The Astronomical Almanac's algorithm for approximate solar position (1950–2050)

Article *in* Solar Energy · December 1988

DOI: 10.1016/0038-092X(88)90045-X

Joseph Michalsky
University of Colorado Boulder

Solar Energy Vol. 40, No. 3, pp. 227–235, 1988
Printed in the U.S.A.

0038–092X/88 $3.00 + .00

THE ASTRONOMICAL ALMANAC'S ALGORITHM FOR APPROXIMATE SOLAR POSITION (1950–2050)

JOSEPH J. MICHALSKY
Atmospheric Sciences Research Center, State University of New York, Albany, NY 12222, U.S.A.

Abstract—This article implements an algorithm for the calculation of solar position that has a stated, and partially demonstrated, accuracy of 0.01 deg until the year 2050. The algorithm is taken from *The Astronomical Almanac*, which has published it as an addendum to their very accurate tabulations since 1984. It uses the same approach as an earlier paper by Walraven, but has a more simplified form. This subject of the calculation of the solar position is revisited because of the somewhat confusing sequence of notes and letters that followed Walraven's publication in this journal and because of a report challenging the accuracy claimed by him. *The Almanac* approximation is compared to exact values and to some commonly used approximations. An appendix contains an annotated subroutine in FORTRAN, which should be easily converted to other programming languages.

1. INTRODUCTION

The solar literature does not have a shortage of papers on the subject of solar position calculation. The motivation for yet another one is proposed in the next paragraphs.

Many formulas have appeared in papers with only vague comments regarding their accuracy. Typically, a statement to the effect that the calculations are "adequate for most engineering applications" accompanies the presentation of same. Indeed, these formulas may be well within the accuracy requirements for some applications. As an example, flat-plate tracking photovoltaic panels do not require extremely accurate tracking. Even if the panels are tilted 5° with respect to the normal to the sun their output is within 0.4% of the maximum response to the direct component. The actual response may not improve even that much for perfect alignment since a significant portion of the radiation received by the panel can be skylight and reflected radiation, which may not be affected significantly by realignment. On the other hand, there are stringent tracking requirements for some applications. Solar thermal power systems with high concentration ratios require accurate tracking. Vant-Hull and Hildebrandt[1] used 1 mrad (0.06°) in their calculations, however, the more accurate the pointing, the better the efficiency up to the mechanical limitations of the tracking mechanism.

Walraven[2] published an algorithm almost a decade ago with a stated accuracy well within this limit. His algorithm differs from most in that it does not require the sun to have the same position at the same time each year. Rather, it is based on the sun's astronomical position with respect to a given point in time. Walraven derived simplified forms for the high precision calculations used in *The American Ephemeris and Nautical Almanac*[3]. The latter formulas are accurate to 0.1 arcseconds. His formulas have a stated accuracy of 0.01°.

Why then should there be interest in writing or even reading another solar position paper? There are two reasons. The primary one has to do with the response evoked by Walraven's paper. At least 10 notes and letters to *Solar Energy* followed its publication[4–13]. Some provided what can be considered useful changes, and others did not. Each time a user implements the solar position calculation, it is necessary to read 11 contributions to decide which changes are meaningful. My initial plan in addressing this subject was to send a brief note to *Solar Energy* with Walraven's subroutine slightly revised to incorporate the changes that I considered useful.

The second reason to address this subject was a report by Zimmerman[14]. In that report, Walraven's accuracy was disputed. The claim was that the accuracy was 0.06° in azimuth and 0.013° in elevation between 1979 and 1986. The implication was that it was less accurate outside this time frame. Since 1986 has passed, I considered submitting a note that would use Walraven's algorithm, but would base the approximation on a more current reference year.

In attempting to reproduce Walraven's results, I consulted a recent copy of *The Astronomical Almanac*[15] (hereafter, *The Almanac*). Beginning with the 1984 issue of that reference (this title replaced *The American Ephemeris and Nautical Almanac*[3] in 1981), I found that a set of formulas are published as an addendum to the very accurate tabulated data. These formulas have an accuracy of 0.01° for the century that began in 1950.

The Almanac's algorithm for the calculation of approximate solar position is presented in the next section. It closely resembles Walraven's[2], but has a simpler form. The coordinate systems' angular ranges differ slightly from those adopted by Walraven, but are more representative of the conventions used by astronomers, when referring to astronomical quantities, and by meteorologists, when referring to the local coordinates. A few ancilliary calculations are added for those who may want to use them. The third section contains comparisons with a few of the more commonly used expressions for solar position. These should allow one to decide whether certain approximations are adequate for their applications. In particular, Walraven's[2] algorithm is compared to *The Almanac* algorithm to test whether there is a need to implement the new algorithm for those currently using the one from Walraven. Comments summarizing the accuracy of the various algorithms are offered in the last section of the paper. An appendix completes the article. It contains a FORTRAN subroutine with ample commentary that should allow straightforward implementation in other languages.

2. *THE ALMANAC* ALGORITHM

One of the most difficult aspects in describing or understanding the movement of the sun is in understanding the various coordinate systems within which its position is specified. These are described verbally and graphically in Walraven's paper[2] and need not be repeated here.

2.1 *Time*

In *The Almanac* algorithm, the time argument is the difference in days between the current Julian date, jd, and JD 2451545.0, which is noon 1 January 2000 Universal Time (UT). The current Julian date may be specified directly or calculated from more familiar expressions of time using

$$\begin{aligned}\mathrm{jd} = 2432916.5 &+ \mathrm{delta} \times 365\\ &+ \mathrm{leap} + \mathrm{day} + \mathrm{hour}/24,\end{aligned}$$

where

$$\mathrm{delta} = \mathrm{year} - 1949,$$

and

$$\mathrm{leap} = \text{integer portion of } (\mathrm{delta}/4).$$

Day is the day of the year (e.g., Feb 1 = 32), and hour is the UT in hours including fractions thereof. The leading number in the formula for jd is the Julian date for midnight 0 January 1949 UT. The choice of 1949 is arbitrary, but it does allow one to work with positive numbers in the calculation of jd for the century in question. The details of specifying local time in terms of UT and day of the year are left to the reader to include in the code that calls the subroutine.

