Back to Bennett

An Assessment of Uncertainties introduced into Cleared Lunar Distances by the use of Bennett's Table

Touching On:

- •JPL Ephemeris
- •FORTRAN
- •Coordinate Systems
- •USNO NOVAS
- •IAU SOFAS
- •LINUX vs. DOS
- •Simulating Tabular Precision
- •Monte Carlo Methods
- •Excel
- •Distributions and Probability

David A. Walden
June 2008

An In-Person "Posting"

There was a discussion on the Google NavList in April and May of 2008 of using Bennett's Table to clear Lunar Distances

Re: Lunars using Bennett	Rule_No.1...	May 23, 23:41 +0200	1340	g	5106
Re: Lunars using Bennett	frankreed@H...	May 24, 02:07 -0400	500	g	5111
Re: Lunars using Bennett	Rule_No.1...	May 24, 17:59 +0200	707	g	5119
Re: Lunars using Bennett	piterr11@g ...	Apr 11, 03:31 +1000	1595	g	4826
Re: Lunars using Bennett	wbnoyce@g	Apr 7, 09:34 -0400	1353	g	4811
Re: Lunars using Bennett	frankreed@H...	Apr 6, 20:21 -0400	1376	g	4806
Re: Lunars using Bennett	piterr11@g ...	Apr 6, 17:42 +1000	31414	g	4801
Re: Lunars using Bennett	frankreed@H...	Apr 6, 02:22 -0400	1055	g	4800
Re: Lunars using Bennett	frankreed@H...	Apr 6, 02:19 -0400	3339	g	4799
Re: Lunars using Bennett	eremenko@m ...	Apr 5, 13:11 -0400 (EDT)	668	g	4798
Re: Lunars using Bennett	piterr11@g ...	Apr 5, 18:33 +1100	1660	g	4797
Re: Lunars using Bennett	piterr11@g ...	Apr 5, 17:27 +1100	885	g	4796
Re: Lunars using Bennett	wbnoyce@g	Apr 4, 15:51 -0400	3112	g	4795
Re: Lunars using Bennett	eremenko@m ...	Apr 4, 13:57 -0400 (EDT)	1390	g	4794
Re: Lunars using Bennett	piterr11@g ...	Apr 5, 04:31 +1100	5880	g	4793
Re: Lunars using Bennett	george@h	Apr 4, 15:23 +0100	5377	g	4792
Re: Lunars using Bennett	eremenko@m ...	Apr 4, 10:00 -0400 (EDT)	1176	g	4791
Re: Lunars using Bennett	piterr11@g ...	Apr 4, 19:29 +1100	1596	g	4790
Re: Lunars using Bennett	waldendand@y...	Apr 3, 20:59 -0700 (PDT)	421	g	4789
Re: Lunars using Bennett	eremenko@m ...	Apr 3, 22:11 -0400 (EDT)	2021	g	4788
Lunars using Bennett	waldendand@y...	Apr 3, 14:20 -0700 (PDT)	21662	g	4787

Included, was this comment from George:

D.Walden can reach no conclusions about potential errors in using the Bennett tables for that purpose, until he has checked out enough predictions to see what the resulting scatter amounts to (say, 10; at least 4, anyway). His one-off "bull's eye" signifies no more than I would achieve if I scored a bull's eye at my first throw at a dartboard. It would be a lucky accident; no more than that. I predict that he will see an overall range of scatter of 3 to 4 minutes or so. That would render it pretty useless for lunar distances, the deduced longitudes covering a range of longitudes getting on for 2 degrees..

So come on, D Walden, spend a bit more time with your tables, and let's see what scatter you come up with.

I took the “jump through the hoop” demand as an intellectual challenge.

So, how to begin?

My thought: Automate the whole thing.

- Impartial (if programmed correctly).
- Fast (could do more calculations).
- Fun (more of a challenge, less drudgery)

What is required:

- Observations
- Computerized version of using Bennett’s Table
- Analysis of Results

Observations:

Just as easy to use “real” data

Need: Time, Location, Positions of Bodies

Without loss of generality, chose one location (can easily be changed or randomized)

Start now and step forward approximately 32 days at a time for some hundreds of steps

Need moon’s position

JPL DE406 Ephemeris is authoritative source of positions of Solar System bodies (If you’re going to land on Mars, you better know exactly where it is!) ((Used in preparing Nautical Almanac.))

