CI(24).
```

```
C   SIGMA0(IS+1)-SIGMA0(IS+N1) MUST CONTAIN THE POTENTIAL ANOMALY
C   DEGREE-VARIANCE CORRECTIONS, CF. REF.(A), EQ.16.
C   KI(3) = K(2) OF DEG.VAR. MODEL 2 OR 3,
C   KI(4) = K(3) OF DEG.VAR. MODEL 3, CF. REF.(A), EQ.(17).
C   KI(5) = THE DEG.VAR. MODEL NUMBER, (EQUAL TO 1, 2 OR 3),
C   KI(6),KI(7) THE INTEGER SPECIFYING THE KIND OF QUANTITY WHICH IS
C   ASSOCIATED WITH P, Q, RESPECTIVELY,
C   KI(26) - KI(34) USER SPECIFIED VALUES FOR KI(10) - KI(23).
C   KI(35) - KI(37) USED BY SUBROUTINE COVCG FOR STATISTICAL PURPOSES.
C   N1 = THE NUMBER OF EMPIRICAL DEGREE-VARIANCES USED (LOCAL =.FALSE.)
C   OR (ORDER+1) OF THE LOCAL COVARIANCE FUNCTION USED (LOCAL=.TRUE.).
C   HMAX, N2, LSUM. HMAX IS THE HEIGHT ABOVE WHICH THE LEGENDRE SERIES
C   OF MAXIMAL DEGREE N2-1 WILL BE USED FOR THE COMPUTATION OF THE CO-
C   GREATHER THAN N1.
C RETURN VALUES:
C   CI(1)-CI(7), THE QUANTITIES C(J,Q) OF REF.(A), EQ.(47), WITH
C   CI(1) - CI(KI(5)+1) = C(J,Q), CI(5) = C(KI(5)+2,Q),
C   CI(6) = C(KI(5)+3,Q), CI(7) = C(KI(5)+4,Q),
C   CI(11),CI(12) QUANTITIES USED TO GIVE THE COMPUTED
C   COVARIANCES THE PROPER UNITS.
C   CI(21) - CI(24) THE QUANTITIES M(1) - M(4) OF REF.(A) EQ. (26) -
C   (29). (CHANGE MADE 1986.10.20).
C   SIGMA(IS+4) - SIGMA(IS+N1), THE POTENTIAL DEGREE-VARIANCES MULTI-
C   PLIED BY THE FACTORS GIVEN IN REF.(A), TABLE 1.
C   SIGMA(IS+1) - SIGMA(IS+3), THE DEGREE-VARIANCES OF DEGREE 0,1,2
C   MINUS TERMS OF THE SAME DEGREES ACQUIRED FROM REF.(A), EQ.(34),(35),
C   (41) AND (42).
C   KI(8),KI(9) THE NUMBER OF DIFFERENTIATIONS IN RADIAL DIRECTION AND
C   WITH RESPECT TO T = COS(SPHERICAL DIST.) TO BE PERFORMED.
C   KI(10) - KI(15) THE CONSTANTS I,K,J,M,J1,M1 OF REF.(A), SECTION 2.
C   KI(16) - KI(19) THE QUANTITIES M(1) - M(4) OF REF.(A), EQ.(26)-(29).
C   KI(20),KI(21) THE EXPONENT OF THE REFERENCE GRAVITY,
C   KI(22),KI(23) THE EXPONENT OF THE RADIAL DISTANCE AND
C   KI(24),KI(25) SUBSCRIPTS OF THE RESULT STORED IN CV (COMMON CMCOV).
C
      DIMENSION SM(2200),SIGMAX(2200,5)
C     DIMENSION SM(2200),SIGMAX(400,5)
C THE ARRAY SM IS USED TO STORE THE DEGREE-VARIANCES WHEN THE LOGICAL
C VARIABLE LSUM IS TRUE. IN CASE THE SUBSCRIPT LIMIT IS CHANGED IS IT
C NECESSARY TO CHANGE THE VALUE OF THE VARIABLE N2 ACCORDINGLY.
C     EQUIVALENCE (SIGMAX(1,1),SIGMA0(401))
C SIGMAX IS USED TO HOLD DEGREE-VARIANCES OF RADIAL DERIVATIVES
C UP TO ORDER 2 IN P AND Q. (CHANGE MAY 1991).
C
C THE ARRAYS K7 - K23 CONTAINS TABLES OF QUANTITIES RELATED TO THE KIND
C OF COVARIANCES (1 - 14) WHICH MAY BE COMPUTED. THEIR ACTUAL VA-
C LUES WILL AFTER CALL OF COVBX BE STORED IN THE ELEMENTS OF THE ARRAY
C KI HAVING SUBSCRIPTS 8 - 25.
C K7 CONTAINS THE ORDER OF DIFFERENTIATION WITH RESPECT TO T,K8 THE
C ORDER OF DIFFERENTIATION WITH RESPECT TO THE RADIUS, CF.REF(A),TABLE
C 1. K9,K11,K13 THE KIND OF DIFFERENTIATIONS TO BE COMPUTED WITH RESPECT
C TO THE LATITUDE (2) AND THE LONGITUDE (3), CF.REF(A),SECTION 3. K15
C AND K17 CONTAINS AN INTEGER, WHICH WILL BE ADDED TO THE DEGREE. THE
C SUM WILL THEN BE MULTIPLIED WITH THE DEGREE-VARIANCE OF THE CORRESPON-
C DING DEGREE WHEN A FIRST AND/OR SECOND DIFFERENTIATION WITH RESPECT
C TO THE RADIAL DISTANCE HAS TAKEN PLACE.
C C11 CONTAIN QUANTITIES USED TO GIVE THE COVARIANCES THE PROPER UNITS.
C
C     LTEST=LTESTS
      LTEST=.false.
      LSPHAR=.FALSE.
      RB2 = CI(9)
      RE2=RE**2
      S=RB2/RE2
      A = CI(8)
      II=KI(3)
      JJ=KI(4)
      KT=KI(5)
      KT1=KT+1
```



```
      N3=N1
      CI(11) = D1
      KI(8)=0
      KI(9)=0
      IF (KI(6).GT.17.OR.KI(7).GT.17) GO TO 19
C
      DO 20 M = 1, 2
      K = KI(M+5)
C FOR M = 1, K IS EQUAL TO THE KIND EVALUATED IN P AND FOR M = 2 EQUAL
C TO THE KIND EVALUATED IN Q.
C
      IF (K.EQ.0.OR.K.GE.16) GO TO 42
      KI(M+9) = K9(K)
      KI(M+11) = K11(K)
      KI(M+13) = K13(K)
      CI(M+20) = K15(K)
      CI(M+22) = K17(K)
      KI(M+19) = K19(K)
      KI(M+21) = K21(K)
      KI(M+23) = K23(K)
C
      CI(11) = CI(11)*C11(K)
C     write(*,*)' k ',K,CI(21),CI(22),CI(23),CI(24)
      KI(8)=KI(8)+K7(K)
      KI(9)=KI(9)+K8(K)
      GO TO 20
C
C USER DEFINED VALUES OF KI AND CI. MAY BE USER FOR DENSITY CONTRAST
C COVARIANCES, CF. REF.(D), SECTION 3.
   42 IF (K.NE.17) THEN
      DO 43 MK=1,8
   43 KI(M+MK*2+7)=KI(MK+25)
      CI(11) = CI(11)*CI(13)
      CI(M+20) = CI(14)
      CI(M+22) = CI(15)
      KI(8)=KI(8)+KI(29)
      KI(9)=KI(9)+KI(30)
c     LSPHAR=.FALSE.
c     write(*,*)' LSPHAR=F '
      ELSE
      LSPHAR=.TRUE.
      CI(21)=D0
      CI(22)=D0
      CI(23)=D0
      CI(24)=D0
c     write(*,*)' LSPHAR=T '
c     WRITE(*,*)' N1,N3 ',N1,N3
      END IF
   20 CONTINUE
C
      KQ = K
      KP = KI(6)
   19 ND = KI(8)
c     WRITE(*,*)' COVBX: ND= ',ND
      NR = KI(9)
C
      NDP=K7(KP)+K8(KP)
      NDQ=K7(KQ)+K8(KQ)
c     WRITE(*,*)' COVBX: ND,NDP,NDQ= ',ND,NDP,NDQ
C ND AND NR ARE THE NUMBER OF DIFFERENTIATIONS WITH RESPECT TO T AND
C THE RADIAL DISTANCES, RESPECTIVELY. NDP, NDQ ARE THE TOTAL NMBER OF
C DERIVATIVES IN P, Q, REPECTIVELY.
C
      IF (LSAT.AND.(.NOT.LSPHAR)) GO TO 100
C UPDATING THE DEGREE-VARIANCES, CF. REF(A), TABLE 1.
      SIGMA(IS+1) = D0
      SIGMA(IS+2) = D0
      SIGMAP(IS+1)= D0
      SIGMAP(IS+2)= D0
```

```
      IF (LSUM) N1 = N2
      IF (N1.GE.2200) WRITE(*,*)' WARNING N1.gt.2200 '
      SNN=S**3
C     DO 21 M = 3, 600
C CHANGE 2008-10-24.
      DO 21 M = 3, 2200
      B = D1
      DO 22 I = 1, 4
   22 IF ( ABS(CI(I+20)).GT.0.0) B = B*(M+CI(I+20)-1)
      BB0=B
      IF (M.LE.N3) SIGMA(IS+M) = SIGMA0(IS+M)*B
      IF (.NOT.(LSUM.OR.LSPHAR).OR.M.EQ.3) GO TO 21
      DO 48 K = 1, KT1
   48 B = B/(M+KI(K)-1)
C STORING THE MODIFIED DEGREE-VARIANCES OF DEGREE M-1 IN SM(M) AND AD-
C DING THE DEGREE-VARIANCE CORRECTIONS FOR M .LE. N3.
      SM(M) = B*A
      SNN=SNN*S
      IF (M.LE.N3) THEN
      SM(M) = SM(M)+SIGMA(IS+M)
C     SIGMAP(M)=SM(M)*SNN
      SIGMAP(M)=SM(M)*SNN/BB0
      ELSE
C     SIGMAP(M)=B*A*SNN
      SIGMAP(M)=A*SNN*B/BB0
      END IF
C
C CF. REF(H), EQ. (4).
      SIGMAP(M)=SIGMAP(M)/(D2*M-D1)
   21 CONTINUE
      IF (N1.GT.2) THEN
      SM(3) = SIGMA(IS+3)
C     SIGMAP(3)=SM(3)*(S**3)/5.0d0
      SIGMAP(3)=SM(3)*(S**3)/(BB0*5.0d0)
      ELSE
      SIGMAP(3)=0.0D0
      END IF
C
c     write(*,*)' SIGMA0, k=1,200 '
c     write(*,249)(sigma0(k),k=1,200)
c     write(*,*)' SIGMA, k=1,200 '
c     write(*,249)(sigma(k),k=1,200)
      IF (LSPHAR.AND.LSPOUT) THEN
      LSPOUT=.FALSE.
      WRITE(*,*)' GRAVITY ANOMALY AND POTENTIAL DEG.VAR. DEG 3-200 '
      WRITE(*,249)
     *(SIGMAP(K)*(2*K-1)*(K-2)**2*1.0D10/RE2,K=3+IS,200+IS)
      WRITE(*,249)(SIGMAP(K),K=3+IS,200+IS)
  249 FORMAT(8F9.4)
      END IF
      IF (LSUM) N1 = N3
C
C EVALUATION OF THE QUANTITIES C(J,NR), CF.REF(A), TABLE 2.
      DO 23 K = 1, 7
   23 CI(K) = D0
C
      DO 25 K = 1, KT1
      CI(K) = D1
      DO 25 KQ = 1, KT1
   25 IF (K.NE.KQ) CI(K) = CI(K)/(KI(KQ)-KI(K))
C CF.,EQ.(19). WE WILL THEN COMPUTE THE QUANTITIES GIVEN IN REF(A)
C REF(A), TABLE 2.
      IF (NR.LT.2) GO TO 29
      RKP = CI(21)+CI(22)+CI(23)+CI(24)
      IF (NR.EQ.4) REM = CI(21)*(CI(22)+CI(23)+CI(24))+CI(22)
     *                  *(CI(23)+CI(24))+CI(23)*CI(24)
C
      GO TO (26,27,28),KT
   26 CI(NR+3) = D1
```



```
      IF (NR.GT.2) CI(NR+2) = RKP+3
      IF (NR.EQ.4) CI(NR+1) = REM+3*RKP+7
      GO TO 29
   27 IF (NR.GT.2) CI(NR+2) = D1
      IF (NR.EQ.4) CI(NR+1) =-KI(3)+3+RKP
      GO TO 29
   28 IF (NR.EQ.4) CI(NR+1) = D1
   29 IF (NR.EQ.0) GO TO 31
C
      DO 30 KP = 1, 4
      DO 30 K  = 1, KT1
   30 IF ( ABS(CI(KP+20)).NE.0.0) CI(K) = CI(K)*(CI(KP+20)-KI(K))
C
C THE LOGICAL ARRAYS L AND LN REGISTER WHICH TERMS THAT WILL HAVE TO
C BE EVALUATED , RESPECTIVELY NOT EVALUATED IN REF.(A), EQ. (47).
   31 DO 38 K = 1, 7
      L(K) =  ABS(CI(K)).GT.1.0E-15
   38 LN(K) = .NOT.(L(K))
C
      DO 32  K = 3, 7
      DO 32  M = 1, 3
      IF (M.EQ.1.AND.K.GT.5.OR.(M+KI(K)-1).EQ.0.AND.K.LT.5.OR.LN(K))
     *GO TO 32
      GO TO (34,34,35,35,34,36,37),K
   34 B = D1
      GO TO 33
   35 B = D1/(M+KI(K) -1)
      GO TO 33
   36 B = (M-1)
      GO TO 33
   37 B = (M-1)*(M-1)
   33 SIGMA(IS+M) = SIGMA(IS+M)-A*CI(K)*B
   32 CONTINUE
      SIGMA(IS+3) = SIGMA(IS+3)-A*CI(2)
      IF (LTEST) write(*,2)(SIGMA(I),I=1,6)
    2 FORMAT(6E12.6,I3)
C
      ND1 = ND+1
      ND2 = ND1+1
      RETURN
C
  100 DO 109 M=1,7
      DO 109 NDT=1,5
  109 LNX(M,NDT)=LT
      NDTOT=NDP+NDQ+1
      ND=NDTOT-1
      ND1=ND+1
      ND2=ND1+1
C
      DO 101 NDT=1,NDTOT
      DO 110 M=1,4
  110 CI(M+20)=D0
      M=1
      IF (NDT.GT.1) THEN
       CI(21)=D1
       M=2
      END IF
      IF (NDT.GT.2) THEN
      IF (NDP.EQ.1.AND.NDQ.EQ.1.AND.NDTOT.EQ.3) THEN
      CI(22)=D1
      ELSE
      CI(22)=D2
      END IF
      M=3
      END IF
      IF (NDT.GT.3) THEN
      CI(23)=D1
      M=M+1
      IF (NDT.EQ.5)THEN
```

```
      CI(24)=D2
      M=M+1
      END IF
      END IF
      NR=M-1
      NDY=NDTOT-M
      IF (LTEST)WRITE(6,*)NDT,CI(21),CI(22),CI(23),CI(24)
C UPDATING THE DEGREE-VARIANCES, CF. REF(A), TABLE 1.
      SIGMAX(1,NDT) = D0
      SIGMAX(2,NDT) = D0
      DO 121 M = 3, N1
      B = D1
      DO 122 I = 2, NDT
  122 B = B*(M+CI(I+19)-1)
C noget galt her.
  121 IF (M.LE.N3) SIGMAX(M,NDT) = SIGMA0(IS+M)*B
C
C EVALUATION OF THE QUANTITIES C(J,NR), CF.REF(A), TABLE 2.
      DO 123 K = 1, 7
  123 CI(K) = D0
C
      DO 125 K = 1, KT1
      CI(K) = D1
      DO 125 KU = 1, KT1
  125 IF (K.NE.KU) CI(K) = CI(K)/(KI(KU)-KI(K))
C CF.,EQ.(19). WE WILL THEN COMPUTE THE QUANTITIES GIVEN IN REF(A)
C REF(A), TABLE 2.
      IF (NR.LT.2) GO TO 129
      RKP = CI(21)+CI(22)+CI(23)+CI(24)
      IF (NR.EQ.4) REM = CI(21)*(CI(22)+CI(23)+CI(24))+CI(22)
     *                 *(CI(23)+CI(24))+CI(23)*CI(24)
C
      GO TO (126,127,128),KT
  126 CI(NR+3) = D1
      IF (NR.GT.2) CI(NR+2) = RKP+3
      IF (NR.EQ.4) CI(NR+1) = REM+3*RKP+7
      GO TO 129
  127 IF (NR.GT.2) CI(NR+2) = D1
      IF (NR.EQ.4) CI(NR+1) =-KI(3)+3+RKP
      GO TO 129
  128 IF (NR.EQ.4) CI(NR+1) = D1
  129 IF (NR.EQ.0) GO TO 131
C
      DO 130 KU = 1, 4
      DO 130 K  = 1, KT1
  130 IF ( ABS(CI(KU+20)).NE.0.0) CI(K) = CI(K)*(CI(KU+20)-KI(K))
  131 DO 106 K=1,7
  106 CIX(K,NDT)=CI(K)
C
C THE LOGICAL ARRAYS L AND LN REGISTER WHICH TERMS THAT WILL HAVE TO
C BE EVALUATED , RESPECTIVELY NOT EVALUATED IN REF.(A), EQ. (47).
      DO 138 K = 1, 7
      IF (NDT.EQ.1) L(K)=LF
      LNX(K,NDT)= ABS(CI(K)).LE.1.0D-10
      L(K) =  ABS(CI(K)).GT.1.0E-10.OR.L(K)
  138 LN(K)=.NOT.(L(K))
      IF (LTEST) WRITE(6,*)'NDT,LN',NDT,(LNX(K,NDT),K=1,7)
C
      DO 132  K = 3, 7
      DO 132  M = 1, 3
      IF (M.EQ.1.AND.K.GT.5.OR.(M+KI(K)-1).EQ.0.AND.K.LT.5.OR.
     *LNX(K,NDT)) GO TO 132
      GO TO (134,134,135,135,134,136,137),K
  134 B = D1
      GO TO 133
  135 B = D1/(M+KI(K) -1)
      GO TO 133
  136 B = (M-1)
      GO TO 133
```



