

太陽位置の計算

この文書は、拡張アメダス気象データ DVD に含まれる EZ DataNavi や EA DataNavi などのプログラム類で用いられる、太陽位置の計算方法について解説したものである。採用されている太陽位置 (太陽高度角と太陽方位角) の計算方法自体は、1 章に示す通り一般的なものであるが、その計算に必要となる赤緯と均時差の計算法として、①松本の式 (2 章参照) と②赤坂の式 (3 章参照) の 2 通りを採用している。

初版の拡張アメダス気象データ刊行以来、各種プログラム類では赤坂の式を用いてきたが、近年のコンピュータ環境の発達に鑑みて、計算手順は比較的長大ではあるものの、より高い精度に期待できる松本の式をデフォルトの計算方法とし、赤坂の式をオプションの計算方法と位置づけることになった^{注1}。ここでは特に、これら 2 つの方法を詳しく解説し、付録 E では計算精度の比較結果についても触れるが、最初の 1 章では太陽位置の計算方法について概説しておく。

1 太陽位置の計算

太陽の位置は、天文学的には、その「視赤緯 δ [°]」と「時角 t [°]」の 2 つのパラメータ (図 1 参照) で表せば事足りる。時角 t [°] は次式で計算するが、 E_t [°] で与える均時差の計算が肝要である。視赤緯 δ と均時差 E_t の計算方法については、2 章と 3 章で詳述する。

$$t = 15(T_m - 12) + (L - L_0) + E_t \quad (1)$$

ここに、 T_m : 計算対象とする標準時 [時]、 L_0 : 標準時を代表する地点の経度 [°]、
 L : 計算対象地点の経度 [°]、 E_t : 計算対象時の均時差 [°] である。

L_0, L は東経を正、西経を負とする。例えば、日本の場合、標準時を代表する地点は東経 135° の明石であるから、 $L_0 = 135$ となるが、仮に西経 135° であれば $L_0 = -135$ とすればよい。

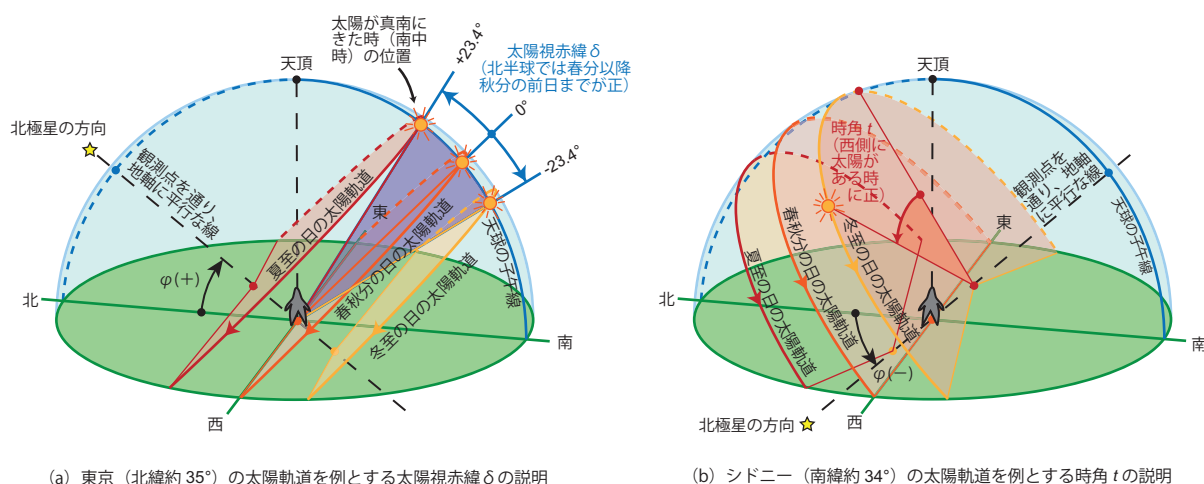


図 1 太陽の視赤緯 δ と時角 t の定義 (季節は北半球の概念による)

建築環境工学分野では、太陽位置は、計算対象地点の緯度 φ [°] も指定して、その地における高度角 h [°] と方位角 A [°] で表す (図 2)。 δ, t と h, A の関係は、球面三角法の公式により次のように表される[1]。

^{注1} 2013 年 5 月時点。EZ DataNavi Ver.1 や EA DataNavi Ver.4a / 4b 以降に適用される。計算精度の違いにより、日射量の直散分離などの結果が若干異なるため、旧版の拡張アメダス気象データプログラム類を用いて来た、古くからのユーザーには不都合な場合もある。そこで、赤坂の式をオプションとして残すことにした。

$$\sin h = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos t \quad (2)$$

$$\sin A = \frac{\cos \delta \sin t}{\cos h} \quad (3)$$

$$\cos A = \frac{\sin h \sin \varphi - \sin \delta}{\cos h \cos \varphi} \quad (4)$$

式 (2) の $\sin h$ が正であれば昼間であるが，負であれば日の出前あるいは日没後である。太陽方位角 A は，真南を 0° とし，西回りを正，東回りを負としている^{注2}。 $A(-180^\circ \leq A \leq 180^\circ)$ の値は $\sin A$ の符号に注意して決めるとよい。