JPL gives x, y, z in “sun fixed” coordinates. Need earth and moon positions to find relative.

JPL data is available on-line. JPL FORTRAN source available for reading data. (Be careful about MS vs. LINUX line termination conventions!)

Need coordinate transformations from x,y,z to RA/Dec to Alt/Az. Use USNO NOVAS FORTRAN source. Calculates precession, nutation, aberration, refraction, etc. (Same algorithms used for Nautical Almanac preparation.)

Need star positions. Use Hipparcus, a modern catalog with recent proper motions. (Movement of the stars.) ((NOVAS apply proper motions.))

With RA/Dec’s and Alt/Az, calculate separations.

Apply “filtering” rules.

Moon above minimum altitude

Star above minimum altitude

Separation (lunar distance) not too big or too small

Since we can calculate with and without refraction and with and without parallax (topocentric and geocentric), we now have everything we need to simulate an observation and to calculate the true lunar distance.

Write these results out to a text file (center to center).

RA Geoc RA Topt Happ Htrue PA

2008 4 8 15.000 -77.000 39.000

MOON	49.7389	50.5184	33.1233	33.0980	0.8457			
alAri(Hamal)	31.9065	23.5016	47.7499	47.7348	16.3382	17.0845	16	20.3
alTau(Aldebara)	69.0977	16.5268	15.0563	14.9959	19.4794	18.6614	19	28.8
beGem(Pollux)	116.4590	28.0077	-9.0087	-9.0087	59.4936	58.9726	59	29.6
alLeo(Regulus)	152.2087	11.9254	-37.6435	-37.6435	96.3718	95.8060	96	22.3
alVir(Spica)	201.4149	-11.2080	-47.5353	-47.5353	150.3605	149.8310	150	21.6
alSco(Antares)	247.4847	-26.4523	-21.9771	-21.9771	163.6778	164.2551	163	40.7
alAql(Altair)	297.7981	8.8862	38.0826	38.0615	106.0552	106.8738	106	3.3
alPsA(Fomalhaut)	344.5246	-29.5783	21.4602	21.4184	82.1416	82.4427	82	8.5
alPeg(Markab)	346.2897	15.2474	66.2368	66.2295	59.9705	60.7214	59	58.2

RA Dec Happ Htrue LDclear LDobs deg min

No refrac

Now, using the “observations” use Bennett’s Table to calculate cleared lunar distance. (For convenience, this is done in a separate FORTRAN program.)

Previous NavList discussions, including input from Mr. Bennett, and references from his book, provide details needed to reproduce the numbers in the Table.

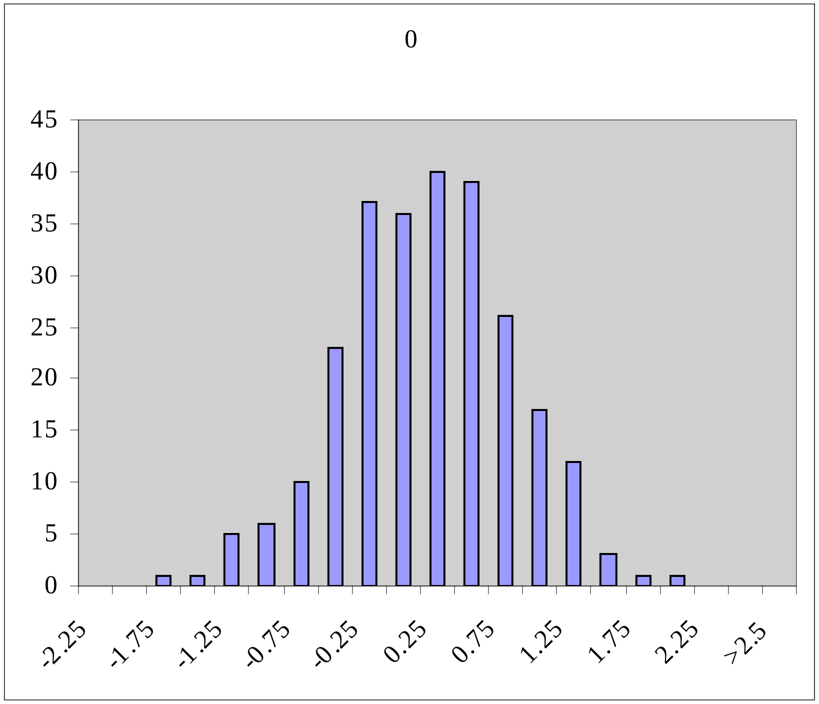
Some care is needed to round Table entries to the precision published. And likewise when going from table entry values to angles to use only the precision of the published table.

