Pergamon

PII: S0038–092X(00)00156–0

Solar Energy Vol. 70, No. 5, pp. 431–441, 2001

0038-092X/01/$ - see front matter

www.elsevier.com/locate/solener

COMPUTING THE SOLAR VECTOR

MANUEL BLANCO-MURIEL*,†, DIEGO C. ALARCÓN-PADILLA*, TEODORO LÓPEZ-MORATALLA** and MARTÍN LARA-COIRA**

*CIEMAT – Plataforma Solar de Almería, Aptdo. 22, E-04200 Tabernas, Almería, Spain

**Real Instituto y Observatorio de la Armada, E-11110 San Fernando, Cádiz, Spain

Received 4 August 2000; revised version accepted 15 November 2000

Communicated by LORIN VANT-HULL

Abstract—High-concentration solar thermal systems require the Sun to be tracked with great accuracy. The higher the system concentration, the greater this accuracy must be. The current trend in solar concentrator tracking systems is to use open-loop controllers that compute the direction of the solar vector based on location and time. To keep down the price of the tracking system, the controller is based on a low-cost microprocessor. These two facts impose important restrictions on the Sun position algorithm to be used in the controller, as it must be highly accurate and efficiently computable at the same time. In this paper, various algorithms currently available in the solar literature are reviewed and a new algorithm, developed at the Plataforma Solar de Almería, which combines these two characteristics of accuracy and simplicity, is presented. The algorithm allows of the true solar vector to be determined with an accuracy of 0.5 minutes of arc for the period 1999–2015. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Solar vector; Sun position; Solar tracking.

1. INTRODUCTION

High-concentration solar thermal systems require the Sun to be tracked with great accuracy. Although the degree of accuracy required depends on the specific characteristics of the concentrating system being analyzed, in general, the higher the system concentration the higher the tracking accuracy needed. For a solar thermal central receiver system (CRS) with a concentration ratio of about 1000, Vant-Hull and Hildebrandt (1976) estimate this to be approximately 3.5 minutes of arc.

The current trend in solar concentrator tracking systems is to use an open-loop local controller that computes the direction of the solar vector based on geographical location and time. In a typical configuration, the local controller does everything necessary to ensure continuous automatic positioning of the concentrator in such a way that communication with the central controller is limited to setting aiming points, sending status data and accepting safety orders. This reduces the need for interaction with the central controller and minimizes data traffic. To keep down the price of the tracking system, the controller is based on a low-cost microprocessor, which, in addition to computing the Sun position, has to be able to control many other tracking system functions.

All this imposes important restrictions on the Sun-position algorithm to be used in the controller. It has to be highly accurate and efficiently computable at the same time. The greater the accuracy in computing the Sun position, the greater the margin of tolerance will be for other sources of error, such as optical or mechanical, that may arise within the concentrating system. It is therefore desirable for the accuracy of the Sun-position algorithm be as high as possible.

This paper reviews what has been published in the last 38 years in the solar literature concerning the determination of the Sun position, compares the different algorithms proposed and introduces a new, more accurate and simpler algorithm.

2. BIBLIOGRAPHIC REVIEW

The solar literature contains a wide range of papers referring to the calculation of the Sun position. These calculations can be classified into two groups. The first is a group of relatively simple formulae and algorithms that, given the day of the year, estimate basic Sun-position

†Author to whom correspondence should be addressed. Tel.: +34-950-387-914; fax: +34-950-365-300; e-mail: manuel.blanco@psa.es

parameters, such as the solar declination or the equation of time (Cooper, 1969; Lamm, 1981; Spencer, 1971; Swift, 1976). The second consists of more complex algorithms (Michalsky, 1988; Pitman and Vant-Hull, 1978; Walraven, 1978) that, given the precise location and instant of observation, compute the position of the Sun in ecliptic (ecliptic longitude, obliquity), celestial (declination, right ascension), and/or local horizontal coordinates (zenith distance and solar azimuth).

Usually, a solar tracking system needs ''to know'' the local horizontal coordinates of the Sun at the system location (specified by its geographic longitude and latitude) at any given instant of time (specified by the date and the universal time). From this point of view, none of the reported algorithms are complete. Either their input is more specialized (i.e., the input values to the algorithm are not the geographical longitude and latitude, date and universal time) or their output falls short of what is required (i.e., the algorithm does not provide the local horizontal Sun coordinates as the result), or both. Table 1 shows the variables calculated by each of the different algorithms analyzed.

In the group of algorithms and formulae used to calculate the celestial coordinates of the Sun from the number of the day of the year, the most important and the most widely cited in the solar literature are the Spencer formulae (Spencer, 1971). In Spencer's article, as well as in Iqbal (1983), these formulae are said to have a maximum error of 3 minutes of arc for the declination and around 35 seconds for the equation of time. However, this claim does not seem to be correct. Spencer himself indicates elsewhere in his article that the mere use of the number of the day of the year as an integer introduces an error in the solar declination that can vary approximately 0.5 degrees at the equinoxes and less than 1 minute of arc at the solstices. Even considering that the day of the year varies continuously over time, any analysis of the solar declination provided by the Spencer formula shows clearly that the maximum error can be as great as 0.28 degrees. Michalsky (1988) has already hinted at this. Other authors (Cooper, 1969; Swift, 1976) have provided equations for calculating the solar declination that are simpler than those of Spencer, but less accurate as well.