2.2 *Ecliptic coordinates*

The ecliptic coordinates are calculated from the time argument according to the following steps:

$$\begin{aligned}n &= \mathrm{jd} - 2451545.0\\ L \text{ (mean long.)} &= 280.460 + 0.9856474 \times n \; (0 \le L < 360^\circ)\\ g \text{ (mean anomaly)} &= 357.528 + 0.9856003 \times n \; (0 \le g < 360^\circ)\\ l \text{ (ecliptic long.)} &= L + 1.915 \times \sin(g) + 0.020 \times \sin(2 \times g) \qquad (0 \le \ell < 360^\circ)\\ \mathrm{ep} \text{ (obliquity of the ecliptic)} &= 23.439 - 0.0000004 \times n \text{ (degs)}\end{aligned}$$

Note that the specification ($0 \le L < 360^\circ$) implies that the quantity L is to have multiples of 360° added or subtracted until is has a value in the stated range.

2.3 *Celestial coordinates*

To calculate the right ascension and declination we use the following formulas:

$$\begin{aligned}\tan(\mathrm{ra}) &= \cos(\mathrm{ep}) \times \sin(l)/\cos(l)\\ \sin(\mathrm{dec}) &= \sin(\mathrm{ep}) \times \sin(l).\end{aligned}$$

The formula for the right ascension is written in this form (note that $\sin(l)/\cos(l) = \tan(l)$) because we wish to assign the right ascension to an angle between 0 and 360°. Having the signs of the numerator and denominator allows us to accomplish this. A two-argument inverse tangent function eases the task somewhat if one is available in the particular language of the subroutine. Although FORTRAN has this function, I have chosen to demonstrate how one derives the correct angle with a single argument inverse tangent function for use in other programming languages.

2.4 *Local coordinates*

In order to convert to azimuth and elevation, we need to know the declination and hour angle. In order to calculate the hour angle, we need to know the local mean sidereal time (lmst) and the right ascension. The lmst can be defined as the right ascension of the point on the meridian at the celestial equator. With this definition, the hour angle is found by subtracting the right ascension from the lmst. The Almanac tabulates Greenwich mean sidereal time (gmst) at midnight and the formula to approximate gmst at an arbitrary time is given by

$$\begin{aligned}\mathrm{gmst} = 6.697375 &+ 0.0657098242 \times n\\ &+ \mathrm{hour(UT)} \qquad (0 \le \mathrm{gmst} < 24 \text{ h}).\end{aligned}$$

The variable "hour" is not multiplied by a factor to convert it to sidereal hours because n already includes the fractional day and this correction is made in that product. The east longitude of the site in question is added to the gmst in order to specify lmst. Since longitude is usually expressed in degrees, the longitude should be divided by 15 to convert to hours. East longitude is considered positive. Therefore,

$$\begin{aligned}\mathrm{lmst} = \mathrm{gmst} &+ (\text{east long.})/15\\ &(0 \le \mathrm{lmst} < 24 \text{ h}), \qquad (2)\end{aligned}$$

and the hour angle (ha) is then given by

$$\mathrm{ha} = \mathrm{lmst} - \mathrm{ra}\ (-12 < \mathrm{ha} \leq 12\ \mathrm{h}). \qquad (3)$$

Before the sun reaches the meridian, its hour angle is negative, and its hour angle is positive in the post-meridian hemisphere. (Note that many solar researchers use the opposite sign convention for ha with the eastern hemisphere positive and western hemisphere negative.)

To calculate the elevation and azimuth, we use the following transformations:

$$\sin(\mathrm{el}) = \sin(\mathrm{dec}) \times \sin(\mathrm{lat}) + \cos(\mathrm{dec}) \times \cos(\mathrm{lat}) \times \cos(\mathrm{ha})\ (\mathrm{degs}) \qquad (4)$$

and

$$\sin(\mathrm{az}) = -\cos(\mathrm{dec}) \times \sin(\mathrm{ha})/\cos(\mathrm{el}) \quad (0 \leq \mathrm{az} < 360°) \qquad (5)$$

where lat is the latitude of the site and hour angle has been converted to a more conventional angular argument. Azimuth is measured from north, where it is 0°, through east to 360°. (Again, note that many in solar energy use the convention that due south is 0° azimuth and east is positive while west is negative azimuth.) In order to assign the correct quadrant, a little care must be taken. Walraven[2] showed that at an azimuth of 90°

$$\sin(\mathrm{el}) = \sin(\mathrm{dec})/\sin(\mathrm{lat}).$$

We will call it elc for el critical. To correctly assign azimuth we use the algorithm that

$$\text{if } \mathrm{el} \geq \mathrm{elc}, \quad \text{then} \quad \mathrm{az} = 180\ \mathrm{degs} - \mathrm{az},$$

and

$$\text{if } \mathrm{el} \leq \mathrm{elc} \quad \text{and} \quad \mathrm{ha} > 0, \quad \text{then} \quad \mathrm{az} = 360 + \mathrm{az}.$$

2.5 *Additional considerations*

The calculations above are for the center of the sun, which is about 1/2° in diameter and for a nonrefracting atmosphere. When observing near the horizon, refraction is important, and when calculating sunrise and sunset the angular extent of the sun needs to be taken into account, since the upper limb (or edge) of the solar disk is used as the reference point. *The Almanac*'s refraction correction is given by

$$r = 3.51561 \times (0.1594 + 0.0196 \times \mathrm{el} + 0.00002 \times \mathrm{el}\hat{\ }2)/(1 + 0.505 \times \mathrm{el} + 0.0845 \times \mathrm{el}\hat{\ }2),$$

where 3.51561 = 1013.2 mb/288.2 K. These are the atmospheric pressure and temperature for the U.S. Standard Atmosphere[16]. To calculate the diameter of the sun at any time, we need to know its distance from the earth. In terms of the astronomical unit, which is the mean sun-earth distance and assigned a value 1, the distance varies according to

$$R = 1.00014 - 0.01671 \times \cos(g) - 0.00014 \times \cos(2 \times g).$$

The radius is then calculated from

$$\text{solar radius} = 0.2666/R.$$

These corrections apply to only one particular structure of the atmosphere, which it seldom, if ever, assumes exactly. It is impractical to expect an accuracy of 0.01° when observing at low solar elevations because of this uncertainty.