```
  137 B = (M-1)*(M-1)
  133 SIGMAX(M,NDT) = SIGMAX(M,NDT)-A*CI(K)*B
  132 CONTINUE
      SIGMAX(3,NDT) = SIGMAX(3,NDT)-A*CI(2)
      IF (LTEST) write(*,2)(SIGMAX(I,NDT),I=1,6),NDT
C
      NDX1(NDT) = NDY+1
      NDX2(NDT) = NDY+2
  101 CONTINUE
      RETURN
      END
C ---------------------------------------------------------------------
      SUBROUTINE COVCX(SM,COV,IS,LSAT)
C ORIGINALLY PROGRAMMED JULY 1975 BY C.C.TSCHERNING AS A SUB-
C ENTRY TO COVAX. SEPARATE SUBROUTINE CREATED SEPT 1987 BY CCT.
C LATEST MODIFICATION JUL 25, 2003 BY CCT.
C
C COMPUTATION OF THE COVARIANCE IN A SPECIFIC PAIR OF POINTS, OR
C BETWEEN A FUNCTIONAL ASSOCIATED WITH A POINT AND A SPHERICAL-HARMONIC
C COEFFICIENT. THE  VALUE IS RETURNED THROUGH THE PARAMETER COV.
C THE COVARIANCES COMPUTED WILL BE IN UNITS CORRESPONDING TO THE KIND
C OF QUANTITIES, I.E. FOR KIND (1) METERS, (2) EOTVOS (E), (3) MGAL,
C (4),(5) E, (6),(7) ARCSECONDS, (8) - (14) E, (17) UNITLESS.
C THE FOLLOWING QUANTITIES MUST BE STORED IN THE ELEMENTS OF THE ARRAY
C CR WHEN COVCX IS CALLED: (1) COSINE TO THE SPHERICAL DISTANCE BET-
C WEEN P AND Q, (2),(3) THE HEIGHT OF P, Q RESPECTIVELY, (4),(5) SINE
C OF THE LATITUDE THE  OF P, Q, RESPECTIVELY, (6),(7) COSINE OF THE
C LATITUDE OF P, Q, RESPECTIVELY, (8),(9) SINE AND COSINE OF THE
C LONGITUDE DIFFERENCE. THE REFERENCE GRAVITY WILL HAVE TO BE STORED
C IN CR(10),CR(11) FOR P, Q RESPECTIVELY (WHEN USED, OTHERWISE STORE
C 1.0). FOR KIND 17, COS AND SIN OF LONGITUDES MUST BE STORED IN THE
C COMMON BLOCK /PDEGV/.
C
C THE CALL OF COVCX WILL RESULT IN THE COMPUTATION OF THE COVARIANCE ,
C WHICH IS TRANSFERRED TO THE CALLING PROGRAM THROUGH THE VARIABLE COV.
C THE RESULT WILL ALSO BE TRANSFERRED IN THE COMMON CMCOV, BY THE ARRAY
C CV(2,2). IN CASE IT IS POSSIBLE TO COMPUTE MORE THAN ONE QUANTITY AT
C A TIME (I.E. WHEN DERIVATIVES WITH RESPECT TO T=COS(SPHERICAL DIST-
C TANCE) ARE COMPUTED, KINDS 6 - 11, 13 AND 15), THE COVARIANCE
C OF TYPE 6, 8, 10 AND 23 WILL BE STORED IN THE ELEMENT WITH SUBSCRIPT
C 2 AND OTHERWISE IN THE ELEMENT WITH SUBSCRIPT 1. THE KIND OF THE
C FUNCTIONALS IN P WILL DETERMINE THE VALUE OF THE FIRST SUBSCRIPT
C WHILE THE KIND OF THE FUNCTIONALS IN Q WILL DETERMINE THE SECOND
C SUBSCRIPT. EXAMPLE: KIND 6 IN P AND KIND 1 IN Q WILL DELIVER
C THE COVARIANCE BETWEEN THE PRIME-VERTICAL VERTICAL DEFLECTION AND
C AND THE HEIGHT ANOMALY IN CV(1,1), BETWEEN THE MERIDIAN VERTICAL
C DEFLECTIAN AND THE HEIGHT ANOMALY IN CV(2,1).
C
C WHEN LSAT IS TRUE, THE 4D ARRAY COVX HOLDS THE VECTORS OR MATRICES
C OF COVARIANCES BETWEEN ALL 0, 1 OR 2 DERIVATIVES.
C
      implicit none
      REAL*8 CI,CR,SIGMA0,SIGMA,HMAX,CV,D
      INTEGER KI,N1,N2,ITCOUN,IIMAX,IMAX1,I21,I,ILAST,JMAX1,J,JKK,IKK,
     *NCASE,KPQ,IDIF,KKC,KKD,M,K1,K2,I1,I2,NDTOT,NDT,IS,J1,M1,IJ,KM,IX,
     *IIX,IIY,JX,IX1,JX1,K6,M6 ,KZ
C    *,id,jd
      REAL*8 D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,C11,CN,CY,R2PQ,
     *A,S,RB2,T,B,HP,HQ,SP,SQ,CP,CQ,SD,CD,RP,RQ,RE2,CLAT,SLAT,CLON,SLON,
     *RH,GAMM,COV,CJLO,SJLO,WWC,GGC,WWS,GGS,COVC,WW,DDDC,DDDS,
     *GG,DDD,SJL1,SC,CS,SCC,CC,CCS,COVS,CSC,CPSD,CQSD,CPCD,CQCD,SS,C,
     *S2,ST,T2,P2,P3,CX,GI,GJ,SI,SM,DC,SIGMAX,DCN,RL,RL2,R,RL1,RN,
     *RNL,P,RL3,RL5,S3,RL4,RL7,S4,S5,RL6,U,RM,V,Q,G,SS1,SS2,RP2,
     *RQ2,RPQ,FAK5,RP2Q,CNX,DD,RPQ2,D3132,D313,CN23,CN33,D37,D27,
     *CF,CZ,C11P,C11Q
      INTEGER KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,ND2,
     *KSAT,NDX1,NDX2,NDP,NDQ,NWAR,IIOLD,JOLD
     *,NFU,KEYH,NINTH,NTABH,NHE,NSTART,IIDEG,JJDEG
      REAL*8 COVX,CIX,CFA,RRC,HTA,TMAX,SIZEI,SIGMAP,slop,sloq,
```

```
     *clop,cloq,root0,SUMIJ,CCCIJ,
     *sq2,ys,yc,vv,v1,gs,gc,dds,ddc,pii,pim0,pim1,pim2,dlp,
     *dlp0,dlp1,dlp2,dap,dap0,
     *dap1,dap2,ddap,ddap0,ddap1,ddap2,ddal0,ddal1,vi
      INTEGER K7,K9,K11,K13,K15,K17,K19,K21X,K23,K8,J2,I3,I4

      parameter (iimax=20000)
C
      LOGICAL LOCAL,LSUM,LSUMC,LOLDP,LOLDQ,LN,L,LF,LT,LTEST,LTABH
     *,LTA,LSAT,LTESTS,LDGP,LDGQ,LSPHAR,LSPOUT,LTSPH,LCOS
     *,LSPHP,LSPHQ,LX,LNX,LFINI
c
      COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
     *CV(2,2),D(36),KI(37),N1,N2,LOCAL,LSUM,LFINI
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
      COMMON /DDX/K7(17),K9(17),K11(17),K13(17),K15(17),K17(17),
     *K19(17),K21X(17),K23(17),K8(17),C11(17),J2(2),I3(2),I4(2),
     *LN(7),L(7)
C     COMMON /DDX/K7(15),K9(15),K11(15),K13(15),K15(15),K17(15),K19(15),
C    *K21(15),K23(15),K8(15),C11(15),J2(2),I3(2),I4(2),LN(7),L(7)
      COMMON /DDY/A,S,RB2,T,B,KT,KT1,K,II,JJ,N3,KK,KQ,KP,ND,NR,ND1,ND2
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFA,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
      COMMON /CTABH/RRC(1200),HTA(5),TMAX,SIZEI,NFU(5),KEYH(5,5),
     *NINTH,NTABH(15,5,5),NHE,NSTART,LTABH
      COMMON /PDEGV/SIGMAP(2200),slop,sloq,clop,cloq,IIDEG,JJDEG,LSPOUT
      common  /rrsph/root0(iimax)
      common /con3/SUMIJ(32761),CCCIJ(32761),
     *sq2,ys,yc,vv,v1,gs(3),gc(3),dds(3,3),
     *ddc(3,3),iiold,jold,LSPHAR,LTSPH
      common /sphold/pii,pim0,pim1,pim2,dlp,dlp0,dlp1,dlp2,dap,dap0,
     *dap1,dap2,ddap,ddap0,ddap1,ddap2,ddal0,ddal1,vi
C THE COMMON BLOCK CONTAINS THE VALUES OF PARAMETERS USED FOR THE COM-
C PUTATIONS AND RETURN VALUES OF FUNCTIONS AND CONSTANTS, WHICH HAVE
C BEEN USED IN THE COMPUTATIONS.
C PARAMETERS USED FOR THE COMPUTATIONS:
C   CI(8) = THE CONSTANT A(I) OF REF.(A), EQ.(17) IN UNITS OF (M/SEC)**4
C   CI(10) THE SQUARE OF THE RATIO BETWEEN THE BJERHAMMAR-SPHERE RADIUS
C   (RB) AND THE MEAN RADIUS OF THE EARTH (RE), OR IF NEGATIVE RB-RE,
C   (CHANGE MADE 3 JULY 1985).
C   CI(13) USER DEFINED VALUE OF CI(11). CI(14), CI(15) USER DEFINED
C   VALUES OF CI(21) - CI(24).
C   NEW VARIABLES ADDED MAY 1, 1986 AND NOV 1986:
C   CI(16) - CI(24), WHERE CI(20)=0.0 IF PRECISE FORMULAE FOR DERIVATIVES
C   MAY BE USED. IN THIS CASE IS CI(16)=SIN(LONGITUDE DIFFERENCE/2)**2,
C   CI(17)=SIN(LATITUDE DIFFERENCE/2), CI(18)=COS(LATITUDE DIFFERENCE),
C   CI(19)=COS(LATITUDE DIFFERENCE/2). OTHERWISE CI(20)=1.0.
C   CR(2),CR(3) THE HEIGHT OF P, Q, RESPECTIVELY, (UNITS METERS),
C   CR(4),CR(5) SINE OF THE LATITUDE OF P, Q, RESPECTIVELY,
C   CR(6),CR(7) COSINE OF THE LATITUDE OF P, Q, RESPECTIVELY,
C   CR(8),CR(9) SINE AND COSINE OF THE LONGITUDE DIFFERENCE,
C   CR(10),CR(11) THE REFERENCE GRAVITY IN P, Q, RESPECTIVELY (WHEN
C   USED, OTHERWISE STORE 1.0E0), (UNITS M/SEC**2).
C   KI(3) = K(2) OF DEG.VAR. MODEL 2 OR 3,
C   KI(4) = K(3) OF DEG.VAR. MODEL 3, CF. REF.(A), EQ.(17).
C   KI(5) = THE DEG.VAR. MODEL NUMBER, (EQUAL TO 1, 2 OR 3),
C   KI(6),KI(7) THE INTEGER SPECIFYING THE KIND OF QUANTITY WHICH IS
C   ASSOCIATED WITH P, Q, RESPECTIVELY,
C   KI(26) - KI(34) USER SPECIFIED VALUES FOR KI(10) - KI(23).
C   KI(35) - KI(37) USED BY SUBROUTINE COVCG FOR STATISTICAL PURPOSES.
C   N1 = THE NUMBER OF EMPIRICAL DEGREE-VARIANCES USED (LOCAL =.FALSE.)
C   OR (ORDER+1) OF THE LOCAL COVARIANCE FUNCTION USED (LOCAL=.TRUE.).
C   HMAX, N2, LSUM. HMAX IS THE HEIGHT ABOVE WHICH THE LEGENDRE SERIES
C   OF MAXIMAL DEGREE N2-1 WILL BE USED FOR THE COMPUTATION OF THE CO-
C   VARIANCES WHEN LSUM IS TRUE. N2 MUST BE GREATHER THAN 2 AS WELL AS
C   GREATHER THAN N1.
C RETURN VALUES:
C   CR(ND*8+12), THE VALUES OF THE ND'TH DERIVATIVE OF THE SUM OF THE
C   FINITE LEGENDRE-SERIES, CF.REF.(A), EQ.(20),(48) AND (52).
```



```
C   CR(ND*8+13) - CR(ND*8+19), THE VALUES OF THE ND'TH DERIVATIVES OF
C   THE FUNCTIONS F(-2), F(-1), F(KI(3)), F(KI(4)), S0, S1, S2, CF. REF.
C   (A), EQ. (42), (41), (39), (39), (30), (34) AND (35).
C   SIGMA0(IS+1) - SIGMA0(IS+N1) THE POTENTIAL DEGREE-VARIANCE
C   CORRECTIONS, CF. REF.(A), EQ.(16), (AFTER THE CALL OF COVAX).
C   SIGMA(IS+4) - SIGMA(IS+N1), THE POTENTIAL DEGREE-VARIANCES MULTI-
C   PLIED BY THE FACTORS GIVEN IN REF.(A), TABLE 1.
C   SIGMA(IS+1) - SIGMA(IS+3), THE DEGREE-VARIANCES OF DEGREE 0,1,2
C   MINUS TERMS OF THE SAME DEGREES ACQUIRED FROM REF.(A), EQ.(34),(35),
C   (41) AND (42).
C   KI(8),KI(9) THE NUMBER OF DIFFERENTIATIONS IN RADIAL DIRECTION AND
C   WITH RESPECT TO T = COS(SPHERICAL DIST.) TO BE PERFORMED.
C   KI(10) - KI(15) THE CONSTANTS I,K,J,M,J1,M1 OF REF.(A), SECTION 2.
C   KI(16) - KI(19) THE QUANTITIES M(1) - M(4) OF REF.(A), EQ.(26)-(29).
C   KI(20),KI(21) THE EXPONENT OF THE REFERENCE GRAVITY,
C   KI(22),KI(23) THE EXPONENT OF THE RADIAL DISTANCE AND
C   KI(24),KI(25) SUBSCRIPTS OF THE RESULT STORED IN CV (COMMON CMCOV).
C
C ARRAYS CN, DCN, SIGMAX, DD ADDED MAY 1991.
C
C REFERENCES (A)-(I) SEE COVAX.
C
      DIMENSION SM(2200),CX(6,8),DC(6),SIGMAX(400,5),CN(8,5),DCN(8,5),
     *C(6),V(6),U(6),G(6),P(6),R(6),SS1(4),CZ(5),RM(6),Q(6),DD(6,6)
C THE ARRAY SM IS USED TO STORE THE DEGREE-VARIANCES WHEN THE LOGICAL
C VARIABLE LSUM IS TRUE. IN CASE THE SUBSCRIPT LIMIT IS CHANGED IS IT
C NECESSARY TO CHANGE THE VALUE OF THE VARIABLE N2 ACCORDINGLY.
      DIMENSION GG(3),DDD(3,3),GGC(3),GGS(3),DDDC(3,3),DDDS(3,3)
C
      EQUIVALENCE (CX(1,1),C(1)),(CX(1,2),V(1)),(CX(1,3),U(1)),
     *(CX(1,4),G(1)),(CX(1,5),P(1)),(CX(1,6),R(1)),(CX(1,7),SS1(1)),
     *(CX(2,8),SS2),(SIGMAX(1,1),SIGMA0(401)),(D(1),DD(1,1))
C K7 CONTAINS THE ORDER OF DIFFERENTIATION WITH RESPECT TO T,K8 THE
C ORDER OF DIFFERENTIATION WITH RESPECT TO THE RADIUS, CF.REF(A),TABLE
C 1. K9,K11,K13 THE KIND OF DIFFERENTIATIONS TO BE COMPUTED WITH RESPECT
C TO THE LATITUDE (2) AND THE LONGITUDE (3), CF.REF(A),SECTION 3. K15
C AND K17 CONTAINS AN INTEGER, WHICH WILL BE ADDED TO THE DEGREE. THE
C SUM WILL THEN BE MULTIPLIED WITH THE DEGREE-VARIANCE OF THE CORRESPON-
C DING DEGREE WHEN A FIRST AND/OR SECOND DIFFERENTIATION WITH RESPECT
C TO THE RADIAL DISTANCE HAS TAKEN PLACE.
C C11 CONTAIN QUANTITIES USED TO GIVE THE COVARIANCES THE PROPER UNITS.
C
      LTEST=LSPHAR.AND.ITCOUN.LT.5.AND.LTESTS
      IF (LTEST) THEN
      KI(35)=KI(35)+1
      ITCOUN=ITCOUN+1
      END IF
      LSPHP=LF
      LSPHQ=LF
      T = CR(1)
      HP = CR(2)
      HQ = CR(3)
      SP = CR(4)
      SQ = CR(5)
      CP = CR(6)
      CQ = CR(7)
      SD = CR(8)
      CD = CR(9)
      RP = RE+HP
      RQ = RE+HQ
      RE2= RE**2
C
      KP=KI(6)
      KQ=KI(7)
C
C CHANGE 2003-03-22.
      LDGP=KP.EQ.3
      LDGQ=KQ.EQ.3
      IF (KP.EQ.17.OR.KQ.EQ.17) THEN
```