Pitman and Vant-Hull (1978) were among the first to point out that high-precision algorithms were needed for tracking systems or to compare the predictions of direct irradiance models with pyrheliometer readings. Based on the equations of The Astronomical Ephemeris and The American Ephemeris and Nautical Almanac (1961), they presented formulae for the calculation of the ecliptic Sun coordinates, its declination and the equation of time. They also discuss many of the phenomena affecting the Sun's position, and estimate the errors resulting when these phenomena are disregarded. Although they do not provide formulae to correct for refraction, they set limits to the errors associated with the absence of such a correction. These are estimated to be around 5 minutes of arc for over 10 degrees elevation and around 4 minutes of arc for over 15 degrees. They claim that their program, called SUNLOC and based on their formulae, in its most accurate mode, can compute the true horizontal coordinates of the Sun (i.e., the horizontal Sun coordinates disregarding the effects of atmospheric refraction) with a maximum error on the order of 41 seconds of arc. Unfortunately, they do not provide the SUNLOC program code in their article.

In the same year, 1978, Walraven presented a complete algorithm to calculate the local horizontal coordinates of the Sun. This algorithm, like the Pitman and Vant-Hull formulae, was also based on the equations of The Astronomical Ephemeris (1961), although it uses these equations in a simpler form. In his paper, Walraven claims that the algorithm can be used to estimate the position of the Sun with an accuracy of 36 seconds of arc. Contradicting this claim, Zimmerman (1981) reports errors of as much as 216 seconds of arc in

Table 1. Solar parameters calculated by different authors

Reference	Declination	Right ascension	Equation of time	Azimuth	Zenith distance or elevation
Cooper, 1969	X				
Spencer, 1971	X		X		
Swift, 1976	X				
Pitman and Vant-Hull, 1978	X		X		
Walraven, 1978	X	X		X	X
Lamm, 1981			X		
Michalsky, 1988	X	X	X	X	X

azimuth and 47 seconds of arc in elevation when using the algorithm to estimate the Sun position for the period 1979–1986.

The publication of the Walraven article was followed by an Erratum (Walraven, 1979) and several technical notes and letters to the editor (Archer, 1980; Ilyas, 1983; 1984; Muir, 1983; Pascoe, 1984; Spencer, 1989; Wilkinson, 1981; 1983). Most of these were concerned with the FORTRAN implementation of the algorithm as presented in the original paper or amended in the Erratum.

Interestingly enough, one significant error associated with the implementation of Walraven's algorithm seems to have remained unreported. It has to do with the computation of one of the inputs – the day number. For Walraven this number is ''*the day number of the year starting with* 1 *for* 1 *January, except in leap years when* 1 *should be subtracted from the day number if it is before* 1 *March*''. However, his algorithm does not work correctly for days before 1 March in leap years if the above mentioned subtraction is carried out. Among the reported errors, the fact that the algorithm does not work appropriately for the Southern Hemisphere is the most important one (Spencer, 1989).

To facilitate the use of Walraven's algorithm, Archer (1980) proposed a subroutine to calculate the day number, given the year, month and day-of-month as input. This subroutine is valid for the period 1901–2099 and does not include Walraven's subtraction. Archer's article is the first of those reviewed in this study which provides an equation to calculate the correction for atmospheric refraction in determining the solar elevation.

In 1981, Lamm presented an expression to calculate the equation of time, claiming an absolute maximum error of 3.6 seconds of time. Holland and Mayer (1988) implemented a program to compute the Sun position using the equation of time proposed by Lamm and compared its results with those obtained using the algorithm of Walraven. From this comparison they conclude that errors of about 5 degrees in determining the solar elevation could result from the use of Lamm expression. Surprisingly, this conclusion is in contradiction with Lamm's claims about the accuracy of its equation, a claim that Holland and Mayer do not dispute.

Ten years after the publication of Walraven's algorithm, Michalsky (1988) published another algorithm also based on the simplified equations of the Nautical Almanac. Although his algorithm uses fewer operations to compute the true horizontal solar coordinates and does it more accurately than that of Walraven, it has a similar drawback – in its original form, it does not work properly for the Southern Hemisphere either.

Unlike Walraven, Michalsky includes an expression to estimate the effect of refraction on the apparent elevation of the Sun. This refraction model provides the apparent Sun elevation as a function of its true elevation. Like the rest of the formulae in the algorithm, the refraction model proposed by Michalsky is also based on expressions given in the Astronomical Almanac (The Astronomical Almanac, 1985). Kambezidis and Papanikolaou (1990) sustain the need to adopt this model instead of the one proposed by Archer (1980) since the Archer refraction model causes problems in estimating the time of Sunrise and Sunset. This claim is unfounded, however. What Kambezidis and Papanikolaou seem to be implicitly addressing is the fact that the Archer model is not defined for a short range of negative true elevation angles – those in the interval $[-2.726°; -2.678°]$. This range of values, however, is so far from the minimum for which any refraction model would be used, that it does not preclude the use of the Archer refraction model for any practical application.

Others elsewhere have proposed more sophisticated refraction models which take as input relevant meteorological variables, such as ambient temperature and pressure, in addition to the true elevation of the Sun (Camblor-Ordiz, 1995; Meeus, 1998; Duffett-Smith, 1992). However, the discussion of such models lies outside the scope of this article.