There has been no mention of the equation of time in this discussion. The equation for ecliptic longitude (1) accounts for the differences in the length of the solar day that gives rise to this equation. For completeness, the equation of time can be calculated from the previously derived variables using

$$\mathrm{EOT} = L(\text{mean longitude}) - \mathrm{ra}(\text{right ascension}).$$

Frequently, solar researchers wish to know the solar position in order to find the sun's angle of incidence on an arbitrarily oriented flat surface. Rewriting Iqbal's[18] eqn 1.6.5a, we use the calculated variables to define

$$\cos(\mathrm{aoi}) = \sin(\mathrm{el}) \times \cos(\mathrm{slope}) + \cos(\mathrm{el}) \times \sin(\mathrm{slope}) \times \cos(\mathrm{ra} - \mathrm{saz}),$$

where "slope" is the angle that the surface is tilted from the horizontal, "saz" is the surface azimuth defined similarly to local azimuth, and "aoi" is the angle of incidence with respect to the surface normal.

3. COMPARISONS OF APPROXIMATIONS

3.1 *True Almanac values vs Almanac approximations*

We cannot check our calculations using azimuth and elevation since these quantities are not tabulated in *The Almanac*. The quantities needed to calculate azimuth and elevation are the hour angle, the declination, and the latitude (eqns (4) and (5)). Hour angle is calculated from right ascension and local mean sidereal time (eqn (3)). Local sidereal time is obtained from the longitude and the Greenwich mean sidereal time (eqn (2)). The variables in the calculation of azimuth and elevation for a fixed location then are the right ascension, the declination, and the Greenwich mean sidereal time. These are tabulated in *The Almanac* for 0 h UT for each day of the year. To check the accuracy of the approximations, we will compare the tabulated values in *The Almanac* to the nearest 0.1 s of time or nearest arcsecond of angle. These comparisons will be for 1986.

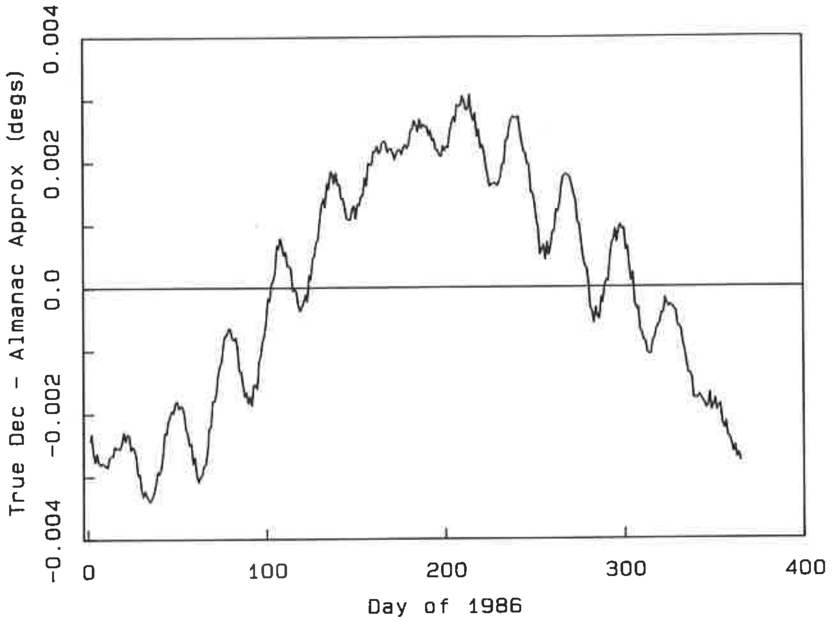

Fig. 1. The tabulated *Almanac* declination minus *The Almanac*'s approximate declination as a function of the day of 1986.

In Fig. 1 the difference of the declination tabulated in *The Almanac* and the declination calculated from *The Almanac*'s lower precision algorithm is plotted as a function of the day of the year in 1986. The true *Almanac* declination is recorded to the nearest arcsecond, which could, at most, affect the difference by about 0.0001°, about the size of the high frequency noise in this plot. The approximation is well within 0.01° of the true declination. In Fig. 2 the same kind of comparison is made for right ascension. In this case, the data are in hours since that is the normal unit for right ascension. In this plot, the true *Almanac* right ascension is rounded to the nearest 0.1 s. To convert to degs, multiply by 15. The maximum error is about 0.009°. The error is not symmetric about zero as it would be if the approximation were determined using a least squares procedure. In any case, the error is better than the 0.01° accuracy claimed for *The Almanac* approximation. In Fig. 3, true Greenwich mean sidereal time to the nearest 0.1 s is differenced with the approximated value in hours. Although the noise appears greater in

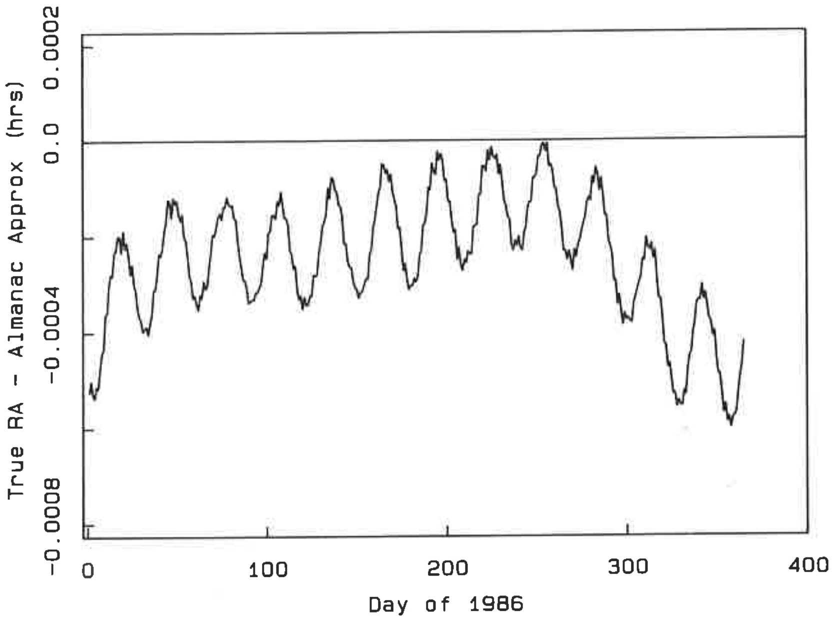

Fig. 2. The tabulated *Almanac* right ascension minus the *Almanac*'s approximate right ascension as a function of the day of 1986.