```
C     write(*,*)'10385 CX, KP,KQ,LSAT= ',KP,KQ,LSAT
C
       IF (KP.NE.17.AND.KQ.EQ.17) THEN
        LSPHQ=LT
        KPQ=KP
c     write(*,*)' LSAT ',LSAT,' KPQ=KP= ',KPQ
        CLAT=CP
        SLAT=SP
        SLON=SLOP
        CLON=CLOP
        RH=RE+HP
        GAMM=CR(11)
       END IF
C
       IF (KQ.NE.17.AND.KP.EQ.17) THEN
        LSPHP=LT
        KPQ=KQ
c       write(*,*)' LSAT ',LSAT,' KPQ=KP= ',KPQ
        CLAT=CQ
        SLAT=SQ
        SLON=SLOQ
        CLON=CLOQ
        RH=RE+HQ
        GAMM=CR(10)
       END IF
C
       sq2=sqrt(d2)
       imax1=iideg+1
       i21=2*(iideg+1)
       if (iimax.lt.i21) write(*,*) ' imax too large '
C
       DO 501 I=1, i21
        ROOT0(I)= SQRT(DFLOAT(I-1))
  501  CONTINUE
C
       ILAST=(iMAX1)**2+1
C
       IF (KP.EQ.17.AND.KQ.EQ.17) THEN
C COV IS THE VARIANCE OF THE (I,J)'TH COEFFICIENT.
        COV=SIGMAP(IIDEG+1)
        if (LTSPH.AND.(IIDEG.GT.7.AND.IIDEG.LT.13))
     *  write(*,*)' IIDEG+1, COV= ',IIDEG+1,COV
       ELSE
C
C SETTING ORDER OF DIFFERENTIATION.
C      IF  (KPQ.GE.1.OR.KPQ.LE.5) THEN   **  ERRONEOUS  **
C ERROR DETECTED 2000-03-27 BY CCT.
        IF  (KPQ.GE.1.AND.KPQ.LE.5) THEN
        IF (LSAT) THEN
         idif=1
         if (KPQ.eq.5) idif=2
        else
         idif=0
        end if
       ELSE
c     IF  (KPQ.GE.6.OR.KPQ.LE.11) THEN   ** ERRONEOUS  **
C ERROR DETECTED 2000-03-27 BY CCT.
       IF  (KPQ.GE.6.AND.KPQ.LE.9) THEN
        idif=1
       else
        idif=2
       end if
      end if
c
      CFA=D1
      iiold=-1
      jold=-1
      pim0=D1
      cjlo=d1
```



```
      sjlo=d0
      jmax1=abs(jjdeg)+1
      LCOS=jjdeg.ge.0
c
C SEE REF(H) EQ. (6).
      do 1002, j=1, jmax1
      do 1001, i=j, imax1
      call spharm(slat,clat,sjlo,cjlo,rh,i-1,j-1,idif,.true.)
c
      if (i.gt.8.and.i.lt.13.and.j.eq.1.AND.LTSPH.and.i.eq.imax1)
     *then
       write(*,*)'i, sigmap, yc, KPQ ',i, sigmap(i),yc,KPQ
      end if
c     write(*,*)' ddc/s11 ',ddc(1,1),dds(1,1)
c
      wwc=sigmap(i)*yc
C UNITS OF M.
      if (idif.gt.0) then
      ggc(1)=sigmap(i)*gc(1)/RE
      ggc(2)=sigmap(i)*gc(2)/RE
      ggc(3)=sigmap(i)*gc(3)/RE
      IF (.not.LSAT) THEN
      ggc(1)=-ggc(1)*RADSEC/GAMM
      ggc(2)=-ggc(2)*RADSEC/GAMM
C UNITS OF ARCSEC.
C GRAVITY DISTURBANCE:
      ggc(3)=-ggc(3)*1.0D5
C UNITS OF MGAL USED.
C GRAVITY ANOMALY MISSING !!
      end if
       if (idif.gt.1) then
        do 990,kkc=1,3
        do 990,kkd=1,3
  990    dddc(kkc,kkd)=sigmap(i)*ddc(kkc,kkd)*1.0D9/RE2
C EU USED.
       end if
c
      end if
c
      wws=sigmap(i)*ys
      if (idif.gt.0) then
      ggs(1)=sigmap(i)*gs(1)/RE
      ggs(2)=sigmap(i)*gs(2)/RE
      ggs(3)=sigmap(i)*gs(3)/RE
      IF (.NOT.LSAT) THEN
C GRAVITY ANOMALY MISSING !! (SEE BELOW AT LABEL 1013).
      ggs(1)=-ggs(1)*RADSEC/GAMM
      ggs(2)=-ggs(2)*RADSEC/GAMM
C UNITS OF ARCSEC.
      ggs(3)=ggs(3)*1.0D5
C UNITS OF MGAL USED.
      end if
      if (idif.gt.1) then
      do 991,kkc=1,3
      do 991,kkd=1,3
  991 ddds(kkc,kkd)=sigmap(i)*dds(kkc,kkd)*1.0D9/RE2
c     write(*,*)' dddc/s11 ',dddc(1,1),ddds(1,1)
C EU USED.
       end if
c
      end if
c
      IF (.NOT.LSAT) THEN
       GO TO (1011,1012,1013,1014,1015,1016,1017,1018,1019,1020
     * ,1021,1024,1023,1022,1025),KPQ
C HEIGHT ANOMALY (M).
 1011  covc=wwc/GAMM
       covs=wws/GAMM
       GO TO 1126
```

```
C GRAVITY DISTURBANCE (MGAL).
 1012  covc= wwc*i/rh*1.0d5
       covs= wws*i/rh*1.0d5
       GO TO 1126
C GRAVITY ANOMALY (MGAL).
 1013  covc= wwc*(i-2)/rh*1.0d5
       covs= wws*(i-2)/rh*1.0d5
C
       if (i.gt.8.and.i.lt.13.and.j.eq.jmax1.AND.LTSPH
     * .and.i.eq.imax1) then
        write(*,1092) i,j,covc,ww,yc
 1092   format(' i,j,covc,ww,yc= ',2i3,3d14.6)
        LTSPH=LF
       end if
C
      GO TO 1126
C RADIAL DER. OF GRAVITY ANOMALY (EU).
 1014  covc=wwc*(i-2)*(i+1)/(rh*rh)*1.0d9
       covs=wws*(i-2)*(i+1)/(rh*rh)*1.0d9
       GO TO 1126
C VERTICAL GRAVITY GRADIENT.
 1015  covc=wwc*i*(i+1)/(rh*rh)*1.0d9
       covs=wws*i*(i+1)/(rh*rh)*1.0d9
C
       if (i.gt.8.and.i.lt.13.and.j.eq.jmax1.AND.LTSPH
     * .and.i.eq.imax1) then
        write(*,1091) i,i*(i+1),j,ww,covc
 1091   FORMAT(' i,i*(i+1),j,ww,covc= ',2i3,i4,2d14.6)
        LTSPH=LF
       end if
C
       GO TO 1126
C DEFLECTION, MERIDIAN COMP.
 1016  covc= ggc(2)
       covs= ggs(2)
       IF (LTEST) THEN
        write(*,*)' ggc,ggs1 ', covc,covs
        ITCOUN=ITCOUN+1
       END IF
       GO TO 1126
C DEFLECTION, PRIME VERTICAL COMP.
 1017  covc= ggc(1)
       covs= ggs(1)
       IF (LTEST) THEN
        write(*,*)' ggc,ggs1 ', covc,covs
        ITCOUN=ITCOUN+1
       END IF
       GO TO 1126
C PRIME VERTICAL DER. OF GRAVITY ANOMALY
C ERROR HERE !!!!!
 1019  covc=-ggc(2)*i/rh
       covs=-ggs(2)*i/rh
       GO TO 1126
C MERIDIAN DER. OF GRAVITY ANOMALY
 1018  covc=-ggc(1)*i/rh
       covs=-ggs(1)*i/rh
       GO TO 1126
C MERIDIAN DER. OF GRAVITY DISTURBANCE. CORR. 2000-03-27 BY CCT.
 1020  covc=-dddc(3,2)
       covs=-ddds(3,2)
       GO TO 1126
C PRIME VERTICAL DER. OF GRAVITY DISTURBANCE.
 1021  covc=-dddc(3,1)
       covs=-ddds(3,1)
       GO TO 1126
C 2. ORDER PRIME VERTICAL DER.
 1022  covc=dddc(1,1)
       covs=ddds(1,1)
       GO TO 1126
```



```
C MIXED PRIME VERTICAL & MERIDIAN DER. * 2. (TORSION BALANCE).
 1023  covc= dddc(1,2)*d2
       covs= ddds(1,2)*d2
       GO TO 1126
C 2. ORDER MERIDIAN COMP.
 1024  covc=dddc(2,2)
       covs=ddds(2,2)
       GO TO 1126
C DIFFERENCE 2. ORDER HORIZONTAL DER. (TORSION BALANCE).
 1025  covc=(dddc(1,1)-dddc(2,2))
       covs=(ddds(1,1)-ddds(2,2))
C
 1126  CONTINUE
       IF (LCOS) THEN
        covx(1,1,1,1)=covc
        cov=covc
       ELSE
        covx(1,1,1,1)=covs
        cov=covs
       END IF
C
      ELSE
       IF (LCOS) THEN
        ww=wwc
        DO 1875,IKK=1,3
         gg(IKK)=ggc(IKK)*1.0D5
         DO 1876,JKK=1,3
          ddd(IKK,JKK)=dddc(IKK,JKK)
 1876    CONTINUE
 1875   CONTINUE
       ELSE
        ww=wws
        DO 1877,IKK=1,3
         gg(IKK)=ggs(IKK)*1.0D5
         DO 1878,JKK=1,3
          ddd(IKK,JKK)=ddds(IKK,JKK)
 1878    CONTINUE
 1877   CONTINUE
       END IF
C
       NCASE=NDP+1+NDQ*3
C      write(*,*)NDP,NDQ,NCASE
       GO TO (1801,1802,1803,1804,1810,1810,1807,
     * 1810,1810),NCASE
C NO DERIVATIVES IN P OR Q.
1801   COVX(1,1,1,1)=ww/GAMM
       GO TO 1810
C 1 DERIVATIVE IN P, NONE IN Q.
1802   COVX(1,1,1,1)=gg(1)
       COVX(2,1,1,1)=gg(2)
       COVX(3,1,1,1)=gg(3)
C GRAVITY ANOMALY WITH GEOID. ADDED 1992.09.07.
       IF (LDGP) COVX(3,1,1,1)=gg(3)-1.0D5*D2*ww/RP
C ERROR DETECTED 2003-07-24.
C      IF (LDGP) COVX(3,1,1,1)=gg(3)-D2*ww/RP
       GO TO 1810
C 2 DERIVATIVES IN P, NONE IN Q.
1803  COVX(1,1,1,1)= ddd(1,1)
      COVX(2,1,1,1)= ddd(2,1)
      COVX(1,2,1,1)=COVX(2,1,1,1)
      COVX(3,1,1,1)= ddd(3,1)
      COVX(1,3,1,1)=COVX(3,1,1,1)
      COVX(2,2,1,1)=ddd(2,2)
      COVX(2,3,1,1)= ddd(2,3)
      COVX(3,2,1,1)=COVX(2,3,1,1)
      COVX(3,3,1,1)= ddd(3,3)
      GO TO 1810
C NO DERIVATIVE IN P, 1 IN Q.
 1804 COVX(1,1,1,1)=gg(1)
```

```
      COVX(1,1,2,1)=gg(2)
      COVX(1,1,3,1)=gg(3)
C GRAVITY ANOMALY WITH GEOID. ADDED 1999.09.07, CORR 000.04.28.
      IF (LDGQ) COVX(1,1,3,1)=gg(3)-1.0D5*D2*ww/RQ
C ERROR DETECTED 2003-07-24.
      GO TO 1810
C NO DERIVATIVE IN P, TWO IN Q.
 1807 COVX(1,1,1,1)= ddd(1,1)
      COVX(1,1,2,1)= ddd(2,1)
      COVX(1,1,1,2)=COVX(1,1,2,1)
      COVX(1,1,3,1)= ddd(3,1)
      COVX(1,1,1,3)=COVX(1,1,3,1)
      COVX(1,1,2,2)= ddd(2,2)
      COVX(1,1,3,2)= ddd(3,2)
      COVX(1,1,2,3)=COVX(1,1,3,2)
      COVX(1,1,3,3)= ddd(3,3)
c
 1810 CONTINUE
      COV=COVX(KSAT(KP,1),KSAT(KP,2),KSAT(KQ,1),KSAT(KQ,2))
      IF (LTEST.AND.ITCOUN.LT.200) THEN
       WRITE(*,*)' KSAT ',KSAT(KP,1),KSAT(KP,2),KSAT(KQ,1),
     * KSAT(KQ,2),' COV ',COV,'NDP,NDQ ',NDP,NDQ,' IDIF ',IDIF
      END IF
      ITCOUN=ITCOUN+1
C
C this permits test of LSAT prediction of coefficients for
c LGRID = F. 2000-04-17.
      IF (j.eq.1) THEN
      IF (idif.eq.0) then
      covc=wwc
      else
      if (idif.eq.1) then
      covc=ggc(KSAT(KQ,1))*1.0D5
      else
      covc=dddc(KSAT(KQ,1),KSAT(KQ,2))
      end if
      end if
c
      else
      IF (idif.eq.0) then
      covc=wwc
      covs=wws
      else
      if (idif.eq.1) then
      covc=ggc(KSAT(KQ,1))*1.0D5
      covs=ggs(KSAT(KQ,1))*1.0D5
      else
      covc=dddc(KSAT(KQ,1),KSAT(KQ,2))
      covs=ddds(KSAT(KQ,1),KSAT(KQ,2))
      end if
      end if
      END IF
c
      end if
C
      IF (j.eq.1) THEN
      CCCIJ((i-1)**2+1)=COVC
      ELSE
      CCCIJ((i-1)**2+2*(j-1)+1)=COVS
c     ELSE
      CCCIJ((i-1)**2+2*(j-1))=COVC
      END IF
 1001 if (sigmap(i).gt.1.0d-10.and.LTEST)
     *write(*,511)i,j,covc,covc/sigmap(i),clat,cjlo,kp,kq,idif
  511 format(' i,j,covc,LLY,clat,cjlo,kp,kq= ',2i3,2d15.5,2f6.3,3i3)
C for each order of differentiation a different storage !
C
      sjl1=sjlo
      sjlo=sjlo*clon+cjlo*slon
```



```
      cjlo=cjlo*clon-sjl1*slon
 1002 continue
c
      end if
C
C CHANGE HERE TO TAKE CARE OF KSI, ETA 2000-05-02
      GO TO (2011,2011,2011,2011,2011,2016,2016,2018,2018,2020
     *,2020,2022,2024,2022,2024),KPQ
C EV, DERIVATIVES  Z, dgZ, ZZ.
 2011 IF (LCOS) THEN
      cv(1,1)=covc
      ELSE
      cv(1,1)=covs
      END IF
      GO TO 2026
C
C KSI, ETA.
 2016 IF (LCOS) THEN
      cv(1,1)=ggc(1)
      IF (LSPHP) THEN
      cv(1,2)=ggc(2)
      ELSE
      cv(2,1)=ggc(2)
      END IF
      ELSE
      cv(1,1)=ggs(1)
      IF (LSPHP) THEN
      cv(1,2)=ggs(2)
      ELSE
      cv(2,1)=ggs(2)
      END IF
      END IF
      GO TO 2026
C
C DELTAG, X, Y.
 2018 IF (LCOS) THEN
      cv(1,1)=-ggc(1)*i/rh
      IF (LSPHP) THEN
      cv(1,2)=-ggc(2)*i/rh
      ELSE
      cv(2,1)=-ggc(2)*i/rh
      END IF
      ELSE
      cv(1,1)=-ggs(1)*i/rh
      IF (LSPHP) THEN
      cv(1,2)=-ggs(2)*i/rh
      ELSE
      cv(2,1)=-ggs(2)*i/rh
      END IF
      END IF
      GO TO 2026
C
C XZ AND YZ.
 2020 IF (LCOS) THEN
      cv(1,1)=-dddc(3,1)
      IF (LSPHP) THEN
      cv(1,2)=-dddc(3,2)
      ELSE
      cv(2,1)=-dddc(3,2)
      END IF
      ELSE
      cv(1,1)=-ddds(3,1)
      IF (LSPHP) THEN
      cv(1,2)=-ddds(3,2)
      ELSE
      cv(2,1)=-ddds(3,2)
      END IF
      END IF
      GO TO 2026
```

```
C
C XX AND YY
 2022 IF (LCOS) THEN
      cv(1,1)=dddc(1,1)
      IF (LSPHP) THEN
      cv(1,2)=dddc(2,2)
      ELSE
      cv(2,1)=dddc(2,2)
      END IF
      ELSE
      cv(1,1)=ddds(1,1)
      IF (LSPHP) THEN
      cv(1,2)=ddds(2,2)
      ELSE
      cv(2,1)=ddds(2,2)
      END IF
      END IF
      GO TO 2026
C
C 2*XY and YY-XX.
 2024 IF (LCOS) THEN
      cv(1,1)=dddc(2,1)*d2
      IF (LSPHP) THEN
      cv(1,2)=dddc(2,2)-dddc(1,1)
      ELSE
      cv(2,1)=dddc(2,2)-dddc(1,1)
      END IF
      ELSE
      cv(1,1)=ddds(2,1)*d2
      IF (LSPHP) THEN
      cv(1,2)=ddds(2,2)-ddds(1,1)
      ELSE
      cv(2,1)=ddds(2,2)-ddds(1,1)
      END IF
      END IF
C
 2026 RETURN
      END IF
C
C IN HEIGH ALTITUDES AND WHEN LSUM IS TRUE WILL THE COVARIANCE BE COM-
C PUTED BY A SUMMATION OF THE LEGENDRE-SERIES ABBREVIATED TO DEGREE
C N2-1.
      LSUMC = LSUM .AND. (HP.GT.HMAX .OR. HQ.GT.HMAX)
C COMPUTATION OF THE CONSTANT USED TO CONVERT THE COVARIANCE INTO
C PROPER UNITS.
      CI(12) = CI(11)/(RP**KI(22)*RQ**KI(23)
     **CR(11)**KI(21)*CR(10)**KI(20))
C
      S = RB2/(RP*RQ)
C     IF(CI(10).LT.D0) S=D1-(RE*(HP+HQ+D2*(RE-CI(10)))+HP*HQ
C    *- (RE-CI(10))**2)/(RP*RQ)
      LOLDP = (KI(6).EQ.12) .OR. (KI(6).EQ.14) .OR. LSAT
      LOLDQ = (KI(7).EQ.12) .OR. (KI(7).EQ.14) .OR. LSAT
      IF (LSUMC) N1 = N3
C
C COMPUTATION OF THE QUANTITIES D(1)-D(36),CF.REF(A),SECTION 3.
C (MODIFIED ACCORDING TO REF.(C)).
C     if (.true.)write(*,*)' covcx nd=',ND
      DO I=1,36
       D(I)=D0
      END DO
      IF (ND.EQ.0) GO TO 55
C
      D(1) = D1
      CS = CP*SQ
      SC = SP*CQ
      SCC = SC*CD
      CC = CP*CQ
      CCS = CC*SD
```



