Implementation of the Sun Position Calculation in the PDC-1 Control Microprocessor

John A. Stallkamp
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Abstract:

The major portion of this paper presents the several computational approaches to providing the local azimuth and elevation angles of the sun as a function of local time and then the utilization of the most appropriate method in the PDC-1 microprocessor. The full algorithm, in FORTRAN form, is felt to be very useful in any kind or size of computer. It was used in the PDC-1 unit to generate efficient code for the microprocessor with its floating point arithmetic chip. The balance of the presentation consists of a brief discussion of the tracking requirements for PDC-1, the planetary motion equations from the first to the final version, and the local azimuth-elevation geometry.

Introduction, Result, Nomenclature:

THE PDC-1 (Parabolic Dish Concentrator-1) is the first of a planned sequence of concentrators for dish-electric applications. It was designed by General Electric and uses injection molding techniques with plastic reflecting surfaces. The reflecting structure is a load bearing, integral part of the structure. A start-stop, on-off, control system is used to drive the elevation over azimuth configuration. The PDC-1 unit was fabricated and errected at Edwards Test Station (ETS) by Ford Aerospace and Communications Corp. (FACC).

The body of this paper is a mathematical derivation that will be given in a line item format. A minimum of comment and connective verse is used; some of the material and comment from the oral presentation is omitted as not necessary or rearranged for better continuity here.

The abstract adequately says what was done and why. The end result is the FORTRAN code and is the ultimate, almost stand alone, useful output of this presentation. Ordinary mathematical and computer language is freely used and this should not cause difficulties. There is no hope for a standardized or consistent nomenclature or notation for this field as is admitted by no less than the "Explanatory Supplement for the Ephemeris" in Section 1G.

Derivation and Discussion:

1. Requirement: Provide local azimuth and elevation angles of the sun.

2. Input: Local latitude, longitude and date/time. Date/time is the elapsed time from a recent epoch, viz., January 0, 19xx at 0 hours UT (Universal Time). The year is selectable; the required data are given in the Astronomical Almanac.

3. Accuracy: 0.01 degree.

4. Initial Approach: Paper by Robert Walraven (1). Paper was followed by corrections in subsequent issue (2).

5. Microprocessor for PDC-1 Concentrator: Advanced Micro Devices Model Am 95/4006 with an Am 9511 APU (Arithmatic Processing Unit). The ephemeris calculation was to be done at each concentrator in floating point, 32 bit, arithmetic and converted to fixed point integer units, 4096 counts per 90 degrees, for use in the control algorithm.

6. Am 9511 APU:

Fixed-Floating point capability: Discussed in Item 5 above.

Stack I/O Operation: Tedious to code but thoroughly adequate.

All direct and inverse trigonometric available: Inverse functions slow, 5000 to 8000 cycles, compared with less than 200 for multiplication.

Error return available for bad operation request, e.g., divide by zero: Not used in initial code but should have been. Not needed in final version.

7. Decision Point: When delivered to JPL, at a minimum the microprocessor code needed to be modified because of lack of complete selection of quadrant for an Arctan operation and potential divisions by zero. Walraven's equations as finally given seemed to be more complicated than needed; several computational simplifications, combinations and omission of some small correction terms should be accomplishable. The planetary motion equations resulting from the new analysis were identical with those given in recent issues of the Astronomical Almanac.

8. Final Planetary Motion Equations from The Astronomical Alamanac:

GMST = Greenwich mean siderial time

from page B6: "... holds during 1981: on days of year d at t hours UT, GMST = 6.6383211 hours + 0.0657098235 d + 1.0073791 t"

From page C20: "SUN, 1981"
"...coordinates of the sun to a precision of 0.01 deg"
"d = ... day of year ... + fraction of day from 0 hours UT"
"Mean longitude of Sun ... L = 279.575 deg + 0.985647 d"
"Mean anomaly: g = 356.967 deg + 0.985600 d"
"Ecliptic longitude: λ = L deg + 1.916 sin g + 0.020 sin 2g"

The epoch numbers, 6.6383211 hours, 279.575 and 356.967 deg are for January 0, 1981 at 0 hours UT. These are the three numbers that are updated to change to a different epoch. All other numerical values are constant.