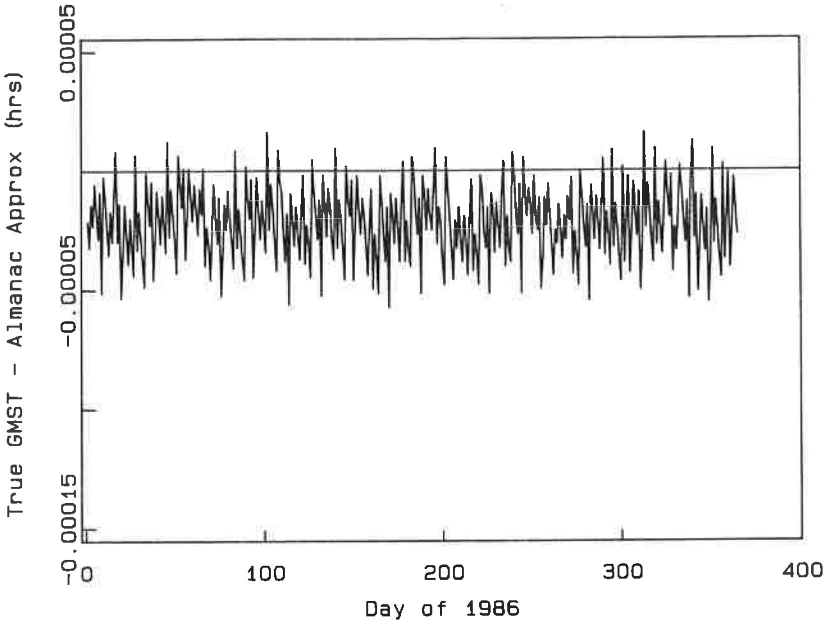

Fig. 3. The tabulated *Almanac* Greenwich mean sidereal time minus *The Almanac*'s approximate Greenwich mean sidereal time as a function of the day of 1986.

this plot, the maximum error is about 1/10th that in the previous plot and, therefore, well within the targeted angular accuracy.

3.2 *True Almanac values vs some common approximations*

In the rest of this section, we compare declination only. This is an arbitrary choice, and any of the other variables could be used to economize on the discussion. One of the more common ways to represent the solar position is through the use of trigonometric series. Spencer[17] represented the solar declination, sun-earth distance, and equation of time with truncated Fourier series. Iqbal[18] suggests these formulas for those interested in a reasonable yet simple approximation. Figure 4 is a plot of the true declination minus the Spencer approximation for each day of 1986. *The Almanac* approximation differenced with the true declination is also plotted on this scale. *The Almanac* approximate formula for declination quite clearly is a more accurate representation of the declination.

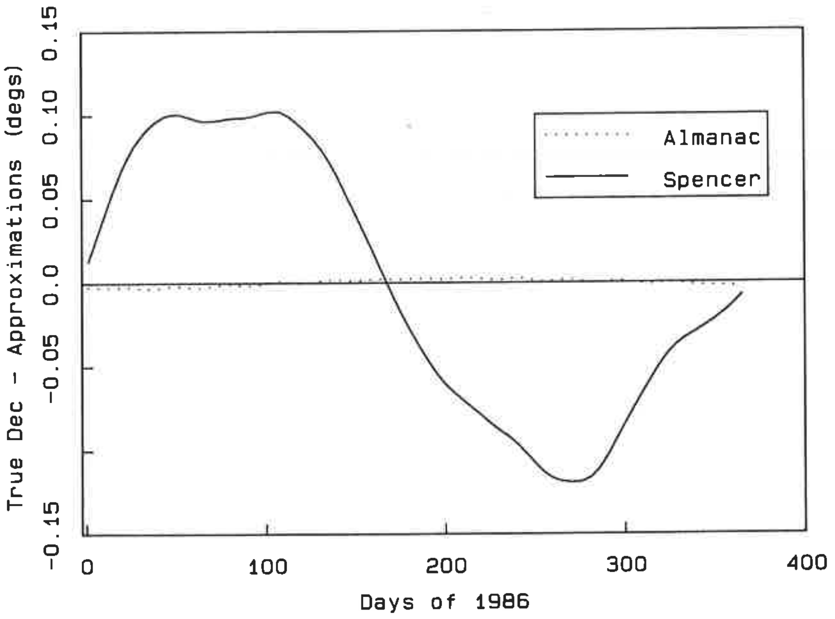

Fig. 4. The Spencer formula for declination and *The Almanac* formula for declination compared with the tabulated declination for 1986.

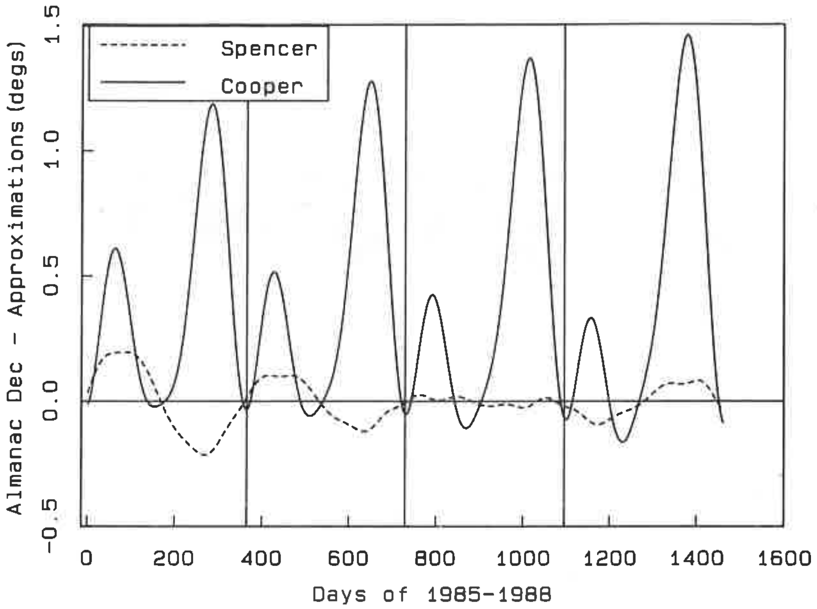

Fig. 5. The Cooper formula for declination and the Spencer formula for declination compared to *The Almanac* approximation for the years 1985–1988. The vertical lines are drawn through the last day of the year.