Using the procedure given first by Bennett and then independently by Walden, clear the lunar distance. Write out the cleared lunar distance via Bennett’s Table and portions of the input data.

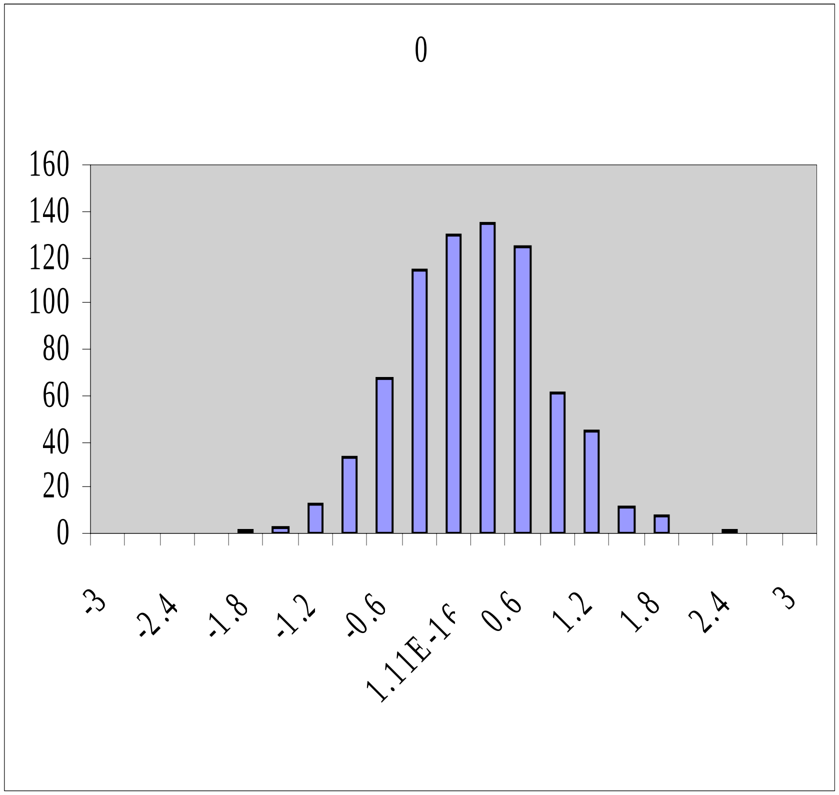
#	LD	LD Bennett	tdif	totsq	60*dif
1	16.35017	16.33333	1.01040	1.02091	1.01040
2	19.52332	19.51667	1.40980	1.18043	0.39940
3	82.17139	82.18333	0.69340	1.69365	-0.71640
4	59.98506	59.96667	1.79684	2.91123	1.10344
5	16.35017	16.33333	2.80724	3.93214	1.01040
6	19.52332	19.51667	3.20663	4.09166	0.39940
7	82.17139	82.18333	2.49023	4.60489	-0.71640
8	59.98506	59.96667	3.59367	5.82247	1.10344
9	35.26164	35.26667	3.29201	5.91347	-0.30167
10	81.07699	81.08334	2.91115	6.05852	-0.38086

Read this file into an EXCEL spread sheet for analysis and plotting.