3. COMPARISON OF THE DIFFERENT ALGORITHMS

Of all the algorithms mentioned above, only four (Spencer, Pitmann and Vant-Hull, Walraven and Michalsky) are complete in the sense that, without the need for additional information, the true horizontal coordinates of the Sun can be estimate based on their results. To establish which is the best, their accuracy has been assessed and compared. Assessment of the accuracy of an algorithm is possible thanks to the Multiyear Interactive Computer Almanac (MICA). This software product of the United States Naval Observatory (1998) can be used to obtain highly accurate astronomical estimates regarding the

position and motion of celestial objects, including the Sun. Using MICA, reference values of the true horizontal coordinates of the Sun were generated for the period 1990–2005, with an accuracy of half a second of arc. A series of such reference values were obtained for 184 104 instants of time, (those generated by sampling the period from 0^h UT January 1st, 1990 to 23^h 40^m UT December 31st every 20 minutes) and compared with the corresponding values provided by the algorithms. The comparison was performed for the location of the Plataforma Solar de Almería (longitude W 2°21′36″, latitude N 37°6′2″). For each algorithm the following statistical indicators were evaluated:

1. Average of the error in azimuth ($\overline{\Delta\gamma}$), error in zenith distance ($\overline{\Delta\theta_z}$), and angular deviation ($\overline{\Theta}$).
2. Standard deviation of the error in azimuth (σ_γ), error in zenith distance(σ_{θ_z}), and angular deviation (σ_Θ).
3. Mean deviation of the error in azimuth (MD_γ), error in zenith distance (MD_{θ_z}), and angular deviation (MD_Θ).
4. Range of the error in azimuth (R_γ), error in zenith distance (R_{θ_z}), and angular deviation (R_Θ).

The errors in azimuth and zenith distance (γ and θ_z), are computed as the difference between the value supplied by the algorithm and that supplied by MICA. The angular deviation is computed as the angle between the solar vector supplied by the algorithm and that supplied by MICA. Fig. 1 shows the origin and positive direction of the angles considered. Table 2 shows the expressions of the statistical indicators used. Two main conclusions can be drawn from the results obtained:

1. The Spencer Algorithm, because of its relatively poor accuracy, is not suited for use in concentrating solar thermal tracking systems,
2. The other algorithms analyzed are accurate enough for that purpose, the one proposed by Michalsky being the most accurate.

Table 2. Statistical indicators used for all variables analyzed (error in zenith distance, error in azimuth and Sun vector deviation)

Statistical indicator	Expression
Average	$\bar{x} = \frac{1}{N}\,\Sigma_{i=1}^{N}\, x_i$
Standard deviation	$\sigma_x = \left(\frac{1}{N-1}\,\Sigma_{i=1}^{N}\,(x_i - \bar{x})^2\right)^{1/2}$
Mean deviation	$MD_x = \frac{1}{N}\,\Sigma_{i=1}^{N}\,(x_i - \bar{x})$
Range	$R_x = [\min(x_i), \max(x_i)]\ i = 1 \ldots N$

4. THE PSA ALGORITHM

Even though the Michalsky algorithm may be considered good enough for most solar tracking applications, its accuracy, computing efficiency, and ease of use can still be improved. These tasks have been undertaken and, as a result, a new algorithm called the PSA algorithm, having the following characteristics, has been developed:

1. Its ease of use has been improved by incorporating an efficient method of computing the Julian Day from the calendar date and Universal Time.
2. Memory management has been improved by controlling the scope and life span of variables.
3. Speed and robustness have been improved by eliminating unnecessary operations and using simple robust expressions for calculating the solar azimuth, which are valid for both hemispheres.
4. Accuracy has been improved by modifying the simplified equations of the Nautical Almanac used by Michalsky with the introduction of new coefficients and new terms, and including parallax correction.

4.1. Accuracy

The accuracy of the PSA algorithm was evaluated in the same manner as the Michalsky and the three other algorithms referred to earlier. The results are shown in Table 4. A comparison of

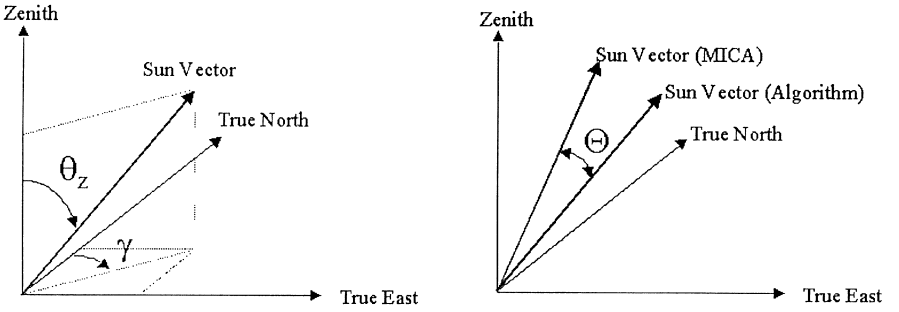

Fig. 1. Angular coordinates analyzed.