Perhaps, even more frequently the approximation of declination is made using the formula of Cooper[19]. This formula is a sine wave approximation of the declination. In Fig. 5 *The Almanac* approximation for declination is taken as the standard against which the Spencer[17] and Cooper[19] formulas are compared. Four consecutive years are compared to show how the uncertainties vary over the leap period. The differences will repeat in about the same pattern every four years with only a slight shift. Clearly, the Cooper formula is the least accurate of the three we have discussed. On the other hand, it is quite adequate if we only need to position to within about 2°. The Spencer formula can be quite accurate if it is used to approximate a particular year's declination, which is the way it was derived originally (note the third year). If it is based on four years, the maximum inaccuracy can be reduced as shown in Fig. 6, but there is little to be gained with this slight change. Therefore, the Fourier series coefficients given by Spencer are not amended here.

3.3 *Walraven vs Almanac approximation*

A question raised in the Zimmerman review[14] was the accuracy of the Walraven[2] formulas. In Fig. 7, the differences of two approximations from the

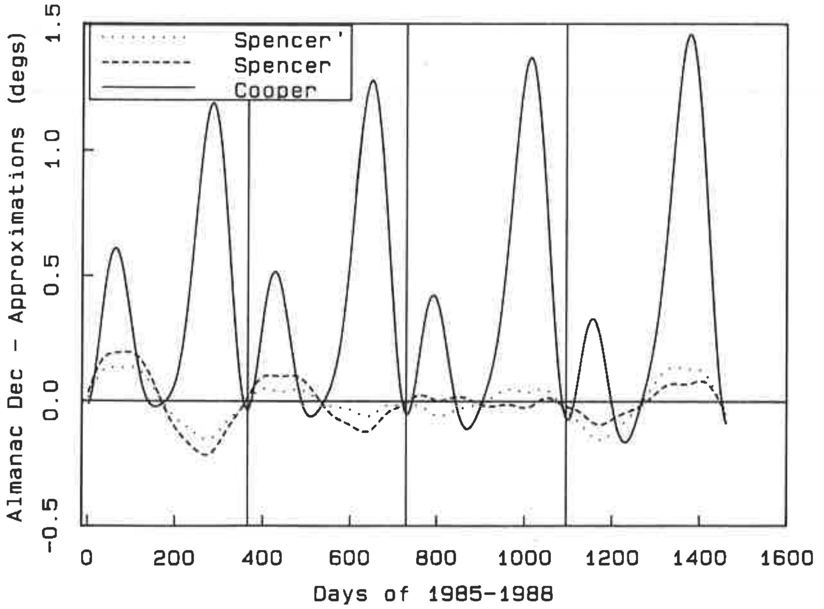

Fig. 6. The same as Fig. 5 except the Spencer-type approximation is rederived based on four years of declination values.

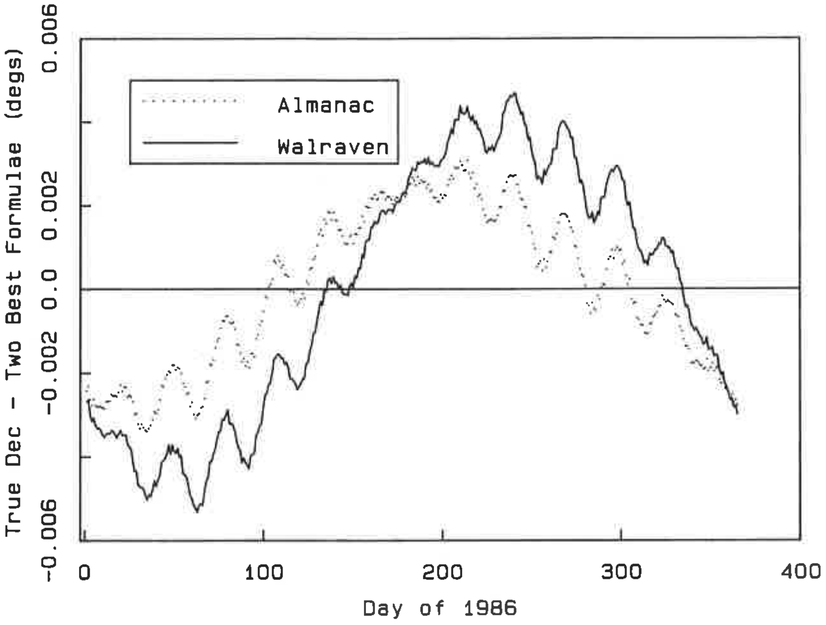

Fig. 7. The tabulated *Almanac* declination minus *The Almanac* formula and the Walraven formula for days in 1986.

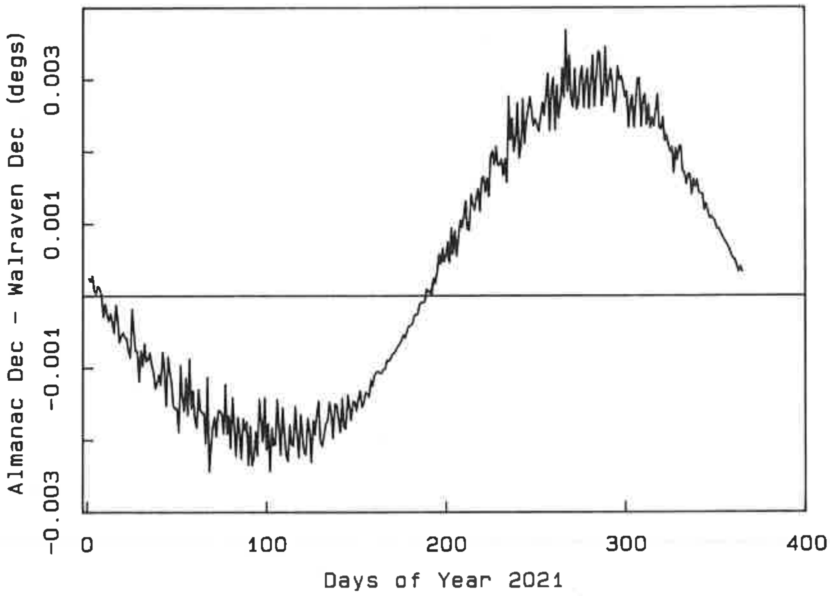

Fig. 8. The difference in the *Almanac* and Walraven approximation declination in 2021.

tabulated declination for 1986 are plotted. As we can see, *The Almanac* approximation performs slightly better than the Walraven approximation for that year. However, the Walraven approximation is well within the accuracy goal of 0.01°. In Fig. 8, we have calculated the difference in the declination between *The Almanac* and Walraven approximations for the year 2021. The range of differences is only slightly worse than that indicated in Fig. 7.