```
      CSC = CS*CD
      IF (CI(20).GT.0.5) GO TO 201
C CF. REF.(D), EQ. (7) AND (8).
C ERROR 2002-10-06. CHANGE OF SIGN ON CI(17)*CI(19).
      D(2)= D2*(CI(17)*CI(19)+SP*CQ*CI(16))
      D(7)= D2*(-CI(17)*CI(19)+SQ*CP*CI(16))
      IF (ABS(D(2)-CS+SCC).GT.1.0D-6 .OR.
     * ABS(D(7)-SC+CSC).GT.1.0D-6) THEN
       WRITE(*,*)
     *' WARNING D(2) ',D(2),(CS-SCC)
       WRITE(*,*)
     *' WARNING D(7) ',D(7),(SC-CSC)
       write(*,*)ci(16),ci(17),ci(19)
      END IF
      GO TO 202
 201  D(2) = CS-SCC
      D(7) = SC-CSC
 202  CPSD = CP*SD
      CPCD = CP*CD
      CQSD = CQ*SD
      CQCD = CQ*CD
      D(3) = CQSD
      D(13)=-CPSD
C
      IF (ND.EQ.1) GO TO 55
      SS = SP*SQ
      D(8) = CC+SS*CD
C CF. REF.(D). EQ.(9).
      IF(CI(20).LT.0.5) THEN
       D(8)=CI(18)-D2*SP*SQ*CI(16)
       IF (ABS(D(8)-(CC+SS*CD)).GT.1.0D-6) THEN
        WRITE(*,*)' D(8) ',D(8),(cc+SS*CD)
        D(8)=-D(8)
       END IF
      END IF
      D(9) = -SQ*SD
      D(14)= SP*SD
      D(15)= CD
      IF (LOLDP) GO TO 91
      D(4) = D(2)+D(3)
      D(6) = D(3)-D(2)
      GO TO 92
   91 D(4) = -T
      D(6) = -CQCD/CP
   92 IF (LOLDQ) GO TO 93
      D(19)= D(13)+D(7)
      D(31)= D(13)-D(7)
      GO TO 94
   93 D(19)= -T
      D(31)= -CPCD/CQ
C
   94 IF (ND.EQ.2) GO TO 55
      IF (LOLDP) GO TO 95
      D(10) = D(9)+D(8)
      D(12) = D(9)-D(8)
      D(16) = D(15)+D(14)
      D(18) = D(15)-D(14)
      GO TO 96
   95 D(10) = -D(7)
      D(12) = SQ*CD/CP
      D(16) = CPSD
      D(18) = SD/CP
   96 IF (LOLDQ) GO TO 97
      D(20) = D(14)+D(8)
      D(32) = D(14)-D(8)
      D(21) = D(15)+D(9)
      D(33) = D(15)-D(9)
      GO TO 98
   97 D(20) = -D(2)
```

```
      D(21) = -CQSD
      D(32) = SP*CD/CQ
      D(33) = -SD/CQ
C
   98 IF (ND.EQ.3) GO TO 55
      IF (.NOT.(LOLDP.AND.LOLDQ)) GO TO 99
      D(22) = T
      D(24) = CQCD/CP
      D(34) = CPCD/CQ
      D(36) = CD/CC
      GO TO 55
   99 IF (.NOT.LOLDQ) GO TO 100
      D(22) = D(21)+D(20)
      D(24) = D(21)-D(20)
      D(34) = D(33)+D(32)
      D(36) = D(33)-D(32)
      GO TO 55
  100 D(22) = D(16)+D(10)
      D(34) = D(16)-D(10)
      D(24) = D(18)+D(12)
      D(36) = D(18)-D(12)
   55 CONTINUE
      if (.false.) write(*,1555)(D1-T),CI(20),CI(17),CR(8)
 1555 format(' t1,ci20,17,cr8',4d14.5)
C
      IF (LTABH) CALL TABH(C,LTA)
      IF (LTA.AND.LTABH) GO TO 204
      S2 = S*S
      ST = S*T
      T2 = T*T
      P2 = (D3*T2-D1)/D2
      P3 = (D3*ST+D1)/D2
C
C INITIALIZING ARRAY ELEMENTS. NOTE THE USE OF THE EQUIVALENCING.
      DO 50  K = 1, 8
      DO 50 M = 1, ND2
   50 CX(M,K) = D0
      DO 51 K = 1, ND2
      C(K) = D0
   51 DC(K) = D0
      DO 52 K = 1, 40
   52 CR(K+11) = D0
      Q(1)=D0
      RM(1)=D0
C
      IF (.NOT.LSAT) THEN
C
C SUMMATION AND DIFFERENTIATION OF THE LEGENDRE SERIES, CF.REF(A),EQ.
C (49) AND (51).
      IF (LSUMC) N1 = N2
      K1 = N1
      K2 = N1+1
      K = N1-1
      DO 54 M = 1, N1
      GI = (D2*K+D1)*S/K1
      GJ = -K1*S2/K2
      K2 = K1
      K1 = K
      K = K-1
      IF (.NOT.LSUMC) SI = SIGMA(IS+K2)
      IF (LSUMC) SI = SM(K2)
      I2 = 0
      I1 = 1
      DO 53 I = 2, ND2
      B = DC(I)
      DC(I) = C(I)
      C(I) = GI*(DC(I)*T+I2*DC(I1))+GJ*B+SI
      SI = D0
      I2 = I1
```



```
   53 I1 = I
   54 CONTINUE
      IF (LSUMC) N1 = N3
C
C     IF (LSUMC) GO TO 75
      ELSE
       KP=KI(6)
       KQ=KI(7)
       LDGP=KP.EQ.3
       LDGQ=KQ.EQ.3
C
C INITIALIZING ARRAY ELEMENTS. NOTE THE USE OF THE EQUIVALENCING.
       DO 150  K = 1, 8
       DO 150 M = 1, 6
  150  CX(M,K) = D0
       NDTOT=NDP+NDQ+1
       DO 151 K = 1, 8
       DO 151 NDT=1,NDTOT
        CN(K,NDT) = D0
  151  DCN(K,NDT) = D0
C
C SUMMATION AND DIFFERENTIATION OF THE LEGENDRE SERIES, CF.REF(A),EQ.
C (49) AND (51).
       K1 = N1
       K2 = N1+1
       K = N1-1
       DO 154 M = 1, N1
        GI = (D2*K+D1)*S/K1
        GJ = -K1*S2/K2
        K2 = K1
        K1 = K
        K = K-1
        DO 154 NDT=1,5
         SI = SIGMAX(K2,NDT)
         I2 = 0
         I1 = 1
         DO 153 I = 2, NDX2(NDT)
          B = DCN(I,NDT)
          DCN(I,NDT) = CN(I,NDT)
          CN(I,NDT) = GI*(DCN(I,NDT)*T+I2*DCN(I1,NDT))+GJ*B+SI
          SI = D0
          I2 = I1
  153    I1 = I
  154   CONTINUE
C
       END IF
       IF (LSUMC) GO TO 75
C COMPUTATION OF THE FUNCTIONS L=R(1), N=1/RN, M=RM(2), F0=P(2), CF.
C REF.(A), EQ. (31)-(33),(40) AND (77A).
       RL2 = D1-D2*ST+S2
       RL =  SQRT(RL2)
       R(1) = RL
       RL1 = D1/RL
       RN = D1/(D1+RL-ST)
       RL2 = D1/RL2
       RNL = RN*RL1
       RM(2) = D1-RL-ST
       P(2) = S*DLOG(D2*RN)
       RL3 = RL2*RL1
       RL5 = RL3*RL2
       S3 = S2*S
       R(2) = -S*RL1
       IF (ND.EQ.0) GO TO 56
C
C COMPUTATION OF THE DERIVATIVES WITH RESPECT TO T.
C CF. REF.(A), EQ. (77B),(69A),(57).
       R(3) = -S2*RL3
       RM(3) = -R(2)-S
       P(3) = S2*(RNL+RN)
```

```
      IF (ND.EQ.1) GO TO 56
C
C CF. REF.(A), EQ. (77C),(69B),(58).
      R(4) = -D3*S3*RL5
      RM(4) = -R(3)
      P(4) = S3*(RL3+(D1+(D2+RL1)*RL1)*RN)*RN
      IF (ND.EQ.2) GO TO 56
C
C CF. REF.(A), EQ. (77D),(69C),(59).
      RL4 = RL2*RL2
      RL7 = RL5*RL2
      S4 = S2*S2
      R(5) = -15.0E0*S4*RL7
      RM(5) = -R(4)
      P(5) = S4*(D3*RL5+((D3+D3*RL1)*RL3+D2*(D1+(D3+(D3+RL1)*RL1)*RL1)
     * *RN)*RN)*RN
      IF (ND.EQ.3) GO TO 56
C
C CF. REF.(A), EQ. (69D),(60).
      S5 = S4*S
      RL6 = RL4*RL2
      RM(6) = -R(5)
      P(6) = D3*S5*((D5*RL7+((D4+D5*RL1)*RL5+((D4+(8.0E0
     * +D4*RL1)*RL1)*RL3+(D2+(8.0E0+(12.0E0+(8.0E0+D2*RL1)*RL1)
     * *RL1)*RL1)*RN)*RN)*RN)*RN)
C
   56 IF (LN(2)) GO TO 58
C COMPUTATION OF THE FUNCTION F-1 AND ITS DERIVATIVES, CF. REF.(A),
C EQ. (41) AND (61) - (65).
      U(2) = S*(RM(2)+T*P(2))
      IF (ND2.LT.3) GO TO 58
      DO 57 K = 3, ND2
   57 U(K) = S*(RM(K)+T*P(K)+(K-2)*P(K-1))
C
   58 IF (LN(1)) GO TO 60
C COMPUTATION OF THE FUNCTION F-2 AND ITS DERIVATIVES, CF. REF.(A) EQ.
C (42), AND (65)- (68).
      DO 59 K = 2, ND2
       GO TO (61,61,62,63,64,65),K
   61  CY = S*(D1-T2)/4.0E0
       GO TO 59
   62  CY = -ST/D2
       GO TO 59
   63  CY = D3*P(2)-S/D2
       GO TO 59
   64  CY = 9.0E0*P(3)
       GO TO 59
   65  CY = 18.0E0*P(4)
   59 V(K) = S*(RM(K)*P3+S*((K-2)*D3*RM(K-1)/D2+P2*P(K)+D3*T*P(K-1)*
     * (K-2)+CY))
C
   60 IF (LN(3)) GO TO 73
C COMPUTATION OF THE FUNCTION F1 AND ITS DERIVATIVES, CF. REF.(A) EQ.
C (36), REF.(B), EQ.(101) AND REF.(A), EQ.(70),(71).
      Q(2) = DLOG(D1+D2*S/(D1-S+RL))
      IF (ND.EQ.0) GO TO 66
      Q(3) = S2*RNL
      IF (ND.EQ.1) GO TO 66
      Q(4) = S3*((RL1+D1)*RN+RL2)*RNL
      IF (ND.EQ.2) GO TO 66
      Q(5) = S4*(D3*RL4+((D2+D3*RL1)*RL2+(D2 +(D4+D2*RL1)*RL1)*RN)
     * *RN)*RNL
      IF (ND.EQ.3) GO TO 66
      Q(6) = D3*S5*(D5*RL6+((D3+D5*RL1)*RL4+((D2+(6.0E0+D4*RL1)
     * *RL1)*RL2+(D2+(6.0E0+(6.0E0+D2*RL1)*RL1)*RL1)*RN)*RN)*RN)*RNL
C
C COMPUTATION OF THE FUNCTION F2 AND ITS DERIVATIVES, CF. REF.(A), EQ.
C (3),(72)-(75).
   66 P(2) = (RL-D1+T*Q(2))/S
```



```
      IF (ND.EQ.0) GO TO 68
      DO 67  K = 3, ND2
   67 P(K) = (R(K-1)+T*Q(K)+(K-2)*Q(K-1))/S
   68 I1 = II-1
      K1 = 1
      J1 = I1
      IF (I1.GE.2) GO TO 149
      DO 49 M = 2, ND2
       IF (I1.EQ.0) G(M) = Q(M)
       IF (I1.EQ.1) G(M) = P(M)
   49 CONTINUE
  149 IF (L(4)) J1 = JJ-1
      IF (J1.LE.1) GO TO 73
C
C CF. REF.(A), EQ. (38),(76).
      DO 71  K = 2, J1
       DO 69  M = 2, ND2
        B = Q(M)
        Q(M) = P(M)
   69  P(M) = (R(M-1)+(2*K-1)*((M-2)*Q(M-1)+T*Q(M))-K1/S*B)/(K*S)
       IF (K.NE.I1) GO TO 71
       DO 70 M = 2, ND2
   70  G(M) = P(M)
   71 K1 = K
C
   73 IF (LN(6)) GO TO 72
C CF. REF.(A), EQ. (34),(55).
      SS1(2) = S2*(T-S)*RL3
      IF (ND.GT.0) SS1(3) = S2*(RL3+D3*(T-S)*S*RL5)
C
C CF. REF.(A), EQ. (35).
   72 IF (L(7)) SS2= S2*((T+S)*RL3+D3*S*(T2-D1)*RL5)
C
   75 IF (.NOT.LSAT) THEN
C ADDING THE DIFFERENT TERMS, CF. REF.(A), EQ. (22),(47).
C TIPLIED BY RB**2 IN UNITS OF MGAL**2, THE INTEGERS K(2),K(3) OF EQ.
      DO 79 M = 2, ND2
C CF. REF.(A), EQ. (50),(52).
      C(M) = S*C(M)
      if (ltest)write(*,*)' CM',C(M),M
      CR(M*8 -4) = C(M)
      DO 78 K = 1, 7
      IF (LN(K)) GO TO 78
C STORING THE TERMS FOR TRANSFER TO THE CALLING PROGRAM USING THE COMMON
C AREA /CMCOV/.
      CR(M*8+K -4) = A*CX(M,K+1)*CI(K)
      IF (K.EQ.5) CR(M*8+K-4) = -CR(M*8+K-4)
      C(M) = C(M)+CR(M*8+K -4)
      if (ltest)write(*,1)CX(M,K+1),CI(K),C(M),k
    1 format(' CX,CI,C,K,NDT ',3E15.7,3I2)
   78 CONTINUE
   79 CR(M+50)=C(M)
C
      ELSE
C
C FOR THIS SECTION SEE REF.(I) FOR ALL EQUATIONS.
       RP2=RP*RP
       RQ2=RQ*RQ
       RPQ=RQ*RP
       DO 178 NDT=1,5
       DO 178 M = 2, NDX2(NDT)
        CN(M,NDT)=CN(M,NDT)*S
        if (ltest)write(*,*)' CM',CN(M,NDT),M,NDT,S,LSAT
        DO 179 K = 1, 7
         IF (LNX(K,NDT))GO TO 179
         FAK5=D1
         IF (K.EQ.5) FAK5=-D1
         CN(M,NDT)=CN(M,NDT)+A*CX(M,K+1)*CIX(K,NDT)*FAK5
         if (ltest)write(*,1)CX(M,K+1),CIX(K,NDT),CN(M,NDT),k,NDT
```