For 258 simulations, the average error is 0.0426 minutes, the standard deviation is 0.6276 minutes, deviations range from 1.7610 to -1.8400. The distribution is shown below.



For 745 simulations, the average error is 0.0238 minutes, the standard deviation is 0.6235 minutes, deviations range from 2.1405 to -1.8400. The distribution is shown below.



Program Listings

```
              program lunarmain2
              implicit none
              character*16 snam
              integer locatn, icoord,j
              integer iyr,imo,ida,ihr,imin,idloop
              real*8 rsec,hr,delt,gst,ZD, AZ, RAR, DECR,tt,agst,d2r
              real*8 moonRAgeo,moondecgeo,moongha,moonratopo,moondectopo
              real*8 starRAgeo,stardecgeo,stargha,starratopo,stardectopo
              real*8 moonPA,moonaltw,moonaltwo,sep,staraltw,staraltwo
              real*8 tsep,fracd
              real*8 tjd,star(6),observ(6),skypos(7)
              open(unit=2,file="lunarstarsnova.dat")
              open(unit=3,file="lunarmain2.out")
              print*,'from lunarmain2'
              print*,'enter iyr,imo,ida,ihr,imin,rsec  '
              read(5,*)iyr,imo,ida,ihr,imin,rsec
              print901,iyr,imo,ida,ihr,imin,rsec
901           format('Echo of input:',i5,i3,i3,i3,i3,f12.6)
              hr=ihr+imin/60.d0+rsec/3600.d0
c             print*,'hr= ',hr
              call JULDAT (iyr,imo,ida,hr,tjd)
              print900,tjd
900           format('tjd= ',f20.9)
      call iau_DAT ( IYr, IMo, IDa, 0.d0, DELT, J )
              if(j.ne.0)stop 'iau_DAT error flag j NE 0'
              print*,'delt = ',delt
              delt=delt+32.1856d0
              idloop=0
100           continue
              if(idloop.gt.1)tjd=tjd+7.125
              if(idloop.gt.1000)stop "idloop gt 1000"
              tt=tjd+delt/60.d0/60.d0/24.d0
*     LOCATN=0 AND ICOORD=1 APPARENT PLACE
*     LOCATN=1 AND ICOORD=1 TOPOCENTRIC PLACE
              locatn=0
              icoord=1
c geocentric moon
      call PLACE ( Tt, "MOON", LOCATN, ICOORD, STAR, OBSERV,
     .                   SKYPOS )
c             print*,skypos
              call printhms(skypos(4))
              call printdms(15.d0*skypos(4))
              call printdms(skypos(5))
              moonRAgeo=15.d0*skypos(4)
              moonDecgeo=skypos(5)
```

```
c               tjd=2454565.125d0
c 1 means apparent GST
      call SIDTIM ( TJD, 0.d0, 1,   GST )
               print*,'App Gst'
               call printdms(15.d0*gst)
               agst=15.d0*gst
               call printdms(15*(gst-skypos(4)))
               moongha=15.d0*(gst-skypos(4))
               locatn=1
               icoord=1
*                   OBSERV(1) = GEODETIC (ITRS) LONGITUDE OF OBSERVER
*                               (EAST +) IN DEGREES (IN)
*                   OBSERV(2) = GEODETIC (ITRS) LATITUDE OF OBSERVER
*                               (NORTH +) IN DEGREES (IN)
*                   OBSERV(3) = HEIGHT OF OBSERVER ABOVE ELLIPSOID
*                               IN METERS (IN)
*                   OBSERV(4) = VALUE OF DELTA-T IN SECONDS (IN)
*                               (DELTA-T=TT-UT1)
               observ(1)=-77.d0
               observ(2)=39.d0
               observ(3)=0.d0