9. Ecliptic to Equatorial Coordinates:

RA = Right Ascension = equatorial longitude
DEC = Declination = equatorial latitude
ϵ = Obliquity of the Ecliptic
= 23.442 deg (1981) = 23.443 deg (1974)

The constants cos ϵ and sin ϵ are available and may be updated (need is questionable) from page C20 of the AA.

sin DEC = sin ϵ * sin λ
tan RA = cos ϵ * tan λ

Local Hour Angle = Greenwich Mean Sidereal Time - West Longitude - Right Ascension

NLHA = - NLA = Negative Local Hour Angle
= RA - GMST + West Longitude
= RA - LMST (Local Mean Sidereal Time)

Before looking at these equations, consider how the results are to be used in the final calculation of local azimuth and elevation.

10. Spherical Triangle Equations: See Figure 1.

When all the complementary angles are removed, the law of cosines gives the elevation angle, and then the law of sines gives the azimuth angle.

sin ELV = [sin LAT] * [sin DEC]
+ [cos LAT] * [cos DEC * cos NLHA]

sin AZM = [cos DEC * sin NLHA] [cos ELV]

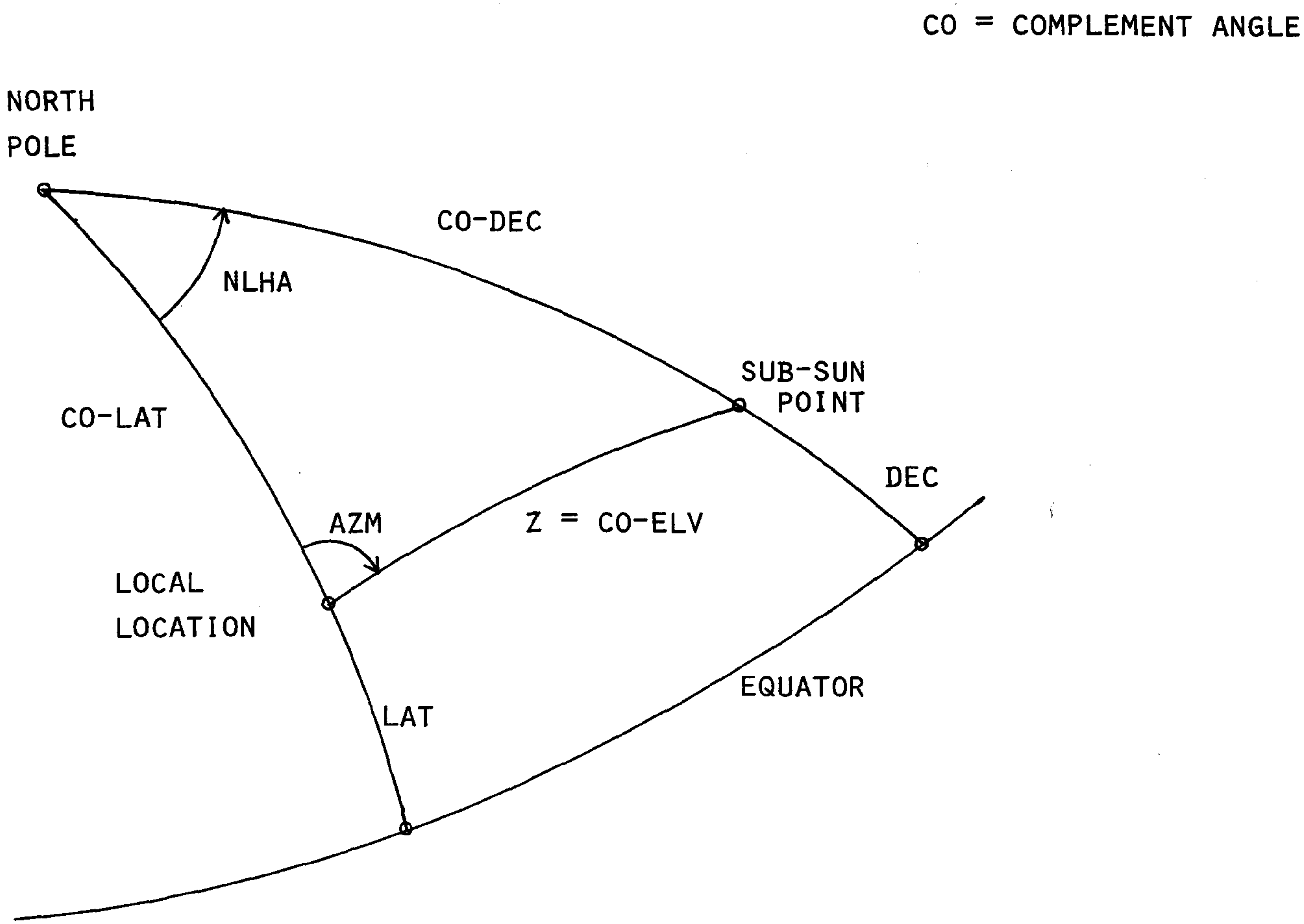

FIGURE 1 SPHERICAL TRIANGLE

11. Quadrant Selection:

$-90. \leq$ ELV $\leq +90.$ From sign of sin ELV equation

$0. \leq$ AZM $\leq 360.$ Two steps required.

```
IF(SIN(NLHA).GT.ZERO)  0. < AZM < 180.
IF(SIN(LAT) * SIN(ELV).LT.SIN(DEC)) 90. < AZM < 270. (Walraven's paper)
```

12. Bracketed Quantities in Item 10:

Numerical values of the terms in [ ] are required for the computations. The terms sin LAT and cos LAT are local constants. The angular measure of DEC and NLHA are not required nor is the numerical value of cos DEC if the bracketed numerical forms can be obtained otherwise.

13. Obtaining the Variable Input Terms of the Spherical Triangle Equations of Item 10 from the Equations of Item 9:

[sin DEC] = sin ϵ * sin λ is obtained directly.

The other bracketted variable terms in the equations of Item 10 are obtained from:

NLHA = RA - LMST
tan RA = cos ϵ * tan λ
RA and must be in the same quadrant

from which

sin RA = (cos ϵ * sin λ)/DENOM
cos RA = cos λ /DENOM
DENOM = + SQRT (cos$^2\epsilon$ * sin$^2\lambda$ + cos$^2\lambda$)

14. Some Available Computational Approaches for RA Leading to the Bracketted Terms:

School Boy: Do inverse tangent directly,
Select quadrant,
Continue,

Delta Angle: Let RA = λ + A,
Solve for A,
Continue,
(Expect automatic quadrant selection)

Trigonometric Sum Identity: NLHA = RA - LMST
sin NLHA = sin RA * cos LMST
-cos RA * sin LMST
(Expect avoidance of arc-trig operation and other simplifications)

15. School Boy Examples:

Large Computer:

```
SINRA = COS(ε) * SIN(λ)
COSRA = COS (λ)
   RA = ATAN2(SINRA, COSRA)
 NHLA = RA - LMST
```

The ATAN2 double argument input subroutine does the entire job and is not bothered by COSRA = 0.

Small Computer:

```
      SINRA and COSRA as above
      RA = ATAN(SINRA/COSRA)
or    RA = ATAN(COS(ε) * TAN(λ))
```

Now must protect from COSRA = 0 or large TAN(λ) and also must complete quadrant selection.