```
  179   CONTINUE
        CN(M-1,NDT)=CN(M,NDT)*(-1)**(NDT+1)
  178  CONTINUE
       IF (LTEST)WRITE(*,*)' NDP,NDQ,KP,KQ=',NDP,NDQ,KP,KQ
C
C WE NOW CALCULATE THE CROSS-COVARIANCES BETWEEN ALL QUANTI-
C TIES OF THE GIVEN ORDERS.
       NCASE=NDP+1+NDQ*3
       GO TO (801,802,803,804,805,806,807,808,809),NCASE
C NO DERIVATIVES IN P OR Q.
 801  COVX(1,1,1,1)=CN(1,1)
       GO TO 810
C 1 DERIVATIVE IN P, NONE IN Q. REF(I), EQ. (16) AND (17).
 802  COVX(1,1,1,1)=D(3)*CN(2,1)/RP
      COVX(2,1,1,1)=D(2)*CN(2,1)/RP
      COVX(3,1,1,1)=CN(1,2)/RP
C GRAVITY ANOMALY WITH GEOID. ADDED 1992.09.07.
      IF (LDGP) COVX(3,1,1,1)=(-CN(1,2)-D2*CN(1,1))/RP
      GO TO 810
C 2 DERIVATIVES IN P, NONE IN Q. REF(I), EQ. (24)-(28).
 803  COVX(1,1,1,1)=(D(3)*D(3)*CN(3,1)+CN(1,2)-T*CN(2,1))/RP2
      COVX(2,1,1,1)=D(2)*D(3)*CN(3,1)/RP2
      COVX(1,2,1,1)=COVX(2,1,1,1)
      COVX(3,1,1,1)=D(3)*(CN(2,2)-CN(2,1))/RP2
      COVX(1,3,1,1)=COVX(3,1,1,1)
      COVX(2,2,1,1)=(D(2)*D(2)*CN(3,1)-T*CN(2,1)+CN(1,2))/RP2
      COVX(2,3,1,1)=(D(2)*(CN(2,2)-CN(2,1)))/RP2
      COVX(3,2,1,1)=COVX(2,3,1,1)
      COVX(3,3,1,1)=CN(1,3)/RP2
      GO TO 810
C NO DERIVATIVE IN P, 1 IN Q. REF(I), EQ. (18), (19).
  804 COVX(1,1,1,1)=D(13)*CN(2,1)/RQ
      COVX(1,1,2,1)=D(7)*CN(2,1)/RQ
      COVX(1,1,3,1)=CN(1,2)/RQ
C GRAVITY ANOMALY WITH GEOID. ADDED 1992.09.07.
      IF (LDGQ) COVX(3,1,1,1)=(-CN(1,2)-D2*CN(1,1))/RQ
      GO TO 810
C 1 DERIVATIVE IN BOTH P AND Q. REF(I), EQ. (20)-(23).
  805 COVX(1,1,1,1)=(D(3)*D(13)*CN(3,1)+D(15)*CN(2,1))/RPQ
      COVX(2,1,1,1)=(D(2)*D(13)*CN(3,1)+D(14)*CN(2,1))/RPQ
      COVX(3,1,1,1)=D(13)*CN(2,2)/RPQ
      COVX(1,1,2,1)=(D(3)*D(7)*CN(3,1)+D(9)*CN(2,1))/RPQ
      COVX(2,1,2,1)=(D(2)*D(7)*CN(3,1)+D(8)*CN(2,1))/RPQ
      COVX(3,1,2,1)=D(7)*CN(2,2)/RPQ
      COVX(1,1,3,1)=D(3)*CN(2,2)/RPQ
      COVX(2,1,3,1)=D(2)*CN(2,2)/RPQ
      COVX(3,1,3,1)=CN(1,3)/RPQ
C GRAVITY ANOMALY WITH GRAVITY VECTOR AND GRAVITY. ADDED 1992.09.30.
      IF (LDGP.AND.(.NOT.LDGQ)) THEN
      COVX(3,1,1,1)=D(13)*(-CN(2,2)-D2*CN(2,1))/RPQ
      COVX(3,1,2,1)=D(7)*(-CN(2,2)-D2*CN(2,1))/RPQ
      COVX(3,1,3,1)=(-CN(1,3)-D2*CN(1,2))/RPQ
      END IF
      IF ((.NOT.LDGP.AND.LDGQ)) THEN
      COVX(1,1,3,1)=D(3)*(-CN(2,2)-D2*CN(2,1))/RPQ
      COVX(2,1,3,1)=D(2)*(-CN(2,2)-D2*CN(2,1))/RPQ
      COVX(3,1,3,1)=(-CN(1,3)-D2*CN(1,2))/RPQ
      END IF
      IF (LDGP.AND.LDGQ)
     *COVX(3,1,3,1)=(CN(1,3)+D4*(CN(1,2)+CN(1,1)))/RPQ
      GO TO 810
C 2 DERIVATIVES IN P, ONE IN Q. REF(I), EQ. (29)-(33).
  806 RP2Q=RP2*RQ
      CNX=CN(2,2)-T*CN(3,1)+D(3)*D(3)*CN(4,1)-CN(2,1)
      COVX(1,1,1,1)=(D(13)*CNX+D2*DD(3,3)*D(3)*CN(3,1))/RP2Q
      COVX(1,1,2,1)=(D(7)*CNX+D2*DD(3,2)*D(3)*CN(3,1))/RP2Q
      COVX(1,1,3,1)=(CN(1,3)+CN(1,2)+D(3)*D(3)*CN(3,2)-T*CN(2,2))/RP2Q
c     COVX(2,1,1,1)=(D(2)*D(3)*D(13)*CN(4,1)+D(17)*CN(2,1)
c    *     +(D(2)*D(15)+D(3)*D(14)+D(13)*D(7))*CN(3,1))/RP2Q
```



```
      COVX(2,1,1,1)=(D(2)*D(3)*D(13)*CN(4,1)
     *     +(D(2)*D(15)+D(3)*D(14))*CN(3,1))/RP2Q
C POSSIBLE ERROR 2002-10-29
      COVX(2,1,2,1)=(DD(2,2)*DD(3,1)*CN(3,1)+DD(2,1)*(DD(3,2)*CN(3,1)
     *     +DD(1,2)*DD(3,1)*CN(4,1)))/RP2Q
      COVX(2,1,3,1)=D(2)*D(3)*CN(3,2)/RP2Q
      COVX(3,1,1,1)=(DD(1,3)*DD(3,1)*(CN(3,2)-CN(3,1))+DD(3,3)
     *     *(CN(2,2)-CN(2,1)))/RP2Q
      COVX(3,1,2,1)=(DD(1,2)*DD(3,1)*(CN(3,2)-CN(3,1))+DD(3,2)
     *     *(CN(2,2)-CN(1,2)))/RP2Q
      COVX(3,1,3,1)=DD(3,1)*CN(2,3)/RP2Q
c     COVX(3,1,3,1)=DD(1,3)*CN(2,3)/RP2Q
      COVX(1,2,1,1)=COVX(2,1,1,1)
      COVX(1,2,2,1)=COVX(2,1,2,1)
      COVX(1,2,3,1)=COVX(2,1,3,1)
      CNX=CN(2,2)-T*CN(3,1)+D(2)*D(2)*CN(4,1)-CN(2,1)
      COVX(2,2,1,1)=(DD(1,3)*CNX+D2*D(2)*DD(2,3)*CN(3,1))/RP2Q
      COVX(2,2,2,1)=(DD(1,2)*CNX+D2*D(2)*DD(2,2)*CN(3,1))/RP2Q
      COVX(2,2,3,1)=(CN(1,3)+CN(1,2)+D(2)*D(2)*CN(3,2)-T*CN(2,2))/RP2Q
      CNX=DD(2,1)*(CN(3,2)-CN(3,1))
      COVX(2,3,1,1)=(DD(1,3)*CNX+DD(2,3)*(CN(2,2)-CN(2,1)))/RP2Q
      COVX(2,3,2,1)=(DD(1,2)*CNX+DD(2,2)*(CN(2,2)-CN(2,1)))/RP2Q
      COVX(2,3,3,1)=DD(2,1)*CN(2,3)/RP2Q
      COVX(1,3,1,1)=COVX(3,1,1,1)
      COVX(1,3,2,1)=COVX(3,1,2,1)
      COVX(1,3,3,1)=COVX(3,1,3,1)
      COVX(3,2,1,1)=COVX(2,3,1,1)
      COVX(3,2,2,1)=COVX(2,3,2,1)
      COVX(3,2,3,1)=COVX(2,3,3,1)
      COVX(3,3,1,1)=DD(1,3)*CN(2,3)/RP2Q
      COVX(3,3,2,1)=DD(1,2)*CN(2,3)/RP2Q
      COVX(3,3,3,1)=CN(1,4)/RP2Q
C GRAVITY ANOMALY ADDED 1992.09.30.
      IF (LDGQ) THEN
      COVX(1,1,3,1)=-(CN(1,3)+D3*CN(1,2)+D(3)*D(3)*(CN(3,2)+D2*CN(3,1))
     *-T*(CN(2,2)+D2*CN(2,1)))/RP2Q
      COVX(2,1,3,1)=-D(2)*D(3)*(CN(3,2)+D2*CN(3,1))/RP2Q
      COVX(3,1,3,1)=-DD(3,1)*(CN(2,3)+D2*CN(2,2))/RP2Q
C     COVX(3,1,3,1)=-DD(1,3)*(CN(2,3)+D2*CN(2,2))/RP2Q
      IF (LTEST) write(*,*)' COVX(3,1,3,1)= ',COVX(3,1,3,1)
      COVX(1,2,3,1)=COVX(2,1,3,1)
      COVX(2,2,3,1)=-(CN(1,3)+D3*CN(1,2)+D(2)*D(2)*(CN(3,2)+D2*CN(3,1))
     *-T*(CN(2,2)+D2*CN(2,1)))/RP2Q
      COVX(2,3,3,1)=-DD(2,1)*(CN(2,3)+D2*CN(2,2))/RP2Q
      COVX(1,3,3,1)=COVX(3,1,3,1)
      COVX(3,2,3,1)=COVX(2,3,3,1)
      COVX(3,3,3,1)=-(CN(1,4)+D2*CN(1,3))/RP2Q
      END IF
      GO TO 810
C NO DERIVATIVE IN P, TWO IN Q. REF(I), EQ. (24)-(28).
  807 COVX(1,1,1,1)=(CN(1,2)+D(13)*D(13)*CN(3,1)-T*CN(2,1))/RQ2
      COVX(1,1,2,1)=D(13)*D(7)*CN(3,1)/RQ2
      COVX(1,1,1,2)=COVX(1,1,2,1)
      COVX(1,1,3,1)=(D(13)*(CN(2,2)-CN(2,1)))/RQ2
C ERROR 2002-11-26.
C     COVX(1,1,3,1)=(D(3)*(CN(2,2)-CN(2,1)))/RQ2
      COVX(1,1,1,3)=COVX(1,1,3,1)
      COVX(1,1,2,2)=(CN(1,2)+D(7)*D(7)*CN(3,1)-T*CN(2,1))/RQ2
      COVX(1,1,3,2)=(D(7)*(CN(2,2)-CN(2,1)))/RQ2
      COVX(1,1,2,3)=COVX(1,1,3,2)
      COVX(1,1,3,3)=CN(1,3)/RQ2
      GO TO 810
C ONE DERIVATIVE IN P, TWO IN Q.  REF(I), EQ. (29)-(33).
  808 RPQ2=RP*RQ2
      CNX=CN(2,2)-T*CN(3,1)+D(13)*D(13)*CN(4,1)-CN(2,1)
      COVX(1,1,1,1)=(D(3)*CNX+D2*DD(3,3)*D(13)*CN(3,1))/RPQ2
      COVX(2,1,1,1)=(D(2)*CNX+D2*DD(2,3)*D(13)*CN(3,1))/RPQ2
      COVX(3,1,1,1)=(CN(1,3)+CN(1,2)+D(13)*D(13)*CN(3,2)
     *    -T*CN(2,2))/RPQ2
```

```
C ERROR CORRECTED 1992.09.04 BY CCT.
      COVX(1,1,2,1)=(D(7)*D(13)*D(3)*CN(4,1)+(D(7)*DD(3,3)+D(13)
     *    *DD(3,2))*CN(3,1))/RPQ2
C     COVX(2,1,2,1)=(DD(2,2)*DD(1,3)*CN(3,1)+DD(1,2)*(DD(3,2)*CN(3,1)
C CHANGE 2002-11-01.
      COVX(2,1,2,1)=(DD(2,2)*DD(1,3)*CN(3,1)+DD(1,2)*(DD(2,3)*CN(3,1)
     *    +DD(2,1)*DD(1,3)*CN(4,1)))/RPQ2
      COVX(3,1,2,1)=DD(1,2)*DD(1,3)*CN(3,2)/RPQ2
      COVX(1,1,3,1)=(DD(3,1)*DD(1,3)*(CN(3,2)-CN(3,1))+DD(3,3)
     *    *(CN(2,2)-CN(2,1)))/RPQ2
      COVX(2,1,3,1)=(DD(2,1)*DD(1,3)*(CN(3,2)-CN(3,1))+DD(2,3)
     *    *(CN(2,2)-CN(1,2)))/RPQ2
      COVX(3,1,3,1)=DD(1,3)*CN(2,3)/RPQ2
      COVX(1,1,1,2)=COVX(1,1,2,1)
      COVX(2,1,1,2)=COVX(2,1,2,1)
      COVX(3,1,1,2)=COVX(3,1,2,1)
      CNX=CN(2,2)-T*CN(3,1)+D(7)**2*CN(4,1)-CN(2,1)
      COVX(1,1,2,2)=(D(3)*CNX+D2*D(7)*DD(3,2)*CN(3,1))/RPQ2
      COVX(2,1,2,2)=(D(2)*CNX+D2*D(7)*DD(2,2)*CN(3,1))/RPQ2
      COVX(3,1,2,2)=(CN(1,3)+CN(1,2)+D(7)**2*CN(3,2)
     *    -T*CN(2,2))/RPQ2
      CNX=D(7)*(CN(3,2)-CN(3,1))
      COVX(1,1,3,2)=(D(3)*CNX+DD(3,2)*(CN(2,2)-CN(2,1)))/RPQ2
      COVX(2,1,3,2)=(D(2)*CNX+DD(2,2)*(CN(2,2)-CN(2,1)))/RPQ2
C POSSIBLE ERROR 1992.09.08.
      COVX(3,1,3,2)=D(7)*CN(2,3)/RPQ2
      COVX(1,1,1,3)=COVX(1,1,3,1)
      COVX(2,1,1,3)=COVX(2,1,3,1)
      COVX(3,1,1,3)=COVX(3,1,3,1)
      COVX(1,1,2,3)=COVX(1,1,3,2)
      COVX(2,1,2,3)=COVX(2,1,3,2)
      COVX(3,1,2,3)=COVX(3,1,3,2)
      COVX(1,1,3,3)=D(3)*CN(2,3)/RPQ2
      COVX(2,1,3,3)=D(2)*CN(2,3)/RPQ2
      COVX(3,1,3,3)=CN(1,4)/RPQ2
C GRAVITY ANOMALY ADDED 1992.09.30.
      IF (LDGP) THEN
      COVX(3,1,1,1)=-(CN(1,3)+D3*CN(1,2)+D(13)*D(13)*(CN(3,2)
     *   +D2*CN(3,1))-T*(CN(2,2)+D2*CN(2,1)))/RPQ2
C 2000-04-03
      COVX(3,1,2,1)=-DD(1,2)*DD(1,3)*(CN(3,2)+D2*CN(3,1))/RPQ2
C     COVX(3,1,3,1)=-DD(3,1)*(CN(2,3)+D2*CN(2,2))/RPQ2
      COVX(3,1,3,1)=-DD(1,3)*(CN(2,3)+D2*CN(2,2))/RPQ2
      IF (LTEST) write(*,*)' COVX(3,1,3,1)', COVX(3,1,3,1)
      COVX(3,1,1,2)=COVX(3,1,2,1)
      COVX(3,1,2,2)=-(CN(1,3)+D3*CN(1,2)+D(7)**2*(CN(3,2)
     *   +D2*CN(3,1))-T*(CN(2,2)+D2*CN(2,1)))/RPQ2
      COVX(3,1,3,2)=-D(7)*(CN(2,3)+D2*CN(2,2))/RPQ2
c     COVX(3,1,3,2)=-D(2)*(CN(2,3)+D2*CN(2,2))/RPQ2 cc 2000-04-05
      COVX(3,1,1,3)=COVX(3,1,3,1)
      COVX(3,1,2,3)=COVX(3,1,3,2)
      COVX(3,1,3,3)=-(CN(1,4)+D2*CN(1,3))/RPQ2
      END IF
      GO TO 810
C TWO DERIVATIVES IN BOTH P AND Q. REF(I), EQ. (34)-(46).
  809 R2PQ=RPQ**2
      D3132=D(3)**2+D(13)**2
      D313=D(3)*D(13)
      COVX(1,1,1,1)=(CN(1,3)+CN(1,2)-D2*T*CN(2,2)+D3132*CN(3,2)
     *  +T*CN(2,1)+CN(3,1)*(D2*(CD**2-D3132)+T2)
     *  -CN(4,1)*(D4*CD*SD**2*CP*CQ+T*D3132)
     *  +CN(5,1)*D313**2)/R2PQ
      COVX(2,1,1,1)=(D(2)*D(3)*(CN(3,2)+D(13)**2*CN(5,1)-T*CN(4,1))
     *  +CN(3,1)*D2*(-D(2)*D(3)+DD(2,3)*DD(3,3))
     *  +CN(4,1)*D2*(D313*DD(2,3)+D(2)*D(13)*DD(3,3)))/R2PQ
      CN23=CN(2,3)-CN(2,2)+CN(2,1)
      COVX(3,1,1,1)=(D(3)*(CN23+D(13)**2*(CN(4,2)-CN(4,1))
     *  +T*(CN(3,1)-CN(3,2)))+D2*D(13)*DD(3,3)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(1,2,1,1)=COVX(2,1,1,1)
```