               observ(4)=delt
c topocentric
      call PLACE ( Tt, "MOON", LOCATN, ICOORD, STAR, OBSERV,
     .                   SKYPOS )
               print*,skypos
               print*,'topoc ra,dec'
               call printhms(skypos(4))
               call printdms(15.d0*skypos(4))
               call printdms(skypos(5))
               moonRAtopo=15.d0*skypos(4)
               moonDectopo=skypos(5)
               d2r=4.d0*datan(1.d0)/180.d0
               print*,moondecgeo,moondectopo,moonrageo,moonratopo
               moonpa=dacos(dsin(moondecgeo*d2r)*dsin(moondectopo*d2r)
     &  +dcos(moondecgeo*d2r)*dcos(moondectopo*d2r)
     &  *dcos(d2r*(moonrageo-moonratopo)))
               moonpa=moonpa/d2r
               print*,'moon PA= ',moonpa*60.d0
      call ZDAZ(tjd,0.d0,0.d0,observ(1),observ(2),
     &  0.d0,skypos(4),skypos(5),1,
     .                  ZD, AZ, RAR, DECR )
               print*,'alt with refrac'
               call printdms(90.d0-zd)
               moonaltw=90.d0-zd
      call ZDAZ(tjd,0.d0,0.d0,observ(1),observ(2),
     &  0.d0,skypos(4),skypos(5),0,
     .                  ZD, AZ, RAR, DECR )
               print*,'alt without refrac'
               call printdms(90.d0-zd)
               moonaltwo=90.d0-zd
               if(moonaltw.lt.10. .or. moonaltwo.lt.10.)go to 100
```

```
               idloop=idloop+1
      call iau_JD2CAL ( tjd, 0.d0, IYr, IMo, IDa, FracD, J )
c              if(idloop.eq.5)then
c              print*,tjd,iyr,imo,ida,fracd,j,int(24.d0*fracd),
c     &             60.d0*(24.d0*fracd-int(24.d0*fracd))
c              stop
c              endif
               hr=fracd*24.d0
               print902,iyr,imo,ida,hr,observ(1),observ(2)
902            format(i5,i3,i3,f7.3,f8.3,f7.3,a5,5f8.4)
               print903  ,"MOON",moonRAgeo,moonRAtopo,moonaltw,moonaltwo,moonPA
               write(3,902)iyr,imo,ida,hr,observ(1),observ(2)
               write(3,903)"MOON",moonRAgeo,moonRAtopo,moonaltw,moonaltwo,moonPA
               do 1 j=1,9
               read(2,*)star
               backspace(2)
               read(2,'(85x,a16)')snam
               print*,'snam',snam
c geocentric star
      call PLACE ( Tt, "*", LOCATN, ICOORD, STAR, OBSERV,
     .                    SKYPOS )
c              print*,skypos
               call printhms(skypos(4))
               call printdms(15.d0*skypos(4))
               call printdms(skypos(5))
               starRAgeo=15.d0*skypos(4)
               starDecgeo=skypos(5)
c topocentric
               locatn=1
               icoord=1
      call PLACE ( Tt, "*", LOCATN, ICOORD, STAR, OBSERV,
     .                    SKYPOS )
               starRAtopo=15.d0*skypos(4)
               starDectopo=skypos(5)
               tsep=dacos(dsin(moondecgeo*d2r)*dsin(stardecgeo*d2r)
     &  +dcos(moondecgeo*d2r)*dcos(stardecgeo*d2r)
     &  *dcos(d2r*(moonrageo-starrageo)))
               tsep=tsep/d2r
               sep=dacos(dsin(moondectopo*d2r)*dsin(stardectopo*d2r)
     &  +dcos(moondectopo*d2r)*dcos(stardectopo*d2r)
     &  *dcos(d2r*(moonratopo-starratopo)))
               sep=sep/d2r
      call ZDAZ(tjd,0.d0,0.d0,observ(1),observ(2),
     &  0.d0,skypos(4),skypos(5),1,
     .                   ZD, AZ, RAR, DECR )