16. Delta Angle: With RA = λ + A expand sin RA and cos RA equations of Item 13.

$$\sin A = -\sin\lambda * \cos\lambda * (1-\cos\epsilon)/\text{DENOM}$$
$$\cos A = \cos^2\lambda * (1 - \tan^2\lambda * \cos\lambda)/\text{DENOM}$$

$$\tan A = -\tan\lambda * (1-\cos\epsilon) \Big/ (1 + \tan^2\lambda * \cos\epsilon)$$

This actually has excellent computational properties: A is small, sign (quadrant) is automatically selected, and, when tan λ is larger than the available computational range, set A to its obvious value of zero and skip ahead.

17. The Denominator: Expand DENOM of Item 13 and substitute for $\sin\epsilon * \sin\lambda = \sin \text{DEC}$.

$$\begin{aligned}
\text{DENOM}^2 &= \cos^2\epsilon * \sin^2\lambda + \cos^2\lambda \\
&= \cos^2\epsilon * (\sin^2\lambda * \sin^2\epsilon) \Big/ \sin^2\epsilon + 1 - (\sin^2\lambda * \sin^2\epsilon) \Big/ \sin^2\epsilon \\
&= 1 - \sin^2\text{DEC} * (1 - \cos^2\epsilon) \Big/ \sin^2\epsilon \\
&= 1 - \sin^2\text{DEC} = \cos^2\text{DEC}
\end{aligned}$$

$$\text{DENOM} = + \cos \text{DEC}$$

DENOM was defined as a positive quantity and the plus sign is appropriate since the absolute value of DEC is less than or equal to ϵ = 22.4 deg.

18. The Trigonometric Sum Identity: This can now be used to obtain the remaining bracketted terms in a direct manner. We now have:

$$\sin RA = (\cos \epsilon * \sin \lambda) / \cos DEC$$
$$\cos RA = \cos \lambda / \cos DEC$$

$$\cos DEC * \cos RA = \cos \epsilon * \sin \lambda$$
$$\cos DEC * \sin RA = \cos \lambda$$

$$[\cos DEC * \cos NLHA] = \cos DEC * \cos (RA - LMST)$$
$$= \cos \lambda * \cos LMST + \cos \epsilon * \sin \lambda * \sin LMST$$

$$[\cos DEC * \sin NLHA] = \cos DEC * \sin (RA - LMST)$$
$$= \cos \epsilon * \sin \lambda * \cos LMST - \cos \lambda * \sin LMST$$

19. PDC-1 Implementation: The trig sum expansions of Item 18 are used.

Do not need angular measure of RA or DEC or numerical measure of cos DEC.

Only arc-trig functions are at end for obtaining angular measure of elevation and azimuth. Quadrant selection discussed in Item 11.

Minimum number of direct trig operations, sin g , sin 2g , sin λ , cos λ , sin LMST and cos LMST.

No division by zero (no division in main part of program) or other dangerous steps requiring protection except again at end for obtaining the azimuth angle itself. In the PDC-1 microprocessor the few protective steps are done; these would be specific to each computer.

Large sections of the original Am 9511 floating point code went away. The FORTRAN code given in Table 1 is thought to be remarkably short, concise and easy to follow.

20. FORTRAN Code:

Table 1 is schematic FORTRAN code. The line "TIME =" is the start of the formal calculations; this code is full FORTRAN and is directly useable when statement numbers are supplied for the several GOTO commands. The lines preceeding the "TIME =" line provide for input constants and parameters; some of these lines will need to be modified to suit the particular computer being used. The quantities DN and TM are assumed to be supplied from other subroutines; suitable common statements for these and the AZM and ELV output angles must be supplied or other Input/Output provisions made.

References:

(1) Solar Energy, Vol. 20, pp. 393-397. Pergamon Press 1978. Printed in Great Britain.