```
      COVX(2,2,1,1)=(CN(1,3)+CN(1,2)-CN(2,2)*D2*T
     *  +CN(3,2)*(D(13)**2+D(2)**2)+CN(2,1)*T
     *  +CN(3,1)*(D2*(DD(2,3)**2-D(13)**2
     *  -D(2)**2)+T2)+CN(4,1)*(D4*D(2)*D(13)*DD(2,3)-T
     *  *(D(13)**2+D(2)**2))+D(13)**2*D(2)**2*CN(5,1))/R2PQ
      COVX(3,2,1,1)=(D(2)*(CN23
     *  +T*(CN(3,1)-CN(3,2))+D(13)**2*(CN(4,2)-CN(4,1)))
     *  +D2*D(13)*DD(2,3)*(CN(3,2)-CN(3,1)))/R2PQ
c suspected error 2002-10-07
c     *  +T*(CN(3,1)-CN(3,2))+D(13)**2*(CN(4,2)-CN(4,1))
c     *  +D2*D(13)*DD(2,3)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(1,3,1,1)=COVX(3,1,1,1)
      COVX(2,3,1,1)=COVX(3,2,1,1)
      COVX(3,3,1,1)=(CN(1,4)-T*CN(2,3)+D(13)**2*CN(3,3))/R2PQ
C
      COVX(1,1,2,1)=(D(7)*D(13)*(CN(3,2)+D(3)**2*CN(5,1)-T*CN(4,1))
     *  +CN(3,1)*D2*(-D(7)*D(13)+DD(3,2)*DD(3,3))
     *  +CN(4,1)*D2*(D313*DD(3,2)+D(7)*D(3)*DD(3,3)))/R2PQ
      COVX(2,1,2,1)=(CN(3,1)*(DD(2,3)*DD(3,2)+DD(2,2)*DD(3,3))
     *  +CN(4,1)*(DD(2,3)*D(3)*D(7)+DD(3,3)*D(2)*D(7)
     *  +DD(2,2)*D(3)*D(13)+DD(3,2)*D(2)*D(13))
     *  +CN(5,1)*D(2)*D(3)*D(7)*D(13))/R2PQ
C ERROR 2000-04-05.
      COVX(3,1,2,1)=(D(3)*D(13)*D(7)*(CN(4,2)-CN(4,1))
     *  +(D(13)*DD(3,2)+DD(3,3)*D(7))*(CN(3,2)-CN(3,1)))/R2PQ
C     COVX(3,1,2,1)=(D(3)*D(13)*D(7)*(CN(3,2)-CN(3,1))
C     *  +(D(13)*DD(3,2)+DD(3,3)*D(7))*(CN(2,2)-CN(2,1)))/R2PQ
      COVX(1,2,2,1)=COVX(2,1,2,1)
      COVX(2,2,2,1)=(D(7)*D(13)*(CN(3,2)+D(2)**2*CN(5,1))
     *  +CN(3,1)*D2*(DD(2,3)*DD(2,2)+D(13)
     *  *DD(4,2))+CN(4,1)*(D2*(D(7)*D(2)*DD(2,3)+D(2)*D(13)*DD(2,2))
     *  -D(7)*D(13)*T))/R2PQ
      COVX(3,2,2,1)=((D(8)*D(13)+D(7)*DD(2,3))*(CN(3,2)-CN(3,1))
     *  +D(7)*D(2)*D(13)*(CN(4,2)-CN(4,1)))/R2PQ
      COVX(1,3,2,1)=COVX(3,1,2,1)
      COVX(2,3,2,1)=COVX(3,2,2,1)
      COVX(3,3,2,1)=D(7)*D(13)*CN(3,3)/R2PQ
C
      COVX(1,1,3,1)= (D(13)*(CN23+D(3)**2*(CN(4,2)-CN(4,1))
     *  +T*(CN(3,1)-CN(3,2)))+D2*D(3)*DD(3,3)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(2,1,3,1)=((DD(3,3)*D(2)+DD(2,3)*D(3))*(CN(3,2)-CN(3,1))
     *  +D(3)*D(13)*D(2)*(CN(4,2)-CN(4,1)))/R2PQ
c     CN33=CN(3,3)-D2*CN(3,2)+CN(3,1)
      CN33=CN(3,3)-CN(3,2)+CN(3,1)
      COVX(3,1,3,1)=(D(3)*D(13)*CN33+DD(3,3)*CN23)/R2PQ
      COVX(1,2,3,1)=COVX(2,1,3,1)
      COVX(2,2,3,1)=(D(13)*(CN23
     *  +D(2)**2*(CN(4,2)-CN(4,1))
     *  +DD(4,1)*(CN(3,2)-CN(3,1)))
     *  +D2*D(2)*DD(2,3)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(3,2,3,1)=(DD(2,3)*CN23+D(2)*D(13)*CN33)/R2PQ
      COVX(1,3,3,1)=COVX(3,1,3,1)
      COVX(2,3,3,1)=COVX(3,2,3,1)
      COVX(3,3,3,1)=D(13)*(CN(2,4)-CN(2,3))/R2PQ
C
      COVX(1,1,1,2)=COVX(1,1,2,1)
      COVX(2,1,1,2)=COVX(2,1,2,1)
      COVX(3,1,1,2)=COVX(3,1,2,1)
      COVX(1,2,1,2)=COVX(1,2,2,1)
      COVX(2,2,1,2)=COVX(2,2,2,1)
      COVX(3,2,1,2)=COVX(3,2,2,1)
      COVX(1,3,1,2)=COVX(1,3,2,1)
      COVX(2,3,1,2)=COVX(2,3,2,1)
      COVX(3,3,1,2)=COVX(3,3,2,1)
C
      D37=D(3)**2+D(7)**2
      COVX(1,1,2,2)=(CN(1,3)+CN(1,2)+CN(2,2)*(-D2*T)
     *  +CN(3,2)*D37+CN(2,1)*T
     *  +CN(3,1)*(D2*(DD(3,2)**2-D37)
```

```
     *  +T2)+CN(4,1)*(D4*D(7)*D(3)*DD(3,2)-T
     *  *D37)+D(3)**2*D(7)**2*CN(5,1))/R2PQ
      COVX(2,1,2,2)=(D(2)*D(3)*(CN(3,2)+D(7)**2*CN(5,1))
     *  +CN(3,1)*D2*(DD(3,2)*DD(2,2)-D(3)*D(2))
     *  +CN(4,1)*(D2*(D(2)*D(7)*DD(3,2)+D(7)*DD(2,2)*D(3))
     *  +D(2)*D(3)*D(19)))/R2PQ
      COVX(3,1,2,2)=(D(3)*(CN23+D(7)**2*(CN(4,2)-CN(4,1))
     *  +DD(1,4)*(CN(3,2)-CN(3,1)))
     *  +D2*DD(3,2)*D(7)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(1,2,2,2)=COVX(2,1,2,2)
      D27=D(2)**2+D(7)**2
      COVX(2,2,2,2)=(CN(1,3)+CN(1,2)-D2*T*CN(2,2)+D27*CN(3,2)
     *  +T*CN(2,1)+(T2-D2*(D27-DD(2,2)**2))*CN(3,1)
     *  +(D4*D(8)*D(2)*D(7)-T*D27)*CN(4,1)
     *  +(D(2)*D(7))**2*CN(5,1))/R2PQ
      COVX(3,2,2,2)=(D(2)*(CN23+D(7)**2
     *  *(CN(4,2)-CN(4,1))-T*(CN(3,2)-CN(3,1)))
     *  +D2*D(7)*D(8)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(1,3,2,2)=COVX(3,1,2,2)
      COVX(2,3,2,2)=COVX(3,2,2,2)
      COVX(3,3,2,2)=(CN(1,4)+D(7)**2*CN(3,3)-T*CN(2,3))/R2PQ
C
      COVX(1,1,3,2)=(D(7)*(CN23
     *  +T*(CN(3,1)-CN(3,2))+D(3)**2*(CN(4,2)-CN(4,1)))
     *  +D2*D(3)*DD(3,2)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(2,1,3,2)=((D(8)*D(3)+D(2)*DD(3,2))*(CN(3,2)-CN(3,1))
     *  +D(7)*D(2)*D(3)*(CN(4,2)-CN(4,1)))/R2PQ
      COVX(3,1,3,2)=(DD(3,2)*CN23+D(3)*D(7)*CN33)/R2PQ
      COVX(1,2,3,2)=COVX(2,1,3,2)
      COVX(2,2,3,2)=(D(7)*(CN23+D(2)**2
     *  *(CN(4,2)-CN(4,1))-T*(CN(3,2)-CN(3,1)))
     *  +D2*D(2)*D(8)*(CN(3,2)-CN(3,1)))/R2PQ
      COVX(3,2,3,2)=(DD(2,2)*CN23+D(2)*D(7)*CN33)/R2PQ
      COVX(1,3,3,2)=COVX(3,1,3,2)
      COVX(2,3,3,2)=COVX(3,2,3,2)
      COVX(3,3,3,2)=D(7)*(CN(2,4)-CN(2,3))/R2PQ
C
      COVX(1,1,1,3)=COVX(1,1,3,1)
      COVX(2,1,1,3)=COVX(2,1,3,1)
      COVX(3,1,1,3)=COVX(3,1,3,1)
      COVX(1,2,1,3)=COVX(2,1,3,1)
      COVX(2,2,1,3)=COVX(2,2,3,1)
      COVX(3,2,1,3)=COVX(3,2,3,1)
      COVX(1,3,1,3)=COVX(1,3,3,1)
      COVX(2,3,1,3)=COVX(2,3,3,1)
      COVX(3,3,1,3)=COVX(3,3,3,1)
C
      COVX(1,1,2,3)=COVX(1,1,3,2)
      COVX(2,1,2,3)=COVX(2,1,3,2)
      COVX(3,1,2,3)=COVX(3,1,3,2)
      COVX(1,2,2,3)=COVX(1,2,3,2)
      COVX(2,2,2,3)=COVX(2,2,3,2)
      COVX(3,2,2,3)=COVX(2,3,3,2)
      COVX(1,3,2,3)=COVX(3,1,3,2)
      COVX(2,3,2,3)=COVX(3,2,3,2)
      COVX(3,3,2,3)=COVX(3,3,3,2)
C
      COVX(1,1,3,3)=(CN(1,4)-T*CN(2,3)+D(3)**2*CN(3,3))/R2PQ
      COVX(2,1,3,3)=D(2)*D(3)*CN(3,3)/R2PQ
      COVX(3,1,3,3)=D(3)*(CN(2,4)-CN(2,3))/R2PQ
      COVX(1,2,3,3)=COVX(2,1,3,3)
      COVX(2,2,3,3)=(CN(1,4)+D(2)**2*CN(3,3)-T*CN(2,3))/R2PQ
      COVX(3,2,3,3)=D(2)*(CN(2,4)-CN(2,3))/R2PQ
      COVX(1,3,3,3)=COVX(3,1,3,3)
      COVX(2,3,3,3)=COVX(3,2,3,3)
      COVX(3,3,3,3)=CN(1,5)/R2PQ
  810 END IF
C
  204 IF (.NOT.LSAT) THEN
```



```
C INTEGERS SPECIFYING THE KINDS OF DIFFERENTIATION WITH RESPECT TO THE
C LATITUDES AND/OR THE LONGITUDES, CF. REF.(A), SECTION 3.
      I = KI(10)
      J = KI(12)
      K = KI(11)
      M = KI(13)
      J1 = KI(14)
      M1 = KI(15)
      IF (.NOT.(LOLDP.OR.LOLDQ)) GO TO 110
C
      IJ = I+J
      IF (I.GT.3) IJ = 5
      KM = K+M
      IF (K.GT.3) KM = 5
C
C COMPUTATION OF THE DERIVATIVES OF ORDER ND WITH RESPECT TO THE LATI-
C TUDES AND THE LONGITUDES, CF. REF.(A), EQ. (43) - (46).
      GO TO (80,81,82,83,84),ND1
   80 COV = C(2)
      GO TO 85
   81 COV = -C(3)*D(I+6*(K-1))
      GO TO 85
   82 COV = D(I)*D(J1)*D(6*(K-1)+1)*D(6*(M1-1)+1)*C(4)+D(IJ+6*(KM-1))
     **C(3)
      GO TO 85
   83 COV = (-D(IJ+6*(KM-1))*C(3)+(D(IJ)*D(6*(KM-1)+1)+D(I+6*(K-1))
     **D(J1+6*(M1-1))+D(I+6*(M1-1))*D(J1+6*(K-1)))*C(4)
     *+D(I)*D(J1)*D(6*(K-1)+1)*D(6*(M1-1)+1)*C(5))
      GO TO 85
   84 COV  = D(IJ+6*(KM-1))*C(3)+(D(IJ+6*(K-1))*D(6*(M-1)+1)
     *+D(I+6*(KM-1))*D(J)+D(J+6*(KM-1))*D(I)+D(IJ+6*(M-1))
     **D((K-1)*6+1)+D(IJ)*D(6*(KM-1)+1)+D(I+6*(K-1))*D(J+6*(M-1))
     *+D(I+6*(M-1))*D(J+6*(K-1)))*C(4)+(D(IJ)*D(6*(K-1)+1)*D(6*(M-1)+1)
     *+D(I+6*(K-1))*D(J)*D(6*(M-1)+1)+D(I+6*(M-1))*D(J)*D(6*(K-1)+1)
     *+D(J+6*(K-1))*D(I)*D(6*(M-1)+1)+D(J+6*(M-1))*D(I)*D(6*(K-1)+1)
     *+D(6*(KM-1)+1)*D(I)*D(J))*C(5)+D(I)*D(J)*D(6*(K-1)+1)*D(6*(M-1)
     *+1)*C(6)
C
C GIVING THE COVARIANCE THE PROPER UNITS.
   85 COV = COV*CI(12)
C
      GO TO 199
  110 CF=CI(12)
      IF (KI(6).EQ.13) CF=CF/D2
      IF (KI(7).EQ.13) CF=CF/D2
      DO 111 IX = 2, ND2
  111 CZ(IX-1) = C(IX)*CF
      CV(1,2) = D0
      CV(2,1) = D0
      CV(2,2) = D0
      GO TO (112, 113, 114, 115, 115), ND1
  112 CV(1,1) = CZ(1)
C  =============================================================
       KZ=1
      GO TO 198
  113 IF (I.EQ.1) GO TO 116
      CV(1,1) = CZ(2)*D(3)
      CV(2,1) = CZ(2)*D(2)
C  =============================================================
       KZ=2
      GO TO 198
  116 CV(1,1) = CZ(2)*D(13)
      CV(1,2) = CZ(2)*D(7)
C  =============================================================
       KZ=3
      GO TO 198
  114 IF (I.GT.1) GO TO 117
      CV(1,2) = CZ(3)*D(19)*D(31)
      CV(1,1) = CZ(3)*D(7)*D(13)*D2
```

```
C  ============================================================
      KZ=4
      GO TO 198
  117 IF (K.GT.1) GO TO 118
      CV(2,1) = CZ(3)*D(4)*D(6)
      CV(1,1) = CZ(3)*D(2)*D(3)*D2
C  ============================================================
      KZ=5
      GO TO 198
  118 CV(1,1) = CZ(2)*D(15)+CZ(3)*D(13)*D(3)
      CV(2,2) = CZ(2)*D(8) +CZ(3)*D(2)*D(7)
      CV(1,2) = CZ(2)*D(9) +CZ(3)*D(3)*D(7)
      CV(2,1) = CZ(2)*D(14)+CZ(3)*D(13)*D(2)
C  ============================================================
      KZ=6
C FIRST ORDER HORIZONTAL DERIVATIVES IN BOTH P AND Q.
      GO TO 198
  115 CONTINUE
C
      IIX=2
      DO 119 IX = 1, 2
      IIY=2
      DO 120 JX = 1, 2
      IF (ND.EQ.4) GO TO 121
C SECOND ORDER HORIZONTAL DERIVATIVE IN P OR Q.
      IX1=IX
      JX1=JX
      IF (KI(6) .GE. 12) GO TO 122
      CF = JX
      JX1=IIY
      I = J2(IX)
      J1 = 1
      K = I4(JX)
      M1 = I3(JX)
      GO TO 123
  122 CF = IX
      IX1=IIX
      I = I4(IX)
      J1 = I3(IX)
      K = J2(JX)
      M1 = 1
  123 K6 = 6*(K-1)
      M6 = 6*(M1-1)
      CV(IX1,JX1) = (CZ(3)*(D(I+K6)*D(J1+M6)+D(J1+K6)*D(I+M6))
     * +CZ(4)*D(I)*D(J1)*D(K6+1)*D(M6+1))*CF
C  ============================================================
      KZ=7
      GO TO 120
  121 I = I4(IX)
      J = I3(IX)
      K = I4(JX)
      M = I3(JX)
      K6 = 6*(K-1)
      M6 = 6*(M-1)
      CV(IIX,IIY) = (CZ(3)*(D(I+K6)*D(J+M6)+D(I+M6)*D(J+K6))
     *  +CZ(4)*(D(J)*(D(I+K6)*D(M6+1)+D(I+M6)*D(K6+1))
     *         +D(I)*(D(J+K6)*D(M6+1)+D(J+M6)*D(K6+1)))
     *  +CZ(5)*D(I)*D(J)*D(K6+1)*D(M6+1))*IX*JX
C ============================================================
      KZ=8
  120 IIY=1
  119 IIX=1
  198 COV = CV(KI(24),KI(25))
C ============================================================
      IF (LTEST)WRITE(6,7788) KZ,I,J,K,M,CV(1,1),CV(1,2),CV(2,1),
     *CV(2,2)
 7788 FORMAT(/' KZ, I, J, K, M, CV(1,1), CV(1,2), ',
     *' CV(2,1) CV(2,2)'/1X,5I4,4F10.4)
  199 RETURN
```



```
      ELSE
       COV=COVX(KSAT(KP,1),KSAT(KP,2),KSAT(KQ,1),KSAT(KQ,2))
       IF (KP.EQ.15.AND.KQ.NE.15)
     * COV=COV-COVX(2,2,KSAT(KQ,1),KSAT(KQ,2))
C CHANGE, SO THAT UNITS ARE M, MGAL OR EU. 1992.08.26.
       IF (KP.EQ.6.OR.KP.EQ.7) THEN
        C11P=1.0D5
       ELSE
C CHANGE 2003-04-01.
C     C11P=C11(KP)/(CR(10)**K19(KP))
        C11P=C11(KP)
       END IF
       IF (KQ.EQ.6.OR.KQ.EQ.7) THEN
        C11Q=1.0D5
       ELSE
C     C11Q=C11(KQ)/(CR(11)**K19(KQ))
        C11Q=C11(KQ)
       END IF
       CFA=C11P*C11Q
       IF (KP.NE.15.AND.KQ.EQ.15)
     * COV=COV-COVX(KSAT(KP,1),KSAT(KP,2),2,2)
       IF (KP.EQ.15.AND.KQ.EQ.15)
     * COV=COV-COVX(1,1,2,2)-COVX(2,2,1,1)+COVX(2,2,2,2)
       COV=COV*CFA
C 2000-04-04.
       IF (LTEST)WRITE(*,*)' KSAT ',KSAT(KP,1),KSAT(KP,2),KSAT(KQ,1),
     * KSAT(KQ,2),' COV ',COV,' CFA ',CFA
       END IF
       RETURN
       END
       FUNCTION VAR(SM,IS,KP,DRM,AAI,HP,IMAX1,LMEAN,CP,SP,LSAT,SROT)
C PROGRAMMED FEB 1985 BY C.C.TSCHERNING. UPDATE: OCT.08, 2002.
C THE FUNCTION COMPUTES THE VARIANCE OF A SIGNAL QUANTITY OF TYPE
C KP USING COVBX AND COVCX.
C IF DOUBLE PRECISION IS NEEDED, ACTIVATE THE FOLLOWING STATEMENT:
       implicit none
       LOGICAL LT,LF,LSUM,LOCAL,LMEAN,LSAT,LTESTS,LX,LNX,LFINI
       REAL*8 CI,CR,SIGMA0,SIGMA,HMAX,D,D0,D1,D2,D3,D4,D5,RE,RADSEC,
      *PI,GM,STEPN,COSSTN,SINSTN,STEPE,COSSTE,SINSTE,
      *COST2P,SINT2P,FILTER,STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE,
      *COST2Q,SINT2Q,COVX,CIX,CFA,SM,SROT,AAI,DRM,RP,HP,CVV,STEQQN,
      *VAR,COMEAN,CP,SP
       integer ksat,ki,n1,n2,nfilte,ndx1,ndx2,ndp,ndq,nwar,imax1,
      *kp,is,itcoun
C
       COMMON /CMCOV/CI(24),CR(56),SIGMA0(2200),SIGMA(2200),HMAX,
      *D(40),KI(37),N1,N2,LOCAL,LSUM,LFINI
       COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
       COMMON /CMEAN/STEPN,COSSTN,SINSTN,STEPE,COSSTE,SINSTE,
      *COST2P,SINT2P,FILTER(11),NFILTE
       COMMON /CMEAQ/STEQN,COSSQN,SINSQN,STEQE,COSSQE,SINSQE,
      *COST2Q,SINT2Q
       COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFA,KSAT(17,2),
      *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
       DIMENSION SM(2200),SROT(3,3)
C
       CI(8) = AAI
       CI(9) = (RE+DRM)**2
       CI(10)= DRM
       CI(20)= D1
       N1 = IMAX1
       KI(6) = KP
       KI(7) = KP
       RP = RE+HP
c      CALL COVBX(SM,LF,IS)
       CALL COVBX(SM,LSAT,IS)
       CR(1) = D1
       CR(2) = HP
       CR(3) = HP
```