Table 3. Comparison of the performance of different algorithms in determining the Sun position for the years 1990 to 2005. (In minutes of arc)

	Average	Standard deviation	Mean deviation	Range
Error in Zenith Distance				
Spencer	−0.657	13.895	11.34	[−45.568, 40.918]
Pitman and Vant-Hull	−0.119	0.235	0.186	[−1.036, 0.649]
Walraven	−0.119	0.196	0.155	[−0.798, 0.486]
Michalsky	−0.121	0.139	0.110	[−0.666, 0.345]
Error in Azimuth				
Spencer	0.059	15.377	12.378	[−53.516, 42.066]
Pitman and Vant-Hull	−0.068	0.376	0.270	[−2.875, 2.287]
Walraven	−0.176	0.253	0.187	[−2.465, 0.894]
Michalsky	−0.042	0.214	0.155	[−1.917, 1.532]
Sun Vector Deviation				
Spencer	15.966	9.203	7.588	[0.055, 46.460]
Pitman and Vant-Hull	0.322	0.175	0.140	[0.000, 1.097]
Walraven	0.271	0.139	0.112	[0.000, 0.798]
Michalsky	0.207	0.107	0.085	[0.000, 0.667]

Table 4. Performance of the PSA algorithm in determining the Sun position between the years 1990 and 2005 (all quantities in minutes of arc)

	Average	Standard deviation	Mean deviation	Range
Error in zenith distance	−0.001	0.114	0.091	[−0.398, 0.370]
Error in azimuth	0.002	0.190	0.138	[−1.568, 1.461]
Sun vector deviation	0.147	0.080	0.065	[0.000, 0.409]

these results with those presented in Table 3 shows that the PSA algorithm behaves better than the Michalsky algorithm in computing zenith distance, azimuth and direction vector of the Sun. For instance, the standard deviation of the error in computing zenith distance, azimuth and direction vector of the Sun is 18, 11, and 25% lower than for the Michalsky algorithm.

The probability density functions of the errors in calculating the zenith distance, azimuth and vector direction of the Sun using the five algorithms analyzed are represented in Figs. 2–4. In all of these figures, the superior performance of the PSA algorithm is evident.

Figs. 5 and 6 show that the accuracy of the PSA algorithm in computing the true horizontal coordinates of the Sun does not seem to decrease significantly with time in the period analyzed (1990–2005).

Finally, Fig. 7 shows that the behavior of the

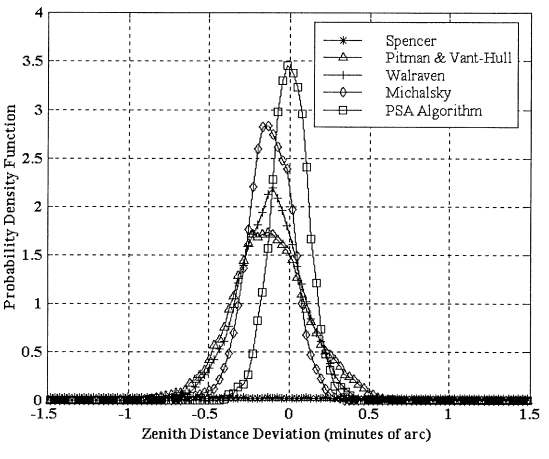

Fig. 2. Distribution of error in the calculation of the zenith distance.

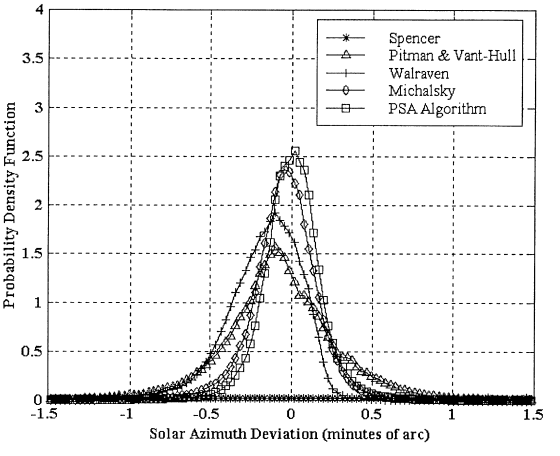

Fig. 3. Distribution of error in the calculation of the solar azimuth.

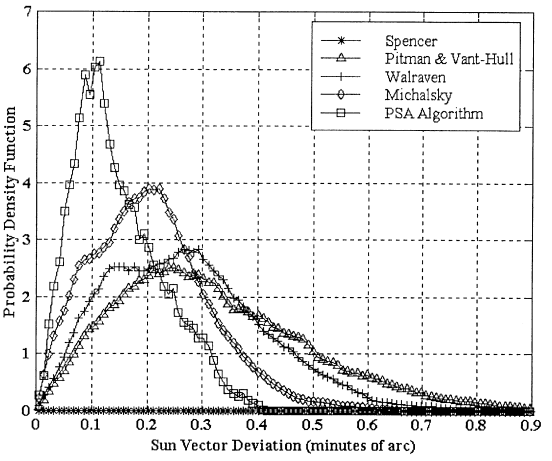

Fig. 4. Distribution of error in determining the solar vector.

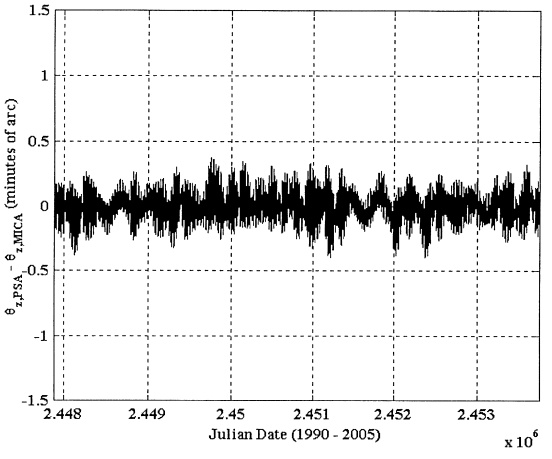

Fig. 5. Evolution of error in determining the zenith distance for the period 1990–2005.