(2) Solar Energy, Vol. 22, pp. 195. Pergamon Press 1979. Printed in Great Britain.

```
DN                      : DAY NUMBER SINCE EPOCH
TM                      : LOCAL STANDARD TIME, HOURS
PI = 3.14159
RAD = PI/180.
GMSTO = 6.622408        : EPOCH GREENWICH SIDEREAL TIME
MAO = 356.711           : EPOCH MEAN ANOMALY
SEMLO = 279.336         : EPOCH SUN ECLIPTIC MEAN LONGITUDE
COSEPS = 0.91747        : EPS=OBLIQUITY OF ECLIPTIC
SINEPS = 0.39781        :    = 22.4XX DEG
ECC1 = 1.916            : ELIPTIC ORBIT CORRECTION
ECC2 = 0.020            :    PARAMETERS
RATE = 0.985647         : SIDEREAL RATE
RATM = 0.985600         : MEAN ANOMALY RATE
TZ = 8.                 : TIME ZONE (PACIFIC HERE)
LAT = 34.992            : LOCAL LATITUDE DEGREES (ETS HERE)
LON = 117.873           : LOCAL LONGITUDE DEGREES (ETS HERE)
SINLAT = SIN(LAT * RAD)    : SUPPLIED OR CALCULATED
COSLAT = COS(LAT * RAD)    : SUPPLIED OR CALCULATED
TIME = DN + (TM + TZ) / 24.  :ELAPSED TIME IN DAYS
DELTA = RATE * TIME          :ELAPSED SIDEREAL ANGLE
MAR = RAD * (MAO + RATM * TIME)  :MEAN ANOMALY
SELD = SEMLO + DELTA + ECC1*SIN(MAR) + ECC2*SIN(2.*MAR)
SINRA = SIN(SELD * RAD)      :=SIN(SUN ECLIPTIC LON)
SINDEC = SINRA * SINEPS      :=SIN(DEC)
SINRA  = SINRA * COSEPS      :=SIN(RA) * COS(DEC)
COSRA = COS(SELD * RAD)      :=COS(RA) * COS(DEC)
GMSTD = 15. * (GMSTO + TM + TZ) + DELTA  :IN DEG
LMSTR = RAD * (GMSTD -LON)   :=NLHA = NEGATIVE LOCAL
SINLMS = SIN(LMSTR)          :  HOUR ANGLE IN RADIANS
COSLMS = COS(LMSTR)
SINELV = SINLAT*SINDEC+COSLAT*(COSRA*COSLMS+SINRA*SINLMS)
COSELV = (1.-SINELV**2.)     :=COS SQUARED HERE
IF(COSELV.LT.O)  GOTO (END)  :ERROR, ABANDON CALCULATION
COSELV = SQRT(COSELV)        :NOW IS COS(ELV)
ELV = ASIN(SINELV) / RAD     :=ELEVATION IN DEGREES
IF(COSELV.EQ.O)  GOTO (END)  :FINISHED, AZM NOT DEFINED
SINAZM = (SINRA*COSLMS-COSRA*SINLMS) / COSELV
COSAZM = (1.-SINAZM**2.)     :=COS SQUARED HERE
IF(COSAZM.LT.O)  GOTO (END)  :ERROR, ABANDON CALCULATION
AZMR = ASIN(SINAZM)          :=AZIMUTH IN RADIANS
QUAD = SINLAT*SINELV-SINDEC  :QUADRANT SELECTION
IF(QUAD.LT.O)  GOTO (+3)     :TO LINE IF(AZMR.GT.O)
AZMR = PI -AZMR
GOTO (+3)                    :TO LINE  AZM = AZMR / RAD
IF(AZMR.GT.O)  GOTO (+2)     :TO LINE  AZM = AZMR / RAD
AZMR = AZMR + 2. * PI
AZM = AZMR / RAD             :=AZIMUTH IN DEGREES
END
```

TABLE 1. SCHEMATIC FORTRAN