               print*,'alt with refrac'
```

```
               call printdms(90.d0-zd)
               staraltw=90.d0-zd
      call ZDAZ(tjd,0.d0,0.d0,observ(1),observ(2),
     &  0.d0,skypos(4),skypos(5),0,
     .                 ZD, AZ, RAR, DECR )
               print*,'alt without refrac'
               call printdms(90.d0-zd)
               staraltwo=90.d0-zd
               print903,snam,15.d0*skypos(4),skypos(5),staraltw,staraltwo,tsep,sep
903            format(a17,6f9.4,i4,f9.1)
               write(3,903),snam,15.d0*skypos(4),skypos(5)
     &    ,staraltw,staraltwo,tsep,sep,int(tsep),60.d0*(-int(tsep)+tsep)
1              continue
               rewind 2
               go to 100
               end
               subroutine printhms(angle)
               implicit none
               real*8 angle,f
      INTEGER NDP
      CHARACTER*1 sign
      INTEGER IDMSF(4)
      F = 1.d0/24.d0
*  Scale then use days to h,m,s routine.
      CALL iau_D2TF (9,ANGLE*F, SIGN, IDMSF )
               print900,sign,idmsf(1),idmsf(2),idmsf(3)+idmsf(4)/1.d9
900            FORMAT(a2,i4,' hr',i3,' min ',f9.6,' sec')
               return
               end
               subroutine printdms(angle)
               implicit none
               real*8 angle,f
      INTEGER NDP
      CHARACTER*1 sign
      INTEGER IDMSF(4)
      F = 15D0/360.d0
*  Scale then use days to h,m,s routine.
      CALL iau_D2TF (9,ANGLE*F, SIGN, IDMSF )
               print900,sign,idmsf(1),idmsf(2),idmsf(3)+idmsf(4)/1.d9
     &  ,idmsf(1),idmsf(2)+(idmsf(3)+idmsf(4)/1.d9)/60.d0
     &  ,idmsf(1)+(idmsf(2)+(idmsf(3)+idmsf(4)/1.d9)/60.d0)/60.d0
900            FORMAT(a2,i4,' deg',i3,"' ",f9.6,'"',
     &   "   OR ",i4,' deg',f10.6,"'",
     &   "   OR ",f10.6, 'deg')
               return
               end
      SUBROUTINE iau_D2TF ( NDP, DAYS, SIGN, IHMSF )
```

```
        program bennettfortran
        implicit none
        character*17 snam
        integer ilha,ialt,itwid,ires,isum,idec,ilat
        integer iyr,imo,ida,j,icount
        real xlong,xlat,hr,xra,xdec,xalt1,xalt2,xpa,xd1,xd2
        real lha,d2r,ldapp,alt,moontrue,moonapp,startrue,starapp
        real twid,xinter,sum,dec,lat,z,res,ldclear,full
        real totdif,totsq
        print*,'from bennettfortan.f'
        totdif=0.
        totsq=0.
        icount=0
        d2r=4.*atan(1.)/180.
c try reading lunarmain2.out file from the NT machine
                open(unit=2,file="lunarmain2.out")

20      continue

                read(2,*)iyr,imo,ida,hr,xlong,xlat
                print*,iyr,imo,ida,hr,xlong,xlat
c moon stuff
                read(2,*)snam,xra,xdec,xalt1,xalt2,xpa
                print*,snam,xra,xdec,xalt1,xalt2,xpa

                moontrue=xalt2+xpa
                moonapp=xalt1
                print*,'moon apparent alt= ',xalt1,int(xalt1),60.*(xalt1-int(xalt1))
                print*,'correction= ',60.*(moontrue-moonapp)
                print*,'moon true alt= ',moontrue,int(moontrue),60.*(moontrue-
int(moontrue))
                print*,' '

c 9 lines of star stuff

        do 11 j=1,9
                read(2,*)snam,xra,xdec,xalt1,xalt2,xd1,xd2
                print*,snam,xra,xdec,xalt1,xalt2,xd1,xd2

                startrue=xalt2
                starapp=xalt1