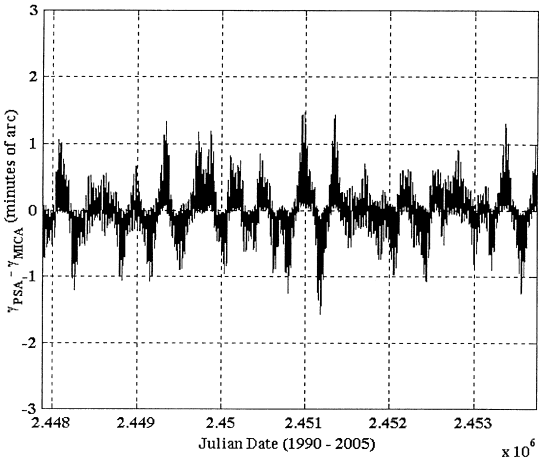

Fig. 6. Evolution of the error in determining solar azimuth for the period 1990–2005.

PSA algorithm behaves correctly for both hemispheres and its accuracy seems to be largely independent of the location at which the Sun position is being computed.

4.2. Computational efficiency

A streamlined C++ implementation of the PSA algorithm (available in electronic form at http://www.psa.es/sdg/sunpos.htm) is given in Appendix A. For purposes of comparison, Table 5 shows the number of main mathematical and logical operations in this implementation of the PSA algorithm and for the original FORTRAN implementation of the Michalsky algorithm as presented in his 1988 article. In comparing those numbers, it should be recalled that the PSA algorithm is more functional than Michalsky's. For instance, while the user of the Michalsky algorithm has to supply the number of day of the year as an integer and the Universal Time as a decimal number as input data, the user of the PSA algorithm has only to supply the date in the usual format (day, month and year) and the Universal Time, also in the familiar format (hours, minutes and seconds).

A direct comparison of the speed of both algorithms is not straightforward, since they are implemented in entirely different computer languages. However because of the fact that the PSA algorithm has slightly more arithmetic operations than the Michalsky algorithm, but fewer if-statements, trigonometric and module operations, it should run faster than Michalsky's, since the latter are computationally more expensive than the former.

5. DESCRIPTION OF THE ALGORITHM

The input to the PSA Algorithm is time and location. The time for the instant under consideration is given as the date (year, month, and day) and the Universal Time (hours, minutes and seconds). The location is given as the longitude and latitude of the observer in degrees. Latitude is considered positive to the North and longitude to the East.

The Julian Day, jd, is computed from the input data by the following expression:

$$\begin{aligned} jd = {} & (1461 \times (y + 4800 + (m - 14)/12))/4 \\ & + (367 \times (m - 2 - 12 \times ((m - 14)/12)))/12 \\ & - (3 \times ((y + 4900 + (m - 14)/12)/100))/4 \\ & + d - 32075 - 0.5 + hour/24.0 \end{aligned} \tag{1}$$

where m is the month, y is the year, d is the day

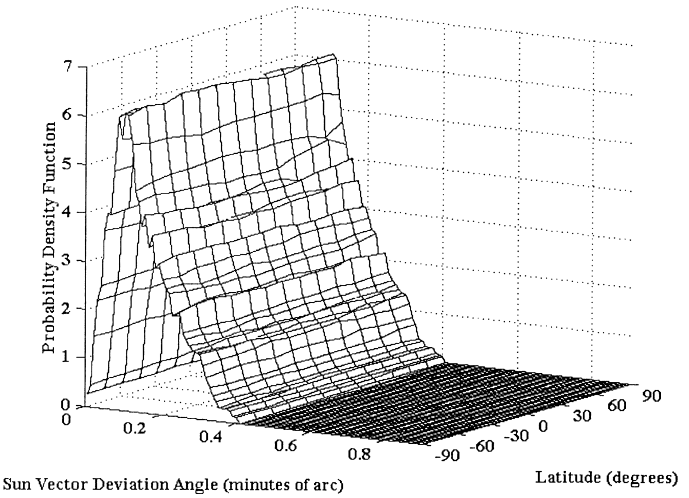

Fig. 7. Distribution of error in determining the solar vector for 1999–2005 and different latitudes with the PSA algorithm.

of the month and *hour* is the hour of the day in Universal Time in decimal format, i.e., it includes the minutes and seconds as a fraction of an hour, and all divisions except the last are integer divisions.

This expression differs from the one proposed by Fliegel and Van Flandern (1968) only in the last two terms. The addition of these terms makes it possible to compute the Julian Day in decimal form for any given instant of time. As for the original Fliegel and Van Flandern expression, the expression proposed is valid for any date after March 1, 4800 B.C., whether Gregorian or proleptic Gregorian.