4. CONCLUSIONS

This article is intended to promote the use of a very accurate algorithm for the calculation of solar position. The article also compares commonly used approximations. The comparisons made in the last section have not been exhaustive. We have chosen to compare the declination in most instances. However, this should be representative of the other variables that are calculated with *The Almanac* algorithm. The examples were selected arbitrarily and not with any intent to amplify the errors in competing algorithms.

Cooper's formula for declination is frequently used because it was included in Duffie and Beckman's[20] book *Solar Energy Thermal Processes*. It is clearly the least accurate of the approximations discussed, but is adequate for many applications. The accuracy of Spencer's formulas vary with year in the leap cycle. They were derived using only a single year's declination as input for the least squares fit. They can be improved in the sense that the maximum deviation is reduced by fitting to four years of data as demonstrated in Fig. 6. Another possibility is to base the coefficients on a single years' data, and then add 1/4 day to the angular argument in each of the next three succeeding years and then repeat the pattern. This is somewhat awkward, but keeps the formulae simple while reducing the error.

The comparisons given in Figs. 7 and 8 would indicate very little reason to change to the present algorithm if one has already implemented the Walraven code. The comparison in Fig. 8 is the least satisfactory one in the article in that it is not made to true positions. These were not available for 2021 since we do not have *The Almanac*'s high precision routines and *The Almanac* is published for the following year only. Assuming that *The Almanac*'s claims for the low precision formulas are correct, however, the Walraven algorithm appears good for several years to come.

In conclusion, *The Almanac* algorithm is the best of the simple approximate formulas for calculating solar position. It is superior to the calculations based on least squares fits to a given set of data since it is based on the long-term progression of the sun in the ecliptic plane. It is somewhat simpler than the algorithm of Walraven, yet is accurate to 0.01° until the year 2050.

REFERENCES

1. L. L. Vant-Hull and A. F. Hildebrandt, Solar thermal power system based on optical transmission. *Solar Energy* **18,** 31–39 (1976).
2. R. Walraven, Calculating the position of the sun. *Solar Energy* **20,** 393–397 (1978).
3. *The American Ephemeris and Nautical Almanac,* 1980 Edition, U.S. Gov't. Printing Office, Washington, DC (1979).
4. R. Walraven, Erratum. *Solar Energy* **22,** 195 (1979).
5. C. B. Archer, Comments on "Calculating the position of the sun." *Solar Energy* **25,** 91 (1980).
6. B. J. Wilkinson, An improved FORTRAN program for the rapid calculation of the solar position. *Solar Energy* **27,** 67–68 (1981).
7. L. R. Muir, Comments on "The effect of atmospheric refraction in the solar azimuth." *Solar Energy* **30,** 295 (1983).
8. B. J. Wilkinson, The effect of atmospheric refraction on the solar azimuth. *Solar Energy* **30,** 295 (1983).
9. M. Ilyas, Solar position programs: Refraction, sidereal time and sunset/sunrise. *Solar Energy* **31,** 437–438 (1983).
10. D. J. B. Pascoe, Comments on "Solar position programs: Refraction, sidereal time and sunrise/sunset." *Solar Energy* **34,** 205–206 (1984).
11. M. Ilyas, Reply to comments by Dr. D. J. B. Pascoe. *Solar Energy* **34,** 206 (1984).
12. B. J. Wilkinson, Re: "Solar position programs: refraction, sidereal time and sunset/sunrise." *Solar Energy* **33,** 383 (1984).
13. M. Ilyas, Reply to B. J. Wilkinson's letter on "Solar position programs: refraction, sidereal time and sunset/sunrise." *Solar Energy* **33,** 383 (1984).
14. J. C. Zimmerman, Sun-pointing programs and their accuracy. SAND81-0761. Sandia National Laboratories, Albuquerque, NM (1981).
15. *The Astronomical Almanac,* 1986 Edition, U.S. Gov't. Printing Office, Washington, DC (1985).
16. *U.S. Standard Atmosphere,* U.S. Gov't. Printing Office, Washington, DC (1962).
17. J. W. Spencer, Fourier series representation of the position of the sun. *Search* **2,** 172 (1971).
18. M. Iqbal, *An Introduction to Solar Radiation.* Academic Press, Toronto (1983).
19. P. I. Cooper, The absorption of solar radiation in solar stills. *Solar Energy* **12,** 333–346 (1969).
20. J. A. Duffie and W. A. Beckman, *Solar Energy Thermal Processes.* John Wiley & Sons, New York (1974).

