A Fortran program to calculate sunrise and sunset

Nicholas Moe

23 April 2007

1 Introduction

This is a description of a program written in Fortran 90/95 to calculate the rise and set ties of the sun, accurate to within a couple of minutes. It is based on pseudocode given in *Practical astronomy with your calculator* by Peter Duffett-Smith. The calculation relies on extrapolations on conditions present at a 0.0 January 1980 epoch. After the extrapolations are made, various conversions are necessary to calculate the times of sunrise and sunset. Corrections for parallax and atmospheric distortion are also made. The basic procedure is as follows:

1. Calculate the ecliptic coordinates of the sun on the date in question.
2. Convert the ecpliptic coordinates to declination and right ascension coordinates.
3. Calculate the ecliptic coordinates of the sun 24 hours later and convert the position to equatorial coordinates.
4. Compute the times of rising and setting for the coordinates found in terms of local sidereal time.
5. Interpolate between these times in order to find a more accurate rise and set time.
6. Correct these times for parallax and atmospheric distortion.
7. Convert these corrected times to local civil time.

Some unfamiliar terms used in this brief outline will be explained in the sections that follow.

1.1 Julian dates

For some of the calculations for this program, it is necessary to count the number of days between the day in question and either the beginning of the year or the beginning of the epoch. The method that is preferred by the book's procedures is easier for a human to do than a computer. The book suggests counting the days between the beginning of the current year and the current date by adding up the number of days in each month preceeding the current month, and then adding the day number.

In order to do this, one has to consider whether the current year is a leap year in order to determine how many days to add during February. If the number of days since the beginning of the epoch is required, then one would have to execute the above procedure and then add up the number of days in each year including the epoch and preceeding the current year, taking into account whether the year is a leap year or not.

I have found it easier to do these day-difference calculations by computing Julian day numbers. The Julian date is the number of days that have passed since "Greenwich mean noon of January 1st 4713 B.C., that is midday as measured on the Greenwich meridian on January 1st of that year."[1] To compute the number of days between two dates, it is only necessary to compute the Julian dates for the two dates in question and find the difference between the two Julian dates. The Julian date calculation takes leap years and the conversion between the Julian and Gregorian calendar into account, so lookup tables are not necessary in order to make the calculation.

1.2 Measures of time

Calculations in this program also rely on a few different types of time: Greenwich mean time (GMT), now known as universal time (UT), local civil time, Greenwich sidereal time (GST), and local sidereal time (LST). GMT is based on the local civil time at 0° longitude. Local civil time is determined by the mean motion of the sun; noon in local civil time is the time at which the mean sun is highest in the sky. The mean sun is a fictitious sun whose motion is uniform as it travels along the equator.[2] The real sun has non-uniform motion due to the elliptical nature of Earth's orbit and the tilt of its axis. The difference between the two times is found using the equation of time.

Sidereal time is based on the motion of the stars. A sidereal day is the time it takes for a star to return to the same position in the sky. A sidereal day is thus shorter than a solar day—23h 56m as opposed to 24h. The prefixes of Greenwich and local specify where the measurement of sidereal time is taken.

In the program, it is necessary to convert from GMT to GST and back again. These conversions depend on constants that do not change from year to year, as well as a constant that is different for each year. A function, called `constant_B`, uses an algorithm to calculate this not-so-constant constant.

Both conversions from GMT to GST and from GST to GMT rely on the number of days between the day in question and January 0.0. January 0.0 is defined as "midnight between December 30th and 31st of the previous year."[3] This simplifies the calculations needed for other astronomical calculations.

[1] 9.

[2] 92

[3] 6.

The local sidereal time is easily calculated by correcting GST by the hour-equivalent of the difference in longitude (there is one hour per 15 degrees of longitude).

2 Ecliptic coordinates

The ecliptic plane is the plane on which the Earth moves as it revolves around the sun. To observers on Earth, the Sun appears to move around the Earth on a path in the sky, called the ecliptic, defined by this plane. Since the sun moves exactly on this path, it is therefore easy to compute its angular distance away from the path; this coordinate, $\beta_{\odot}$, is always zero. The only coordinate that needs to be calculated, then, is the ecliptic longitude, λ. The ecliptic longitude is defined as the angular distance from the first point of Ares, the point at which the ecliptic crosses the celestial equator. The celestial equator, in turn, is a line in the sky where the plane of the Earth's equator would intersect the celestial sphere. The ecliptic longitude is measured in the direction that the Sun moves along the ecliptic—eastward—and the angle at which the ecliptic intersects the equatorial plane is symbolized by ϵ.

Because the Earth orbits around the Sun in an elliptical orbit—or it appears that the Sun orbits around the Earth in an elliptical orbit—it is easier to pretend that the orbit is circular and then to make corrections for the elliptical orbit. The mean anomaly, M, is the angle through which the sun has passed since perigee:

$$M = \frac{360}{365.2422} D + \epsilon_g - \varpi_g, \tag{1}$$

"where ϵ_g and ϖ_g are the mean longitude of the sun at the epoch and perigee respectively."[4] D is the number of days since the epoch. The program then adds or subtracts multiples of 360° until M is in the range 0°–360°.

Once the mean anomaly is known, corrections can be made to allow for the elliptical orbit of the Sun. The eccentric anomaly, E, is found by numerically solving the trancendental equation,

$$E - e \sin E = M, \tag{2}$$

where E and M are expressed as radians.