The ecliptic coordinates of the Sun are computed from the Julian Day, by the following set of equations (all angles in radians):

$$n = jd - 2451545.0 \tag{2}$$

$$\Omega = 2.1429 - 0.0010394594 \times n \tag{3}$$

$$L(\text{mean longitude}) = 4.8950630 + 0.017202791698 \times n \tag{4}$$

$$g(\text{mean anomaly}) = 6.2400600 + 0.0172019699 \times \text{n} \tag{5}$$

$$l(\text{ecliptic longitude}) = L + 0.03341607 \times \sin(g) + 0.00034894 \times \sin(2g) - 0.0001134 - 0.0000203 \times \sin(\Omega) \tag{6}$$

$$ep(\text{obliquity of the ecliptic}) = 0.4090928 - 6.2140 \times 10^{-9} \times n + 0.0000396 \times \cos(\Omega) \tag{7}$$

The conversion from ecliptic to celestial coordinates is accomplished using the standard trigonometric expressions

$$ra(\text{right ascension}) = \tan^{-1}\left[\frac{\cos(ep) \times \sin(l)}{\cos(l)}\right] \tag{8}$$

$$\delta(\text{declination}) = \sin^{-1}[\sin(ep) \times \sin(l)] \tag{9}$$

Care must be taken to ensure that the right ascension is properly computed as a value between 0 and 2π in Eq. (8).

The conversion from celestial to horizontal coordinates is achieved by the following set of equations:

Table 5. Number of main mathematical and logical operations for the PSA and Michalsky algorithms

Algorithm	Additions/ subtractions	Products/ divisions	Direct trigonometric function	Inverse trigonometric function	If statements	Module functions
Michalsky	28	25	19	4	10	5
PSA	35	36	16	4	2	0

Table 6. Comparison between azimuth and zenith distance supplied by the Real Observatorio de la Armada and MICA for 1999–2005 (in seconds of arc)

	Average	Standard deviation	Mean deviation	Range
Difference in zenith distance	−0.015	0.148	0.125	[−0.346, 0.347]
Difference in azimuth	−0.041	0.294	0.220	[−1.350, 1.132]

$$gmst = 6.6974243242 + 0.0657098283 \times n + hour \quad (10)$$

$$lmst = (gmst \times 15 + Long) \times (\pi/180) \quad (11)$$

$$\omega(\text{hour angle}) = lmst - ra \quad (12)$$

$$\theta_z = \cos^{-1}[\cos(\Phi)\cos(\omega)\cos(\delta) + \sin(\delta)\sin(\Phi)] \quad (13)$$

$$\gamma = \tan^{-1}\left[\frac{-\sin(\omega)}{\tan(\delta)\cos(\Phi) - \sin(\Phi)\cos(\omega)}\right] \quad (14)$$

$$Parallax = \frac{EarthMeanRadius}{AstronomicalUnit} \times \sin(\theta_z) \quad (15)$$

$$\theta_z = \theta_z + Parallax \quad (16)$$

where the mean radius of the Earth and distance to the Sun considered are given by:

$$EarthMeanRadius = 6371.01 \text{ km} \quad (17)$$

$$AstronomicalUnit = 149597890 \text{ km} \quad (18)$$

6. LONG-TERM VALIDATION OF THE PSA ALGORITHM

The MICA program only supplies Sun-related data for the interval between 1990 and 2005. However with the collaboration of the Real Observatorio de la Armada in San Fernando (Spain), it has been possible to test validity of the PSA Algorithm through the year 2015.

Following our request, the Real Observatorio generated 447 048 reference values of the true horizontal coordinates of the Sun, by evaluating such coordinates, using high precision formulae, every 20 minutes for the time period elapsed since 0:00 UT January 1, 1999 through 23:40 UT December 31, 2015. Obviously, the accuracy of such reference data is not constant across the time span under consideration. For any year after 2000 the difference between Universal Time and Dynamic Terrestrial Time has to be extrapolated, and the larger the extrapolation the larger the uncertainty. Specifically, to generate the reference data, it was assumed that, on the average, the difference between Universal Time and Dynamic Terrestrial Time has grown steadily by one second per year. As a result the error of the reference data cannot be guaranteed as under half a second of arc for the complete time interval under consideration. The comparison of the values generated by MICA and those provided by the Real Observatorio for the period 1999–2005 shown in Table 6 seems to indicate that this error may be larger – on the order of one second of arc.

Table 7 shows the results of the comparison between the reference values supplied by the Real Observatorio de la Armada and those generated by the PSA and the Michalsky algorithm. In this table it is clear that the PSA algorithm is better than Michalsky's for the whole period under consideration. For this period, the standard deviation of the error in computing the true zenith distance, azimuth and direction vector of the sun is 22, 14, and 28% lower than for the Michalsky algorithm, respectively. Likewise, the range of the

Table 7. Comparison of the PSA and Michalsky algorithms and for the determination of the Sun position for 1999–2015 (in minutes of arc)

	Average	Standard deviation	Mean deviation	Range
Error in Zenith Distance				
Michalsky	−0.128	0.137	0.109	[−0.666 0.340]
PSA Algorithm	−0.008	0.107	0.084	[−0.396 0.366]
Error in Azimuth				
Michalsky	−0.065	0.206	0.150	[−1.903 1.344]
PSA Algorithm	0.000	0.177	0.127	[−1.553 1.443]
Sun Vector Deviation				
Michalsky	0.208	0.110	0.086	[0.000 0.667]
PSA Algorithm	0.136	0.079	0.063	[0.000 0.433]

errors incurred in computing the true zenith distance, azimuth and direction vector of the Sun is 24, 8 and 35% smaller for the PSA algorithm than for the Michalsky.

7. CONCLUSIONS

High-concentration solar thermal systems require algorithms that are accurate and computationally inexpensive to determine the coordinates of the Sun.

The solar literature presents several examples of such algorithms, all of which are based on the use of simplified Nautical Almanac orbital equations.