APPENDIX

```
c    subroutine sunae (year, day, hour, lat, long, az, el)
c
c    This subroutine calculates the local azimuth and ele-
c      vation of the sun at a specific location and time using
c      an approximation to equations used to generate ta-
c      bles in The Astronomical Almanac. Refraction cor-
c      rection is added so sun position is the apparent one.
c
c    The Astronomical Almanac, U.S. Gov't Printing Of-
c      fice, Washington, DC (1985)
c
c    input parameters
c      year = year (e.g., 1986)
c      day = day of year (e.g., Feb. 1 = 32)
c      hour = hours plus fraction in UT (e.g., 8:30 A.M.
c        eastern daylight time is equal to 8.5 + 5 (5 hours
c        west of Greenwich) - 1 (for daylight savings time
c        correction)
c      lat = latitude in degrees (north is positive)
c      long = longitude in degrees (east is positive)
c
c    output parameters
c      a = sum azimuth angle (measured east from north,
c        0 to 360°)
c      e = sun elevation angle (degs) plus others, but note
c        the units indicated before return

       subroutine sunae (year, day, hour, lat, long, az, el)
c    work with real variables and define some constants, in-
c      cluding one to change between degs and radians
       implicit real (a - z)
       data  twopi,  pi,  rad/6.2831853,  3.1415927,
     .017453293/
c
c    get the current julian date (actually add 2,400,000 for
c     jd)
       delta = year - 1949.
       leap = aint(delta/4.)
       jd = 32916.5 + delta*365. + leap + day + hour/
         24.
c    1st no. is mid. 0 jan 1949 minus 2.4e6; leap = leap
c      days since 1949
c
c    calculate ecliptic coordinates
       time = jd - 51545.0
c    51545.0 + 2.4e6 = noon 1 jan 2000
c
c    force mean longitude between 0 and 360 degs
```

```
      mnlong = 280.460 + .9856474*time
      mnlong = mod(mnlong, 360.)
      if(mnlong.lt.0.)mnlong = mnlong + 360.

c   mean anomaly in radians between 0 and 2*pi
      mnanom = 357.528 + .9856003 * time
      mnanom = mod(mnanom, 360.)
      if(mnanom.lt.0.)mnanom = mnanom + 360.
      mnanom = mnanom*rad
c
c   compute ecliptic longitude and obliquity of ecliptic in
c     radians
      eclong = mnlong + 1.915*(mnanom) + 0.20*
        sin(2.*mnanom)
    1   eclong = mod(eciong.360.)
      if(eclong.lt.0.)eclong = eclong + 360.
      oblqec = 23.429 - .0000004*time
      eclong = eclong*rad
      oblqec = oblqec*rad
c
c   calculate right ascension and declination
      num = cos(oblqec)*sin(eclong)
      den = cos(eclong)
      ra = atan(num/den)
c   force ra between 0 and 2*pi
      if(den.lt.0)then
        ra = ra + pi
      elseif(num.lt.0)then
        ra = ra + twopi
      endif
c
c   dec in radians
      dec = asin(sin(oblqec)*sin(eclong))
c
c   calculate Greenwich mean sidereal time in hours
      gmst = 6.697375 + .0657098242*time + hour
c   hour not changed to sidereal since 'time' includes the
c     fractional day
      gmst = mod(gmst, 24.)
      if(gmst.lt.0.)gmst = gmst + 24.
c
c   calculate local mean sidereal time in radians
      lmst = gmst + long/15.
      imst = mod(lmst. 24.)
      if(lmst.lt.0.)lmst = lmst + 24.
      lmst = lmst*15.*rad
c
c   calculate hour angle in radians between -pi and pi
      ha = lmst - ra
      if(ha.lt. - pi)ha = ha + twopi
      if(ha.gt.pi)ha = ha - twopi
c
c   change latitude to radians
      lat = lat*rad
c
c   calculate azimuth and elevation
      el = asin(sin(dec)*sin(lat) +
    cos(dec)*cos(lat)*cos(ha))
      az = asin(-cos(dec)*sin(ha)/cos(el))
c
c   if az = 90 degs. elcritical = asin(sin(dec)/sin(lat))
      elc = asin(sin(dec)/sin(lat))
      if(el.ge.elc)az = pi - az
      if(el.le.elc.and.ha.gt.0.)az = twopi + az
c   this puts azimuth between 0 and 2*pi radians
c
c   calculate refraction correction for US stan. atmosphere
c   need to have el in degs before calculating correction
      el = el/rad
      if(el.gt. - .56)then
        refrac = 3.51561*(.1594 + .0196*el +
    .00002*el**2)/
     1    (1. + .505*el + .0845*el**2)
      else
        refrac = .56
      endif
c   note that 3.51561 = 1013.2 mb/288.2 C
      el = el + refrac
c   elevation in degs
c
c   convert az and lat to degs before returning
      az = az/rad
      lat = lat/rad
c   mnlong in degs, gmst in hours, jd in days if 2.4e6 added;
c   mnanom, eclong, oblqec, ra, dec, lmst, and ha in ra-
      dians
      return
      end
```

```
c   this program calculates the solar position given the year,
day,
c   time, latitude, and longitude
c
c   it outputs solar azimuth, elevation, hour angle,
declination, and
c   air mass
c
c
        program sunpos
        real lat,long, mn
        write(6,'(3x,"This program calculates sun position if you
     # specify time and location!")')
        write(6,'(/)')
        write(6,'(3x,"Type the latitude like this sxx.xx where s
is a
     # sign!")')
        read(5,'(f8.4)')lat
        write(6,'(f8.4)')lat
        write(6,'(3x,"Type the longitude like this sxxx.xx!")')
        read(5,'(f9.4)')long
        write(6,'(f9.4)')long
        write(6,'(3x,"Type the year like this xxxx.!")')
        read(5,'(f5.0)')year
        write(6,'(f5.0)')year
        write(6,'(3x,"Indicate time zone,e.g., est=+5,mst=+7,etc.
(The
     # sign is important)!")')
        read(5,'(i3)')itz
        write(6,'(i3)')itz
        tz=float(itz)
60      write(6,'(3x,"Type the day of year(Feb 1=32) like this
xxx.!")')
        read(5,'(f4.0)')day
        write(6,'(f4.0)')day
50      write(6,'(3x,"Type the local standard time(not daylight
savings
     # time) like this xx.xx.xx., e.g., 12.15.47.!")')
        read(5,'(3f3.0)')hour, mn,sec
        write(6,'(3f3.0)')hour, mn,sec
        tm=((tz+hour)*3600+ mn*60+sec)/3600.