                ldapp=xd2
```

```
c          ldapp=46+32./60.
c          moontrue=37.+29./60.
c          moonapp=36.+44./60.
c          startrue=45.+1./60.
c          starapp=45.+2./60.

           print*,'NOW CALC REAL'
           z=(cos(d2r*ldapp)-
sin(d2r*starapp)*sin(d2r*moonapp))/(cos(d2r*starapp)*cos(d2r*moonapp))
           print*,1/d2r*acos(z)

xinter=sin(d2r*startrue)*sin(d2r*moontrue)+cos(d2r*startrue)*cos(d2r*moontrue)*z
           xinter=1/d2r*acos(xinter)
           print*,xinter,int(xinter),60.*(xinter-int(xinter))
           full=xinter

           print*,'now start bennett'

           print*,ldapp,moontrue,moonapp,startrue,starapp

           ldapp=int(ldapp)+nint(60*(ldapp-int(ldapp)))/60.
           moontrue=int(moontrue)+nint(60*(moontrue-int(moontrue)))/60.
           moonapp=int(moonapp)+nint(60*(moonapp-int(moonapp)))/60.
           startrue=int(startrue)+nint(60*(startrue-int(startrue)))/60.
           starapp=int(starapp)+nint(60*(starapp-int(starapp)))/60.

           if(ldapp.lt.10.)go to 11
           if(moontrue.lt.10.)go to 11
           if(moonapp.lt.10.)go to 11
           if(startrue.lt.10.)go to 11
           if(starapp.lt.10.)go to 11
           if(xd1.gt.89.)go to 11
           if(xd2.gt.89.)go to 11
```



```
           print*,ldapp,moontrue,moonapp,startrue,starapp
```

```
print*,90.-ldapp
ALT=2*100000*1./2.*(1-cos(d2r*ldapp))
ialt=nint(alt)
print*,alt,ialt

twid=abs(-moonapp+starapp)
print*,'twid= ',twid
twid=2*100000*1./2.*(1-cos(d2r*twid))
itwid=nint(twid)
print*,twid,itwid

ires=ialt-itwid

xinter=1/d2r*acos(-ires/100000.+1)
print*,ires,xinter

xinter=int(xinter)+nint(60*(xinter-int(xinter)))/60.

sum=-13030*log10(1./2.*(1-cos(d2r*xinter)))
isum=nint(sum)
print*,sum,isum

dec=-13030*log10(cos(d2r*starapp))
idec=nint(dec)
print*,dec,idec

lat=-13030*log10(cos(d2r*moonapp))
ilat=nint(lat)
print*,lat,ilat

ilha=isum-ilat-idec
print*,ilha

z=1/d2r*acos(1.-2*10**(-ilha/13030.))
print*,z

lha=-13030*log10(1./2.*(1.-cos(d2r*z)))
ilha=nint(lha)
print*,lha,ilha
```

```
      lat=-13030*log10(cos(d2r*startrue))
      ilat=nint(lat)
      print*,lat,ilat

      dec=-13030*log10(cos(d2r*moontrue))
      idec=nint(dec)
      print*,dec,idec

      isum=ilha+ilat+idec
      z=1/d2r*acos(1.-2*10**(-isum/13030.))
      print*,isum,z

      z=int(z)+nint(60*(z-int(z)))/60.

      res=2*100000*1./2.*(1-cos(d2r*z))
      ires=nint(res)
      print*,res,ires

      twid=abs(startrue-moontrue)
      print*,twid
      twid=2*100000*1/2*(1-cos(d2r*abs(twid)))
      itwid=nint(twid)
      print*,twid,itwid

      ialt=ires+itwid
      alt=90.-1/d2r*acos(1-ialt/100000.)
      print*,ialt,alt

      ldclear=90.-alt
      print*,ldclear,int(ldclear),60.*(ldclear-int(ldclear))
      ldclear=int(ldclear)+nint(60*(ldclear-int(ldclear)))/60.
      print*,ldclear,int(ldclear),60.*(ldclear-int(ldclear))

      print*,'FINAL DIF in LDclear= ',60*(full-ldclear),' in minutes'
c     if(abs(60*(full-ldclear)).gt.20.)go to 11
      totdif=totdif+60*(full-ldclear)
      totsq=totsq+(60*(full-ldclear))**2
      icount=icount+1

      write(33,'(i6,5f12.5)')icount,full,ldclear,totdif,totsq,60*(full-ldclear)

11    continue
      print*,'icount,totdif,totsq= ',
      go to 20
      end
```