A new algorithm, called the PSA Algorithm, has been developed, that constitutes an improvement over those in the literature. The new algorithm provides estimates of the true azimuth and zenith angle of the Sun that are respectively 15% and 22% better in average than those provided by any similar algorithm proposed up to now. For the period 1999 to 2015, it enables the direction of the solar vector to be determined with an error of less than 0.5 minutes of arc. In addition, it is simpler and easier to use than existing algorithms.

NOMENCLATURE

jd	Julian Day
hour	Hour of the day in Universal Time and decimal format
n	Difference between the current Julian Day and Julian Day 2 451 545.0 (noon 1 January 2000)
L	Mean longitude of the Sun
g	Mean anomaly of the Sun
l	Ecliptic longitude of the Sun
ep	Obliquity of the ecliptic
ra	Right ascension
δ	Declination
gmst	Greenwich mean sidereal time
lmst	Local mean sidereal time
long	Geographical longitude
Φ	Geographical latitude
ω	Hour angle
θ_z	Zenith distance
γ	Solar azimuth

APPENDIX A

```
// This file is available in electronic form at http://www.psa.es/sdg/sunpos.htm

#ifndef __SUNPOS_H
#define __SUNPOS_H

// Declaration of some constants
#define pi    3.14159265358979323846
#define twopi (2*pi)
#define rad   (pi/180)
#define dEarthMeanRadius     6371.01         // In km
#define dAstronomicalUnit    149597890       // In km

struct cTime
{
       int iYear;
       int iMonth;
       int iDay;
       double dHours;
       double dMinutes;
       double dSeconds;
};

struct cLocation
{
       double dLongitude; // input in degrees positive towards east
       double dLatitude;  // input in degrees positive towards north
};

struct cSunCoordinates
{
       double dZenithAngle; // returned in degrees from zenith
       double dAzimuth;     // returned in degrees positive east from north
};

void sunpos(cTime udtTime, cLocation udtLocation, cSunCoordinates *pudtSunCoordinates);

#endif
```

```
// This file is available in electronic form at http://www.psa.es/sdg/sunpos.htm

#include "sunpos.h"
#include <math.h>

void sunpos(cTime udtTime,cLocation udtLocation, cSunCoordinates* pudtSunCoordinates)
{
	// Main variables
	double dElapsedJulianDays;
	double dDecimalHours;
	double dEclipticLongitude;
	double dEclipticObliquity;
	double dRightAscension;
	double dDeclination;

	// Auxiliary variables
	double dY;
	double dX;

	// Calculate difference in days between the current Julian Day
	// and JD 2451545.0, which is noon 1 January 2000 Universal Time
	{
		double dJulianDate;
		long int liAux1;
		long int liAux2;
		// Calculate time of the day in UT decimal hours
		dDecimalHours = udtTime.dHours + (udtTime.dMinutes
			+ udtTime.dSeconds / 60.0 ) / 60.0;
		// Calculate current Julian Day
		liAux1 =(udtTime.iMonth-14)/12;
		liAux2=(1461*(udtTime.iYear + 4800 + liAux1))/4 + (367*(udtTime.iMonth
			- 2-12*liAux1))/12- (3*((udtTime.iYear + 4900
		+ liAux1)/100))/4+udtTime.iDay-32075;
		dJulianDate=(double)(liAux2)-0.5+dDecimalHours/24.0;
		// Calculate difference between current Julian Day and JD 2451545.0
		dElapsedJulianDays = dJulianDate-2451545.0;
	}

	// Calculate ecliptic coordinates (ecliptic longitude and obliquity of the
	// ecliptic in radians but without limiting the angle to be less than 2*Pi
	// (i.e., the result may be greater than 2*Pi)
	{
		double dMeanLongitude;
		double dMeanAnomaly;
		double dOmega;
		dOmega=2.1429-0.0010394594*dElapsedJulianDays;
		dMeanLongitude = 4.8950630+ 0.017202791698*dElapsedJulianDays; // Radians
		dMeanAnomaly = 6.2400600+ 0.0172019699*dElapsedJulianDays;
		dEclipticLongitude = dMeanLongitude + 0.03341607*sin( dMeanAnomaly )
			+ 0.00034894*sin( 2*dMeanAnomaly )-0.0001134
			-0.0000203*sin(dOmega);
		dEclipticObliquity = 0.4090928 - 6.2140e-9*dElapsedJulianDays
			+0.0000396*cos(dOmega);
	}

	// Calculate celestial coordinates ( right ascension and declination ) in radians
	// but without limiting the angle to be less than 2*Pi (i.e., the result may be
	// greater than 2*Pi)
	{
		double dSin_EclipticLongitude;
		dSin_EclipticLongitude= sin( dEclipticLongitude );
		dY = cos( dEclipticObliquity ) * dSin_EclipticLongitude;
		dX = cos( dEclipticLongitude );
		dRightAscension = atan2( dY,dX );
		if( dRightAscension < 0.0 ) dRightAscension = dRightAscension + twopi;
		dDeclination = asin( sin( dEclipticObliquity )*dSin_EclipticLongitude );
	}
```

```
// Calculate local coordinates ( azimuth and zenith angle ) in degrees
{
        double dGreenwichMeanSiderealTime;
        double dLocalMeanSiderealTime;
        double dLatitudeInRadians;
        double dHourAngle;
        double dCos_Latitude;
        double dSin_Latitude;
        double dCos_HourAngle;
        double dParallax;
        dGreenwichMeanSiderealTime = 6.6974243242 +
                0.0657098283*dElapsedJulianDays
                + dDecimalHours;
        dLocalMeanSiderealTime = (dGreenwichMeanSiderealTime*15
                + udtLocation.dLongitude)*rad;
        dHourAngle = dLocalMeanSiderealTime - dRightAscension;
        dLatitudeInRadians = udtLocation.dLatitude*rad;
        dCos_Latitude = cos( dLatitudeInRadians );
        dSin_Latitude = sin( dLatitudeInRadians );
        dCos_HourAngle= cos( dHourAngle );
        pudtSunCoordinates->dZenithAngle = (acos( dCos_Latitude*dCos_HourAngle
                *cos(dDeclination) + sin( dDeclination )*dSin_Latitude));
        dY = -sin( dHourAngle );
        dX = tan( dDeclination )*dCos_Latitude - dSin_Latitude*dCos_HourAngle;
        pudtSunCoordinates->dAzimuth = atan2( dY, dX );
        if ( pudtSunCoordinates->dAzimuth < 0.0 )
                pudtSunCoordinates->dAzimuth =
                pudtSunCoordinates->dAzimuth + twopi;
        pudtSunCoordinates->dAzimuth=
        pudtSunCoordinates->dAzimuth/rad;
        // Parallax Correction
        dParallax=(dEarthMeanRadius/dAstronomicalUnit)
                *sin(pudtSunCoordinates->dZenithAngle);
        pudtSunCoordinates->dZenithAngle=(pudtSunCoordinates->dZenithAngle
                + dParallax)/rad;
        }
```