```
      CR(4) = D0
      CR(5) = D0
      CR(6) = D1
      CR(7) = D1
      CR(8) = D0
      CR(9) = D1
      CR(10)= GM/(RP*RP)
      CR(11) = CR(10)
      IF (.NOT.LMEAN) THEN
c      CALL COVCX(SM,CVV,IS,LF)
       CALL COVCX(SM,CVV,IS,LSAT)
       IF (LSAT) THEN
        CALL COVROT(SROT,SROT)
        IF (LTESTS) WRITE(*,101)COVX
 101    FORMAT(6D12.4)
c       CVV=COVX(KSAT(KP,1),KSAT(KP,2),KSAT(KP,1),KSAT(KP,2))
        IF (KP.NE.25) THEN
         CVV=COVX(KSAT(KP,1),KSAT(KP,2),KSAT(KP,1),KSAT(KP,2))
C CHANGE 2002-10-23.
        ELSE
C DDT/DXX-DDT/DYY IN P.
         CVV=
     *   (COVX(KSAT(14,1),KSAT(14,2),KSAT(14,1),KSAT(14,2))
     *   -COVX(KSAT(12,1),KSAT(12,2),KSAT(14,1),KSAT(14,2))
     *   +COVX(KSAT(14,1),KSAT(14,2),KSAT(12,1),KSAT(12,2))
     *   -COVX(KSAT(12,1),KSAT(12,2),KSAT(12,1),KSAT(12,2)))
        END IF
        IF (LTESTS) WRITE(*,100)CVV,KSAT(KP,1),KSAT(KP,2),KP
 100    FORMAT(' CVV, KP ',D14.6,3I3)
       END IF
      ELSE
C CHANGE 2001-07-15.
      STEQQN=STEQN
      STEQN=STEPN
      COSSQN=COSSTN
      SINSQN=SINSTN
      STEQE=STEPE
      COSSQE=COSSTE
      SINSQE=SINSTE
      COST2Q=COST2P
      SINT2Q=SINT2P
      CVV=COMEAN(SM,IS,0,CP,SP,D1,D0,CP,SP,D1,D0,5,5,LF,LF,LF)
      END IF
C CHANGE 2000-04-11 AND 2002-09-30 BY CCT.
      IF (LSAT) THEN
       IF (KP.EQ.6.OR.KP.EQ.7.OR.KP.EQ.2) THEN
       CVV=CVV*1.0D10
C CONVERSION TO MGAL.
C      CVV=CVV*(CR(10)*1.0D5/RADSEC)**2
       ELSE
        IF (KP.GT.7.OR.KP.EQ.5) THEN
c scaling for 2-order derivatives (to EU**2).
         CVV=CVV*1.0D18
C SCALING FOR 2*TXY. 2002-11-26.
         IF (KP.EQ.13)CVV=CVV*4.0d0
         IF (LTESTS) WRITE(*,*)' KP, CVV ',KP,CVV
        END IF
       END IF
      END IF
      VAR = CVV
C CHANGE 2001-07-15.
      STEQN=STEQQN
      RETURN
      END
      SUBROUTINE ATBA(A,B,C)
C PROGRAMMED AUG 89 BY C.C.TSCHERNING.
C THE SUBROUTINE WILL COMPUTE THE PRODUCT OF THE 3*3 MATRICES A TRANS-
C POSED, B AND A AND STORE THE RESULT IN C.
      implicit none
```



```
      real*8 A,B,C,D,E
      integer j,k,n
      DIMENSION A(3,3),B(3,3),C(3,3),D(3,3),E(3,3)
C A TRANSPOSED TIMES B STORED IN D: :
      DO 30 K=1,3
      DO 30 J=1,3
      D(K,J)=0.0D0
      DO 30 N=1,3
c  30 D(K,J)= A(K,N)*B(N,J)+D(K,J)
   30 D(K,J)= A(N,K)*B(N,J)+D(K,J)
C
C D TIMES A STORED IN E:
      DO 40 K=1,3
      DO 40 J=1,3
      E(K,J)=0.0D0
      DO 40 N=1,3
c  40 E(K,J)=E(K,J)+D(K,N)*A(J,N)
   40 E(K,J)=E(K,J)+D(K,N)*A(N,J)
C
      DO 50 K=1,3
      DO 50 J=1,3
   50 C(K,J)=E(K,J)
      RETURN
      END
      SUBROUTINE AXV(A,V)
C THE SUBRIUTINE WILL COMPUTE THE PRODUCT OF THE MATRIX A AND THE
C VECTOR V AND RETURN IT IN V. PROGRAMMED 1990.11.03 BY CCT.
      implicit none
      real*8 a,v,y
      integer i,j
      DIMENSION A(3,3),V(3),Y(3)
      DO 10 I=1,3
      Y(I)=V(I)
   10 V(I)=0.0D0
      DO 20 I=1,3
      DO 20 J=1,3
   20 V(I)=A(I,J)*Y(J)+V(I)
      RETURN
      END
      SUBROUTINE COVROT(SROTP,SROTQ)
C THE SUBROUTINE WILL COMPUTE THE ROTATED COVARIANCE MATRIV OR VECTOR
C USING THE ROTATION MATRICES SRORP, SROTQ ASSOCIATED WITH THE POINTS
C P, Q, RESPECTIVELY. SEE REF(I), SECTION 3.
C PROGRAMMED BY C.C.TSCHERNING, GEOPHYSICAL INSTITUTE, UNIVERSITY OF
C COPENHAGEN, JUNE, 1991.
C (I) Tscherning, C.C.: Computation of covariances of derivatives of the
C     anomalous gravity potential in a rotated reference frame.
C     Manuscripta Geodaetica, Vol. 18, no. 3, pp. 115-123, 1993.
C LAST UPDATE 2002-10-24.
C
      implicit none
      REAL*8 COVX,CIX,CFA,SROTP,SROTQ,V,A
      INTEGER KSAT,NDX1,NDX2,NDP,NDQ,NWAR,NCASE,IM,JM,I,J
      LOGICAL LSATS,LNX,LX
C
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFA,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP, NDQ,NWAR,LX(7,5),LNX(7,5),LSATS
      DIMENSION SROTP(3,3),SROTQ(3,3),V(3),A(3,3)
      NCASE=NDP+1+NDQ*3
C
      GO TO (801,802,803,804,805,806,807,808,809),NCASE
C 1 DERIV. IN P, NONE IN Q.
  802 DO 831 IM=1,3
  831 V(IM)=COVX(IM,1,1,1)
      CALL AXV(SROTP,V)
      DO 812 IM=1,3
  812 COVX(IM,1,1,1)=V(IM)
      GO TO 801
C
```

```
C 2 DERIV. IN P, NONE IN Q.
  803 DO 823 IM=1,3
      DO 823 JM=1,3
  823 A(IM,JM)=COVX(IM,JM,1,1)
      CALL ATBA(SROTP,A,A)
      DO 824 IM=1,3
      DO 824 JM=1,3
  824 COVX(IM,JM,1,1)=A(IM,JM)
      GO TO 801
C
C NO DERIV. IN P, 1 IN Q.
  804 DO 832 IM=1,3
  832 V(IM)=COVX(1,1,IM,1)
      CALL AXV(SROTQ,V)
      DO 833 IM=1,3
  833 COVX(1,1,IM,1)=V(IM)
      GO TO 801
C
C 1 DERIV. IN BOTH P AND Q.
  805 DO 834 IM=1,3
      DO 835 JM=1,3
  835 V(JM)=COVX(JM,1,IM,1)
      CALL AXV(SROTP,V)
      DO 836 JM=1,3
  836 COVX(JM,1,IM,1)=V(JM)
  834 CONTINUE
      DO 844 IM=1,3
      DO 845 JM=1,3
  845 V(JM)=COVX(IM,1,JM,1)
      CALL AXV(SROTQ,V)
      DO 846 JM=1,3
  846 COVX(IM,1,JM,1)=V(JM)
  844 CONTINUE
      GO TO 801
C
C 2 DERIV. IN P, 1 IN Q.
  806 DO 854 I=1,3
      DO 855 IM=1,3
      DO 855 JM=1,3
  855 A(IM,JM)=COVX(IM,JM,I,1)
      CALL ATBA(SROTP,A,A)
      DO 856 IM=1,3
      DO 856 JM=1,3
  856 COVX(IM,JM,I,1)=A(IM,JM)
  854 CONTINUE
      DO 955 IM=1,3
      DO 955 JM=1,3
      DO 954 I=1,3
  954 V(I)=COVX(IM,JM,I,1)
      CALL AXV(SROTQ,V)
      DO 956 I=1,3
  956 COVX(IM,JM,I,1)=V(I)
  955 CONTINUE
      GO TO 801
C
C NO DERIV. IN P, 2 IN Q.
  807 DO 923 IM=1,3
      DO 923 JM=1,3
  923 A(IM,JM)=COVX(1,1,IM,JM)
      CALL ATBA(SROTQ,A,A)
      DO 924 IM=1,3
      DO 924 JM=1,3
  924 COVX(1,1,IM,JM)=A(IM,JM)
      GO TO 801
C
C ONE DERIV. IN P, 2 IN Q.
  808 DO 754 I=1,3
      DO 755 IM=1,3
      DO 755 JM=1,3
```



```
  755 A(IM,JM)=COVX(I,1,IM,JM)
      CALL ATBA(SROTQ,A,A)
      DO 756 IM=1,3
      DO 756 JM=1,3
  756 COVX(I,1,IM,JM)=A(IM,JM)
  754 CONTINUE
      DO 975 IM=1,3
      DO 975 JM=1,3
      DO 974 I=1,3
  974 V(I)=COVX(I,1,IM,JM)
      CALL AXV(SROTP,V)
      DO 976 I=1,3
  976 COVX(I,1,IM,JM)=V(I)
  975 CONTINUE
      GO TO 801
C
C 2 DERIV. IN P AND Q.
  809 DO 540 I=1,3
      DO 540 J=1,3
      DO 555 IM=1,3
      DO 555 JM=1,3
  555 A(IM,JM)=COVX(IM,JM,I,J)
      CALL ATBA(SROTP,A,A)
      DO 556 IM=1,3
      DO 556 JM=1,3
  556 COVX(IM,JM,I,J)=A(IM,JM)
  540 CONTINUE
      DO 541 I=1,3
      DO 541 J=1,3
      DO 565 IM=1,3
      DO 565 JM=1,3
  565 A(IM,JM)=COVX(I,J,IM,JM)
      CALL ATBA(SROTQ,A,A)
      DO 456 IM=1,3
      DO 456 JM=1,3
  456 COVX(I,J,IM,JM)=A(IM,JM)
  541 CONTINUE
C
  801 RETURN
      END
      SUBROUTINE PAZIM(RLATP,RLONGP,COSLAP,SINLAP,COSLOP,SINLOP,
     *CAZP,SAZP,COSDT,SINDT,LTEST)
c THE SUBROUTINE WILL FROM A POINT WITH LATITUDE AND LONGITUDE
C SPECIFIED IN THE CALL PRODUCE THE CORRESPONDING VALUES IN A
C NEW POINT IN DISTANCE DT  AND AZIMUTH GIVEN BY
C COS AND SIN - CAZP, SAZP.
C PROGRAMMED BY C.C.TSCHERNING, OCT. 92. LAST CHANGE: 2002-10-24.
      implicit none
      real*8 rlatp,rlongp,coslap,sinlap,coslop,sinlop,cazp,sazp,
     *cosdt,sindt,sidlon,codlon,dlong,raddeg,dlatp,dlongp
      LOGICAL LTEST
      RLONGP=ATAN2(SINLOP,COSLOP)
      SINLAP=COSLAP*SINDT*CAZP+SINLAP*COSDT
      COSLAP=SQRT(1.0D0-SINLAP**2)
      SIDLON=SINDT*SAZP/COSLAP
      CODLON=SQRT(1.0D0-SIDLON**2)
      DLONG=ATAN2(SIDLON,CODLON)
      RLONGP=RLONGP+DLONG
      RLATP=ATAN2(SINLAP,COSLAP)
      COSLOP=COS(RLONGP)
      SINLOP=SIN(RLONGP)
      RADDEG=180.0D0/3.1415926535D0
      DLATP=RADDEG*RLATP
      DLONGP=RADDEG*RLONGP
      IF (LTEST) WRITE(*,*)' PAZIM - LAT,LONG=',DLATP,DLONGP
C
      RETURN
      END
      DOUBLE PRECISION FUNCTION COZERO(PSI,R,MODEL)
```

```
C THE SUBROUTINE WILL EVALUATE A COVARIANCE FUNCTION WHICH
C IS ZERO AFTER THE DISTANCE (PSI) IS LARGER THAN OR EQUAL TO 2*R.
C PROGRAMMED APRIL 1996 BY C.C.TSCHERNING, GEOPHYSICAL
C DEPARTMENT. LAST CHANGE 2002-10-24.
C     PSI -  SPHERICAL DISTANCE IN RADIANS
C     R   -  2*R DISTANCE IN RADIANS FOR WHICH FUNCTION IS ZERO.
C     MODEL - PARAMETER TO DISTINGUISH BETWEEN DIFFERENT MODELS.
C
      implicit none
      real*8 d0,d2,d3,pi,psi2,psi,psi3,r2,r,r4
      integer model
c
      PI = 3.1415926535D0
      D0=0.0D0
      D2=2.0D0
      D3=3.0D0
      PSI2=PSI**2
      PSI3=PSI**3
      GO TO (10,20,30) MODEL
  10  R2=R*R
      R4=R2*R2
      IF (PSI.LT.D2*R) THEN
      COZERO= R4*PI*(R2/D3-PSI2/D2)
     *+(R2*(R2*PSI+4.0D0/D3*PSI3)-PSI3*PSI2/12.0D0)/D3
     **SQRT(R2-PSI2/4.0D0)
     *+(R2*R2*(PSI2-D2/D3*R2))*ASIN(PSI/(D2*R))
      ELSE
      COZERO=D0
      END IF
      GO TO 30
  20  COZERO=D0
  30  CONTINUE
      RETURN
      END
      SUBROUTINE FINDC1(PSI1,R,MODEL)
C PROGRAMMED BY CCT, LAST CHANGE 2003-03-18.
      IMPLICIT NONE
      REAL*8 D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,PSI1,R,C0,C1,DPSI,
     *TEST1,TEST2,TEST3,PSI,C2,C3,COZERO
      INTEGER ITCOUN,MODEL,K
      LOGICAL LF,LT
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
C
      C0= COZERO(D0,R,MODEL)
      PSI=D0
      C1=C0
      DPSI=5.0D-4
      TEST3=D1
  10  PSI= PSI+DPSI
      C2=c1
      C1=COZERO(PSI,R,MODEL)
      TEST1=TEST3
      TEST3=C1/C0
      IF (TEST3.GT.1.0D0.OR.TEST3.LT.D0)
     *WRITE(*,*)' WARNING ',C0,C2,C1
      IF (TEST3.GT.0.5D0) GO TO 10
      DPSI=DPSI/D2
      PSI=PSI-DPSI
      K=0
  50  C3=COZERO(PSI,R,MODEL)
      K=K+1
      TEST2=C3/C0
      DPSI=DPSI/D2
      IF (TEST2.GT.0.5D0.AND.TEST3.LT.0.5D0) THEN
      TEST1=TEST2
      PSI=PSI+DPSI
      ELSE
      TEST3=TEST2
      PSI=PSI-DPSI
```