Once the eccentric anomaly is found, the true anomaly, ν, by solving

$$\tan \frac{\nu}{2} = \left[\frac{1+e}{1-e} \right]^{\frac{1}{2}} \tan \frac{E}{2} \tag{3}$$

with all the angles expressed as radians. The ecliptic longitude is then found with

$$\lambda_{\odot} = \nu + \varpi_g, \tag{4}$$

where the variables are expressed in degrees. Here again, the program then adds or subtracts multiples of 360° until $\lambda_{\odot}$ is in the range 0°–360°.

[4] 81

3 Conversion of ecliptic coordinates to equatorial coordinates

Once the equatorial coordinates are known, it is fairly easy to convert them to equatorial coordinates, expressed as right ascension, α, and declination, δ. Right ascension is ordinarily expressed in hours, and declination is ordinarily expressd as degrees.

If ϵ, the opliquity of the ecliptic (the angle between the ecliptic plane and the equatorial plane) is known, then the right ascension (in degrees) is given by

$$\alpha = \tan^{-1}\left(\frac{\sin\lambda\cos\epsilon - \tan\beta\sin\epsilon}{\cos\lambda}\right), \tag{5}$$

and the declination (in degrees) is given by

$$\delta = \sin^{-1}\left(\sin\beta\cos\epsilon + \cos\beta\sin\epsilon\sin\lambda\right). \tag{6}$$

It is necessary to make sure that $\tan^{-1}$ gives an angle in the correct quadrant. Fortunately, Fortran has the `ATAN2` function which gives the correct angle, taking the numerator and denominator of the argument of $\tan^{-1}$ as its arguments.

Also, since α is given in degrees by this method, it must be converted to hours by dividing α by 15.

4 Rising and setting times

The equations that follow give a rough calculation of the rise and set times of a celestial object, given its equatorial coordinates, date, local latitude, and local longitude. These are rough calculations because they do not take parallax and atmospheric distortion into account.

If the local latitude is expressed by ϕ, then

$$\mathrm{LST_r} = 24 - \frac{1}{15}\cos^{-1}\left(-\tan\phi\,\tan\delta\right) + \alpha \tag{7}$$

$$\mathrm{LST_s} = \frac{1}{15}\cos^{-1}\left(-\tan\phi\,\tan\delta\right) + \alpha \tag{8}$$

give the LST of rising and LST of setting, respectively. Once the LST times are found, they can be converted to GMT times.

Since rise and set times are found for two times—one for the date in question and one 24 hours later—an interpolation can be used to improve the accuracy of the rise and set times:

$$T = \frac{24.07 \times \mathrm{ST1}}{24.07 + \mathrm{ST1} - \mathrm{ST2}}\,\text{hours}, \tag{9}$$

where ST1 and ST2 represent the two sets of times found for rising and setting (a T is found for both rising and setting, known as T_r and T_s, respectively).

5 Corrections for atmospheric effects

The Earth's atmosphere can cause the apparent time of rising or setting to differ by a few minutes from the actual time or rising. If the vertical displacement is x (which accounts for both the sun's finite angular diameter, horizontal parallax, and atmospheric refraction), then the time correction is

$$\Delta t = \frac{240y}{\cos\delta}\,\text{seconds}, \tag{10}$$

where

$$y = \sin^{-1}\left(\frac{\sin x}{\sin\psi}\right)\ \text{degrees} \tag{11}$$

and

$$\psi = \cos^{-1}\left(\frac{\sin\phi}{\sin\delta}\right)\ \text{degrees.} \tag{12}$$

The Sun's angular diameter is $0^\circ.533$, its horizontal parallax is 8.79 arcseconds, and the atmosphere contributes 34 minutes of arc of refraction, so

$$x = \frac{0^\circ.533}{2} + 8''.79 + 34'. \tag{13}$$

For this calculation, δ is the average declination for the two days for which the Sun's position was calculated.

Once Δt has been found, it can be converted to seconds and added to T_s and subtracted from T_r.

6 Final steps

T_r and T_s at this stage are LST times. In order to convert them to local civil times, they first need to be converted to GST times. The GST times are then converted to GMT times. Finally, the time zone offset is applied to the GMT times, resulting in the local civil times for sunrise and sunset.

7 Testing

To see whether the output of my program was any good, I compared its calculated times to times calculated by the U.S. Naval Observatory, using a web form at `http://aa.usno.navy.mil/data/docs/RS_OneDay.html`. I compared times for several dates for an observer in Avon, MN (45.6°N, 94.5°W) as seen in the table below:

Date	USNO T_r	Program T_r	USNO T_s	Program T_s
01 January 1980	07h 58m	07h 55m	16h 44m	16h 40m
24 August 1980	06h 30m	06h 28m	20h 10m	20h 09m
07 September 1980	06h 47m	06h 45m	19h 44m	19h 43m
01 January 2007	07h 59m	07h 55m	16h 45m	16h 40m
24 August 2007	06h 29m	06h 27m	20h 11m	20h 10m
07 September 2007	06h 46m	06h 45m	19h 45m	19h 44m

8 Conclusion

The results from the Fortran program seem reasonably accurate; the times it predicts are within five minutes of the times predicted by the USNO. The results match exactly those which were given by the book for an observer on 7 September 1979 at longitude 0°, 52°N.

Bibliography

Duffett-Smith, Peter. *Practical astronomy with your calculator.* Second Edition. Cambridge, 1981.