REFERENCES

Archer C. B. (1980) Comments on 'Calculating the position of the sun'. *Solar Energy* **25**, 91.

Teoría de Astronomía, Camblor-Ordiz A. (Ed.), (1995). pp. 107–113, Colegio de Oficiales de la Marina Mercante Española, Madrid.

Cooper P. I. (1969) The absortion of radiation in solar stills. *Solar Energy* **12**, 333–346.

Practical Astronomy with your Calculator, (1992). 3rd edn, Duffett-Smith P. (Ed.), pp. 64–65, Cambridge University Press, Cambridge.

Fliegel H. F. and Van Flandern, T. C. (1968). Letter to the Editor of Communications of the ACM (CACM, volume 11, number 10, October 1968, p. 657).

Holland P. G. and Mayer I. (1988) On calculating the position of the Sun. *International Journal of Ambient Energy* **9**(1), 47–52.

Ilyas M. (1983) Solar position programs: refraction, sidereal time and Sunset/Sunrise. *Solar Energy* **31**(4), 437–438.

Ilyas M. (1984) Reply to comments by Dr. D. J. B. Pascoe. *Solar Energy* **34**, 206.

Iqbal M. (1983). *An Introduction to Solar Radiation*, Academic Press, New York.

Kambezidis H. D. and Papanikolaou N. S. (1990) Solar position and atmospheric refraction. *Solar Energy* **44**, 143.

Lamm L. O. (1981) A new analytic expression for the equation of time. *Solar Energy* **26**, 465.

Astronomical Algorithms, (1998). 2nd edn, Meeus J. (Ed.), pp. 105–108, Willmann-Bell, Richmond, VA.

Michalsky J. J. (1988) The astronomical almanac's algorithm for approximate solar position (1950–2050). *Solar Energy* **40**(3), 227–235.

Muir L. R. (1983) Comments on 'The effect of atmospheric refraction on the solar azimuth'. *Solar Energy* **30**, 295.

Pascoe D. J. B. (1984) Comments on 'Solar position programs: refraction, sidereal time and Sunset/Sunrise'. *Solar Energy* **34**, 205–206.

Pitman C. L. and Vant-Hull L. L. (1978). Errors in locating the Sun and their effect on solar intensity predictions. In: *Meeting of the American Section of the International Solar Energy Society*, Denver, 28 Aug 1978, pp. 701–706.

Spencer J. W. (1971). Fourier series representation of the position of the Sun. Search 2, No. 5.

Spencer J. W. (1989) Comments on 'The astronomical almanac's algorithm for approximate solar position (1950–2050)'. *Solar Energy* **42**(4), 353.

Swift L. W. (1976) Algorithm for solar radiation on mountain slopes. *Water Resour. Res.* **12**, 108–112.

The Astronomical Almanac, 1986 Edition (1985). U. S. Governement Printing Office, Washington DC.

The Astronomical Ephemeris (1961). Explanatory Supplement to The Astronomical Ephemeris and The American Ephemeris and Nautical Almanac, Her Majesty's Stationary Office, London, 1961, pg. 98.

United States Naval Observatory (1998). *Multiyear Interactive Computer Almanac 1990–2005. Version 1.5*, Willmann-Bell, Virginia.

Vant-Hull L. L. and Hildebrandt A. F. (1976) Solar thermal power system based on optical transmission. *Solar Energy* **18**, 31–39.

Walraven R. (1978) Calculating the position of the Sun. *Solar Energy* **20**, 393–397.

Walraven R. (1979) Erratum. *Solar Energy* **22**, 195.

Wilkinson B. J. (1981) An improved FORTRAN program for the rapid calculation of the solar position. *Solar Energy* **27**, 67–68.

Wilkinson B. J. (1983) The effect of atmospheric refraction on the solar azimuth. *Solar Energy* **30**, 295.

Zimmerman J. C. (1981). Sun-pointing programs and their accuracy. Sandia National Laboratories, SAND81-0761.