```
      END IF
      IF (K.LT.115.AND.ABS(0.5D0-TEST2).GT.1.0D-8) GO TO 50
      PSI1=PSI
      RETURN
      END
      DOUBLE PRECISION FUNCTION FINDR(PSI1,MODEL)
C THE SUBROUTINE DETERMINES THE VALUE OF A PARAMETER R IN
C A FINITE COVARAINCE FUNCTION.
C PROGRAMMED APRIL 1996 BY C.C.TSCHERNING, DEPARTMENT OF GEOPHYSICS,
C UNIVERSITY OF COPENHAGEN. LAST CHANGE 2002-10-24.
      implicit none
      real*8 pi,r0,psi1,rstep,c0,c1,c2,c3,c4,c5,test2,test3,test1,
     *rd0,cozero
c
      integer model,k
c
      pi=3.1415926535d0
      R0=psi1*1.2
      Rstep=0.5*psi1
      K=0
  40  c0=cozero(0.0d0,R0,MODEL)
      c1=cozero(psi1,R0,MODEL)
c     write(*,30)R0,c0,c1,c1/c0
  30  format(4d16.9)
      c2=cozero(0.0d0,R0+Rstep,MODEL)
      c3=cozero(psi1,R0+Rstep,MODEL)
c     write(*,30)R0+Rstep,c2,c3,c3/c2
      c4=cozero(0.0d0,R0-Rstep,MODEL)
      c5=cozero(psi1,R0-Rstep,MODEL)
c     write(*,30)R0-Rstep,c4,c5,c5/c4
      test2=c1/c0
      test3=c3/c2
      test1=c5/c4
      Rstep=Rstep/2.0d0
      if (test2.lt.0.5.and.test3.gt.0.5) then
      R0=R0+Rstep
      else
      R0=R0-Rstep
      end if
      k=k+1
      if (k.lt.115.and.abs(0.5-test2).gt.1.0d-8) go to 40
      rd0=r0*180.0/pi
C     write(*,60)k,r0,rd0,test2
  60  format(i5,2f10.6,d15.7)
      FINDR=R0
      RETURN
      END
      subroutine spharm(slat,clat,sjlo,cjlo,r,i0,j0,idif,lfull)
c Calculation of the values and the up to 2. order derivatives
c of solid spherical harmonic functions  Y(i0,j0)(lat,lon,r) using
c recursion based on Y(i0-1,j0), Y(i0-2,j0)  when j0 .ne. j0.
c Otherwise the recursion is based on Y(i0-1,j0-1) and
c Y(i0-2,j0-1). The calculation of first order derivatives at the
c poles is done using a recursion formulae, where the cos(lat) the is
c eliminated. The second order derivative with respect to x is at
c the poles calculated using the Laplace equation.
c programmed february 1999  by C.C.Tscherning. Last change 2002-10-24.
c References:
c Tscherning, C.C.: On the Chain-Rule Method for Computing Potential
c Derivatives. Manuscripta Geodaetica, Vol. 1, pp. 125-141, 1976.
c Tscherning, C.C. and K.Poder: Some Geodetic applications of Clenshaw
c Summation. Bolletino di Geodesia e Scienze Affini, Vol. XLI, no. 4,
c pp. 349-375, 1982.
c
c variables at call: slat, clat: sine and cosine of latitude, r the
c size of the radius vector, idif the maximal order of differentiation
c (up to 2),  cjlo, cjlo: cos and sin of j*longitude,
c lfull a logical variable true if fully normalized functions
c are used.
```

```
c Last change 2000-04-24 by cct.
c
      implicit none
      REAL*8 D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,SUMIJ,CCCIJ,
     *sq2,ys,yc,vv,v1,gs,gc,dds,
     *ddc,root0,pii,pim0,pim1,pim2,dlp,dlp0,dlp1,dlp2,dap,dap0,
     *dap1,dap2,ddap,ddap0,ddap1,ddap2,ddal0,ddal1,vi,r,q,ddal,
     *rq,clat,slat,a,ddal2,b,pm,q2,q3,v,cjlo,sjlo,ddc0,fact
c
      integer i,i0,iimax,j,j0,j1,jold,idif,iiold,k,n,itcoun
c
      parameter (iimax=20000)
      logical lfull,LSPHAR,LTSPH,lf,lt
c
      COMMON /DCON/D0,D1,D2,D3,D4,D5,RE,RADSEC,PI,GM,ITCOUN,LF,LT
      common /con3/SUMIJ(32761),CCCIJ(32761),
     *sq2,ys,yc,vv,v1,gs(3),gc(3),dds(3,3),
     *ddc(3,3),iiold,jold,LSPHAR,LTSPH
      common /rrsph/root0(iimax)
c root is a precomputed square root-table (root0(1)=0 !).
      common /sphold/pii,pim0,pim1,pim2,dlp,dlp0,dlp1,dlp2,dap,dap0,
     *dap1,dap2,ddap,ddap0,ddap1,ddap2,ddal0,ddal1,vi
      q=re/r
      j=j0
      i=i0
      j1=j+1
      if (i.eq.j) then
      if (j.ne.(jold+1)) write(*,*)' WARNING j '
      pim2=d0
      pim1=d0
c     write(*,*)r,slat,clat,sjlo,cjlo
      if (idif.gt.0) then
      dlp0=d0
      dlp1=d0
      dlp2=d0
      dap0=d0
      dap1=d0
      if (idif.gt.1) then
      ddal0=d0
      ddal1=d0
      ddap1=d0
      ddap2=d0
      end if
      end if
      if (i.eq.0) then
      pii=q
      pim0=pii
      if (idif.gt.0) then
      dlp=d0
      dlp0=dlp
      dap=d0
      dap0=dap
      if (idif.gt.1) then
      ddap=d0
      ddap0=dap
      ddal=d0
      ddal0=ddal
      end if
      end if
      else
      rq=root0(2*i)/root0(2*i+1)*q
      if (idif.gt.1) then
      ddap=(clat*ddap-d2*slat*dap-clat*pii)*rq
      ddap0=ddap
      ddal=dap*rq
      ddal0=ddal
      end if
      if (idif.gt.0) then
      dap=(-slat*pii+clat*dap)*rq
```



```
      dap0=dap
      dlp=pii*rq
      dlp0=dlp
      end if
      pim0=pii*clat*rq
      pii=pim0
      end if
      else
      if (j.ne.jold.or.i.ne.(iiold+1)) write(*,*)' WARNING i,j '
      a=(2*i-1)/(root0(i+j+1)*root0(i-j+1))*q
      if (idif.gt.1) then
      ddap2=ddap1
      ddap1=ddap0
      ddap0=(slat*ddap0+d2*clat*dap0-slat*pim0)*a
      ddal2=ddal1
      ddal1=ddal0
c correction 1999-02-28 by cct - forgottem until 2000-04-25.
c     ddal0=a*(clat*dlp0+slat*ddap0)
      ddal0=a*(clat*dlp0+slat*ddal0)
      end if
      if (idif.gt.0) then
      dap2=dap1
      dap1=dap0
      dap0=a*(clat*pim0+slat*dap0)
      dlp2=dlp1
      dlp1=dlp0
      dlp0=a*slat*dlp0
      end if
      pim2=pim1
      pim1=pim0
      pim0=a*slat*pim1
      if (i.gt.j) then
      b=-q**2*root0(i-j)*root0(i+j)/(root0(i-j+1)*root0(i+j+1))
      if (idif.gt.1) then
      ddap0=ddap0+b*ddap2
      ddal0=ddal0+b*ddal2
      end if
      if (idif.gt.0) then
      dap0=dap0+b*dap2
      dlp0=dlp0+b*dlp2
      end if
      pm= b*pim2
      pim0=pim0+pm
      end if
      end if
      q2=q*q
      q3=q2*q
      v=pim0
c     yc=v*q*cjlo
c     ys=v*q*sjlo
      yc=v*cjlo
      ys=v*sjlo
      if (idif.gt.0) then
      gs(3)=(-i-1)*q*ys
      gc(3)=(-i-1)*q*yc
      gc(1)=dlp0*q*(-sjlo)*j
      gs(1)=dlp0*q*( cjlo)*j
      gc(2)=dap0*q*cjlo
      gs(2)=dap0*q*sjlo
      if (idif.gt.1) then
      ddc(1,2)=ddal0*q2*(-sjlo)*j
      dds(1,2)=ddal0*q2*( cjlo)*j
      ddc(2,1)=ddc(1,2)
      dds(2,1)=dds(1,2)
      ddc(1,3)=(-i-2)*q2*dlp0*(-sjlo)*j
      dds(1,3)=(-i-2)*q2*dlp0*( cjlo)*j
      ddc(3,1)=ddc(1,3)
      dds(3,1)=dds(1,3)
      ddc(2,2)=(ddap0+(-i-1)*v)*q2*cjlo
```

```
      dds(2,2)=(ddap0+(-i-1)*v)*q2*sjlo
      ddc(2,3)=(-i-2)*q2*dap0*cjlo
      dds(2,3)=(-i-2)*q2*dap0*sjlo
      ddc(3,2)=ddc(2,3)
      dds(3,2)=dds(2,3)
      ddc(3,3)=gc(3)*(-i-2)*q
      dds(3,3)=gs(3)*(-i-2)*q
      if (abs(clat).gt.1.0d-10) then
      ddc0=q2*((-i-1)*v-(slat*dap0+v*j**2/clat)/clat)
      ddc(1,1)=ddc0*cjlo
      dds(1,1)=ddc0*sjlo
      else
      ddc(1,1)=-ddc(2,2)-ddc(3,3)
      dds(1,1)=-dds(2,2)-dds(3,3)
      end if
      end if
      end if
      iiold=i
      jold=j
      if (lfull) then
c normalisation.
      if (j.eq.0) then
      fact= root0(2*i+2)
      else
      fact= root0(2*i+2)*sq2
      end if
      v=v*fact
      yc=yc*fact
      ys=ys*fact
      do 25, k=1,3
C ERROR 2000-05-02 DETECTED.
      gc(k)=gc(k)*fact
      gs(k)=gs(k)*fact
      do 25, n=1,3
      ddc(k,n)=ddc(k,n)*fact
      dds(k,n)=dds(k,n)*fact
  25  continue
c
      end if
      return
      end
      logical function checkc(npos)
c using the Laplace equation to check to covariances.
c programmed 2002-10-07 by C.C.Tscherning, latest update: 2002-11-26.
c Input:
c  npos - call - used to indicate from where the subroutine is called.
c  COVX - CSAT - holds covariances. Two first subscripts realted
c                to one point (P) and the last two to a second point (Q).
c                index 1: East derivative, 2: North derivative,
c                3: up derivative (radius vector).
c ndp,ndq CSAT - number of derivatives in P, Q, respectively.
c
c Output
c nwar - csat - number of warnings
c if lout is true, output of Laplace equation, sum of absolute value of
c    the 3 terms, the 3 terms.
c
      implicit none
      integer i,j,npos,ndp,ndq,nwar,ksat,ndx1,ndx2,ncase
      real*8 test1,test2,test4,test5,covx,cix,cfx,atest1,atest2
      logical lout,ltests,lx,lnx,check,lf
      COMMON /CSAT/COVX(3,3,3,3),CIX(7,5),CFX,KSAT(17,2),
     *NDX1(5),NDX2(5),NDP,NDQ,NWAR,LX(7,5),LNX(7,5),LTESTS
c
      lout=.true.
      lf=.false.
      checkc=.true.
      check=.true.
      j=1
```



```
      NCASE=NDP+1+NDQ*3
      go to (810,810,803,810,810,806,807,808,809),ncase
c zero in P, 2 in Q.
 807  test1=covx(1,1,1,1)+covx(1,1,2,2)+covx(1,1,3,3)
      atest1=abs(test1)
      test4=abs(covx(1,1,1,1))+abs(covx(1,1,2,2))+abs(covx(1,1,3,3))
      if (atest1.gt.test4*1.0d-4.and.atest1.gt.1.0d-10) then
       check=lf
       if (lout)
     * write(*,10)npos,i,j,test1,test4,covx(1,1,1,1),covx(1,1,2,2),
     * covx(1,1,3,3)
       nwar=nwar+1
      end if
      go to 810
c
c TWO in P, ONE IN Q.
 806  do I=1,3
       test2=covx(1,1,i,1)+covx(2,2,i,1)+covx(3,3,i,1)
       atest2=abs(test2)
       test5=abs(covx(1,1,i,1))+abs(covx(2,2,i,1))+abs(covx(3,3,i,1))
       if (atest2.gt.test5*1.0d-4.and.atest2.gt.1.0d-20) then
        check=lf
        if (lout)
     *  write(*,10)npos,i,j,test1,test4,covx(i,j,1,1),covx(i,j,2,2),
     *  covx(i,j,3,3)
        nwar=nwar+1
       end if
      end do
      go to 810
c
 803  test1=covx(1,1,1,1)+covx(2,2,1,1)+covx(3,3,1,1)
      atest1=abs(test1)
      test4=abs(covx(1,1,1,1))+abs(covx(2,2,1,1))+abs(covx(3,3,1,1))
      if (atest1.gt.test4*1.0d-4.and.atest1.gt.1.0d-10) then
       check=lf
       if (lout)
     * write(*,10)npos,i,j,test1,test4,covx(1,1,1,1),covx(2,2,1,1),
     * covx(3,3,1,1)
       nwar=nwar+1
      end if
      go to 810
c
c 1 in P 2 in Q.
 808  do i=1,3
      test1=covx(i,1,1,1)+covx(i,1,2,2)+covx(i,1,3,3)
      atest1=abs(test1)
      test4=abs(covx(i,1,1,1))+abs(covx(i,1,2,2))+abs(covx(i,1,3,3))
       check=lf
       if (atest1.gt.test4*1.0d-4.and.atest1.gt.1.0d-20) then
        if (lout)
     *  write(*,10)npos,i,j,test1,test4,covx(i,1,1,1),covx(i,1,2,2),
     *  covx(i,1,3,3)
        nwar=nwar+1
      end if
      end do
      go to 810
c
c TWO IN BOTH P AND Q.
 809  do i=1,3
      do j=1,3
       test1=covx(i,j,1,1)+covx(i,j,2,2)+covx(i,j,3,3)
       atest1=abs(test1)
       test2=covx(1,1,i,j)+covx(2,2,i,j)+covx(3,3,i,j)
       atest2=abs(test2)
       test4=abs(covx(i,j,1,1))+abs(covx(i,j,2,2))+abs(covx(i,j,3,3))
       test5=abs(covx(1,1,i,j))+abs(covx(2,2,i,j))+abs(covx(3,3,i,j))
       if (atest1.gt.test4*1.0d-4.and.atest1.gt.1.0d-30) then
        check=lf
        if (lout)
```

```
     *  write(*,10)npos,i,j,test1,test4,covx(i,j,1,1),covx(i,j,2,2),
     *  covx(i,j,3,3)
   10   format(' warning ',i2,2i3,5d12.5)
        nwar=nwar+1
       end if
       if (atest2.gt.test5*1.0d-4.and.atest2.gt.1.0d-30) then
        check=lf
        if (lout)
     *  write(*,11)npos,i,j,test2,test5,covx(1,1,i,j),covx(2,2,i,j),
     *  covx(3,3,i,j),check
   11   format(' warning5 ',i2,2i3,5d12.5,l2)
        nwar=nwar+1
       end if
      end do
      end do
  810 checkc=check
      return
      end
C
      SUBROUTINE PRONLL(AN,INUL,H,NT,VAR,LRED,LBS,IANT,INULT,IHT)
C
C     THIS SUBROUTINE USES A CHOLESKY ALGORITHME FOR REDUCING
C     AND SOLVING THE SYSTEM OF LINEAR EQUATIONS
C                (AT*A)*X=AT*Y
C     WHERE (AT*A) IS SYMMETRICAL POSITIV DEFINITE MATRIX OF
C     DIMENSION NT*NT, AND (AT*Y) IS A VECTOR OF DIMENSION NT.
C
C     CONTEND OF ARRAYES:
C             AN(.)         THE UPPER PART OF (AT*A), AND RETURNS
C                           WITH LT, WHERE L*LT=(AT*A), IF LRED =
C                           .TRUE.
C             INUL(.)       INDEX OF THE FIRST NON-ZERO ELEMENT
C                           OF EACH ROW.
C             H(.)          THE RIGHT-HANDSIDE (AT*Y), AND RETURNS
C                           WITH X ,IF LBS = .TRUE., ELSE WITH
C                           (L-1)*(AT*Y).
C             VAR           THE PSEUDO DIAGONAL ELEMENT OF (L-1)*
C                           (AT*Y).
C
C
C     PROGRAMMED BY
C                     PER KNUDSEN
C                     GEODETIC INST.
C                     DK-2920                       12.07.85.
C
C     IMPLICIT REAL*8 (A-H,O-Z)
      implicit none
      integer iht,inult,nt,ir,irt,inul,iant,is,ist,ii,irr,iit
      real*8 an,sum,h,var
      DIMENSION AN(IANT),INUL(INULT),H(IHT)
      LOGICAL LRED,LBS
C
C***  THE UPPER PART OF A IS REDUCED INTO LT IF LRED IS TRUE.
C
      IF(.NOT.LRED) GO TO 50
      DO 25 IS=1,NT
      IST=(IS*(IS-1))/2
      SUM=0.0D0
      DO 10 IR=INUL(IS),(IS-1)
      IRT=(IR*(IR-1))/2
      SUM=0.0D0
      DO 5 II=MAX0(INUL(IS),INUL(IR)),(IR-1)
      SUM=SUM+(AN(IRT+II)*AN(IST+II))
    5 CONTINUE
      AN(IST+IR)=(AN(IST+IR)-SUM)/AN(IRT+IR)
   10 CONTINUE
      SUM=0.0D0
      DO 15 II=INUL(IS),(IS-1)
      SUM=SUM+AN(IST+II)**2
```



```
   15 CONTINUE
C     write(*,*)ist,ii,an(ist+ii),sum
      IF ((AN(IST+IS)-sum).LE.1.0d-10) THEN
      WRITE(*,102)IS
  102 FORMAT(' NUMERICAL SINGULARITY IN ROW ',I4)
      write(*,*)ist,ii,an(ist+ii),sum
      NT=IS-1
      AN(IST+IS)=0.0D0
      GO TO 50
      else
      AN(IST+IS)=SQRT(AN(IST+IS)-SUM)
      end if
C
   25 CONTINUE
C
C
   50 CONTINUE
C***  SOLVE L-1*H
C
      DO 100 IR=1,NT
      IRT=(IR*(IR-1))/2
      SUM=0.0D0
      DO 90 II=INUL(IR),(IR-1)
      SUM=SUM+(AN(IRT+II)*H(II))
   90 CONTINUE
      H(IR)=(H(IR)-SUM)/AN(IRT+IR)
  100 CONTINUE
      SUM=0.0D0
      DO 101 II=1,NT
      SUM=SUM+H(II)**2
  101 CONTINUE
      VAR=SUM
C
C***  THE SOLUTION IS FOUND BY BACK SUBSTITUTUION IF LBS IS TRUE.
C
      IF(.NOT.LBS) RETURN
C
      DO 150 IRR=1,NT
      IR=NT+1-IRR
      IRT=(IR*(IR-1))/2
      SUM=0.0D0
      DO 140 II=(IR+1),NT
      IIT=(II*(II-1))/2
      SUM=SUM+(AN(IIT+IR)*H(II))
  140 CONTINUE
      H(IR)=(H(IR)-SUM)/AN(IRT+IR)
  150 CONTINUE
C
      RETURN
      END
```