        call sunae(year,day,tm,lat,long,az,el,ha,dec,soldst)
        write(6,'(//)')
        write(6,'("The solar azimuth and elevation are")')
        write(6,100)az,el
100     format(3x,f9.4,3x,f8.4,/)
        write(6,'("The solar hour angle and declination are")')
        write(6,100)ha,dec
        am=armass(el)
        write(6,'("The air mass is")')
        write(6,200)am
200      format(3x,f6.2)
      write(6,'("The solar distance is")')
      write(6,250)soldst
250      format(3x,f7.4)
        write(6,'(//)')
        write(6,'("If you want another time same day, type 9!")')
        read(5,'(i1)')i
        if(i.eq.9)go to 50
        write(6,'("If you want another day, type 99!")')
        read(5,'(i2)')i
        if(i.eq.99)go to 60
        end
c-----------------------------------------------------------------
------+
c
|
c   subroutine sunae(year,day,hour,lat,long,az,el,ha,dec,soldst)
|
c
|
c   this subroutine calculates the local azimuth and elevation
of the |
c     sun at a specific location and time using an approximation
to   |
c     equations used to generate tables in The Astronomical
Almanac.  |
c     refraction correction is added so sun position is apparent
one. |
c
|
c   The Astronomical Almanac, U.S. Gov't Printing Office,
Washington, |
c     D.C. (1985).
|
c
|
c   input parameters
|
c     year=year, e.g., 1986
|
c     day=day of year, e.g., feb 1=32
|
c     hour=hours plus fraction in UT, e.g., 8:30 am eastern
```

```
daylight  |
c       time is equal to 8.5 + 5(5 hours west of Greenwich)
-1(for    |
c       daylight savings time correction)
|
c     lat=latitude in degrees (north is positive)
|
c     long=longitude in degrees (east is positive)
|
c
|
c   output parameters
|
c     a=sun azimuth angle (measured east from north, 0 to 360
degs)   |
c     e=sun elevation angle (degs)
|
c     plus others, but note the units indicated before return
|
c
|
c----------------------------------------------------------------
------+
c
      subroutine
sunae(year,day,hour,lat,long,az,el,ha,dec,soldst)
c  work with real variables and define some constants, including
c  one to change between degs and radians
      implicit real (a-z)
      data twopi,pi,rad/6.2831853,3.1415927,.017453293/
c
c   get the current julian date (actually add 2,400,000 for jd)
      delta=year-1949.
      leap=aint(delta/4.)
      jd=32916.5+delta*365.+leap+day+hour/24.
c   1st no. is mid. 0 jan 1949 minus 2.4e6; leap=leap days since
1949
c  the last yr of century is not leap yr unless divisible by 400
      if(amod(year,100.).eq.0.0.and.amod(year,400.).ne.
0.0)jd=jd-1.
c
c   calculate ecliptic coordinates
      time=jd-51545.0
c   51545.0 + 2.4e6 = noon 1 jan 2000
c
c   force mean longitude between 0 and 360 degs
      mnlong=280.460+.9856474*time
      mnlong=mod(mnlong,360.)
      if(mnlong.lt.0.)mnlong=mnlong+360.
c
c   mean anomaly in radians between 0 and 2*pi
      mnanom=357.528+.9856003*time
      mnanom=mod(mnanom,360.)
      if(mnanom.lt.0.)mnanom=mnanom+360.
      mnanom=mnanom*rad
c
c   compute the ecliptic longitude and obliquity of ecliptic in
radians
      eclong=mnlong+1.915*sin(mnanom)+.020*sin(2.*mnanom)
      eclong=mod(eclong,360.)
      if(eclong.lt.0.)eclong=eclong+360.
      oblqec=23.439-.0000004*time
      eclong=eclong*rad
      oblqec=oblqec*rad
c
c   calculate right ascension and declination
      num=cos(oblqec)*sin(eclong)
      den=cos(eclong)
      ra=atan(num/den)
c   force ra between 0 and 2*pi
      if(den.lt.0)then
          ra=ra+pi
      elseif(num.lt.0)then
          ra=ra+twopi
      endif
c
c   dec in radians
      dec=asin(sin(oblqec)*sin(eclong))
c
c   calculate Greenwich mean sidereal time in hours
      gmst=6.697375+.0657098242*time+hour
c   hour not changed to sidereal time since 'time' includes
c   the fractional day
      gmst=mod(gmst,24.)
      if(gmst.lt.0.)gmst=gmst+24.
c
c   calculate local mean sidereal time in radians
      lmst=gmst+long/15.
      lmst=mod(lmst,24.)
      if(lmst.lt.0.)lmst=lmst+24.
      lmst=lmst*15.*rad
c
c   calculate hour angle in radians between -pi and pi
      ha=lmst-ra
      if(ha.lt.-pi)ha=ha+twopi
      if(ha.gt.pi)ha=ha-twopi
c
c   change latitude to radians
      lat=lat*rad
c
c   calculate azimuth and elevation
      el=asin(sin(dec)*sin(lat)+cos(dec)*cos(lat)*cos(ha))
      az=asin(-cos(dec)*sin(ha)/cos(el))
c
c   this puts azimuth between 0 and 2*pi radians
      if(sin(dec)-sin(el)*sin(lat).ge.0.)then
      if(sin(az).lt.0.)az=az+twopi
      else
      az=pi-az
      endif
c
c   calculate refraction correction for US stan. atmosphere
c   need to have el in degs before calculating correction
      el=el/rad
c
      if(el.ge.19.225) then
         refrac=.00452*3.51823/tan(el*rad)
      else if (el.gt.-.766.and.el.lt.19.225) then
         refrac=3.51823*(.1594+.0196*el+.00002*el**2)/
     1   (1.+.505*el+.0845*el**2)
      else if (el.le.-.766) then
         refrac=0.0
      end if
c
c   note that 3.51823=1013.25 mb/288 C
      el=el+refrac
c   elevation in degs
c
c   calculate distance to sun in A.U. & diameter in degs
      soldst=1.00014-.01671*cos(mnanom)-.00014*cos(2.*mnanom)
      soldia=.5332/soldst
c
c   convert az and lat to degs before returning
      az=az/rad
      lat=lat/rad
         ha=ha/rad
         dec=dec/rad
c   mnlong in degs, gmst in hours, jd in days if 2.4e6 added;
c   mnanom,eclong,oblqec,ra,and lmst in radians
      return
      end
```

```
c
c
c   this function calculates air mass using kasten's
c   approximation to bemporad's tables
c
c
        function armass(el)
        z=(90.-el)*3.141592654/180.
        armass=1./(cos(z)+.50572*(6.07995+el)**-1.6364)
        return
        end
```