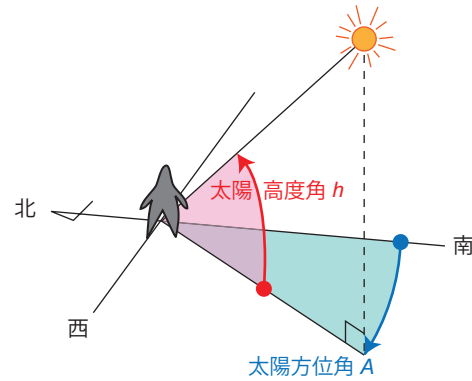


図2 太陽の高度角 h と方位角 A の定義

$$\text{すなわち, } \sin A > 0 \text{ のとき } A = 90 - \tan^{-1} \left(\frac{\cos A}{\sin A} \right) \quad (5)$$

$$\sin A < 0 \text{ のとき } A = -90 - \tan^{-1} \left(\frac{\cos A}{\sin A} \right) \quad (6)$$

以下の2つの章において，松本の式(文献2]~[5])および赤坂の式(文献7]~[10])による太陽視赤緯 δ と均時差 E_t の計算方法を示す。

2 視赤緯，均時差などの計算(その1)—松本の式による

2.1 計算式

日本を対象とする計算手順は以下の通りである(文献2]~[5])。ただし，計算年は1971年以降とする。

(1) 時刻補正項 ΔT_1 [s] の計算に用いるパラメータ T_u [-] の計算

計算対象年7月1日の正午(日本標準時12JST, Japan Standard Time)を協定世界時(Coordinated Universal Time)UTCまたは世界時UT1(=JST-9=3^h)で表した上で^{注3}，計算対象年月日と時刻UTC[時]を用いて，端数のあるユリウス日 JD_u を求める。さらに，これをJ2000.0(2000年1月1日12UTC)を元期とするユリウス世紀数 T_u [-] で表す。

$$T_u = \frac{JD_u - 2,451,545.0}{36,525} \quad (7)$$

なお，ユリウス日(Julian Day)は，BC4713年1月1日の正午(12UTC)から数えた経過日数のことで，直前の正午からの経過時間を小数として含む。その計算方法については，文献11]や2.2節のプログラムコードを参照されたい。

(2) 計算対象時刻JST[時]の地心座標時TCG[時]への変換

計算対象年月日のUTC[時]と地心座標時(Geocentric Coordinate Time)TCG[時]の間の時差

^{注2} 国際的には，太陽方位角は真北を 0° とし，西回りを正，東回りを負としている。式(2)~(4)による A の計算値を，国際基準の太陽方位角 A' に変換するには次のようにする。

$$A' = 180 - A \quad (0 \leq A \leq 180), \quad A' = -180 - A \quad (-180 \leq A \leq 0)$$

^{注3} 厳密にはUTCとUT1の間には時差があるが，1972年以降の補正により，その差が ± 0.9 s 以内になるように調整されている。したがって，1s以下の厳密さを求めない限り， $UTC(=JST-9^h) \approx UT1$ と考えてよい。

補正值 $\Delta T_1 (=3,600 \text{ (TCG - UTC)})$ [s] を式 (8) または式 (9) で小数点以下 3 桁 (ミリ秒) まで求め、計算対象時刻 JST [時] を地心座標時 TCG [時] で表す。

- 1971 年 1 月 1 日 ~ 2010 年 12 月 31 日の場合

$$\Delta T_1 = \frac{80^{\text{s}}.843\,08}{1 + 0.260\,5601 \exp(-4.423\,790 T_u)} - 0^{\text{s}}.311 \quad (8)$$

- 2011 年 1 月 1 日以降の場合

$$\begin{aligned} \Delta T_1 &= \frac{35^{\text{s}}.889\,50}{1 + 0.149\,455\,4 \exp(-9.796\,888 T_u)} + 32^{\text{s}}.184 \\ &+ \frac{86,400^{\text{s}}}{6.969\,290\,134 \times 10^{10}} \times (36,525 T_u + 8,611.999\,627\,5) \end{aligned} \quad (9)$$

$$\text{TCG} = \text{JST} - 9^{\text{h}} + \Delta T_1 / 3,600 \quad (10)$$

(3) 諸計算に用いるユリウス世紀数 T [-] の計算

計算対象年月日と地心座標時 TCG に対するユリウス日 JD_{TCG} を求め、これを J2000.0 を元期とするユリウス世紀数 T [-] で表す。式の形は (7) と同じである。

$$T = \frac{\text{JD}_{\text{TCG}} - 2,451,545.0}{36,525} \quad (11)$$

(4) 太陽の視黄経 ψ [°], 真黄道傾斜角 ε [°] および平均太陽の赤経 α'_m [°] の計算

式 (11) の T をパラメータとして以下の式で計算する。ただし、式 (12) における係数 P_i, Q_i, R_i は、表 1 左欄の値を用いる。

$$\begin{aligned} \psi &= \sum_{i=1}^{18(i \neq 16)} P_i \cos(Q_i T + R_i) + P_{16} T \cos(Q_{16} T + R_{16}) \\ &+ 36,000^{\circ}.769\,5 T + 280^{\circ}.460\,2 \end{aligned} \quad (12)$$

$$\begin{aligned} \varepsilon &= -23^{\circ}26'21''.448 + 46''.815\,0 T + 0''.000\,59 T^2 - 0''.001\,813 T^3 \\ &- 0^{\circ}.002\,56 \cos(1,934^{\circ} T + 235^{\circ}) - 0^{\circ}.000\,15 \cos(72,002^{\circ} T + 201^{\circ}) \end{aligned} \quad (13)$$

$$\begin{aligned} \alpha'_m &= 15^{\circ} \times \left(18^{\text{h}}41^{\text{m}}50^{\text{s}}.548\,41 + 8,640,184^{\text{s}}.812\,866 T \right. \\ &\left. + 0^{\text{s}}.093\,104 T^2 - 0^{\text{s}}.000\,006\,2 T^3 \right) \end{aligned} \quad (14)$$

(5) 視赤緯 δ [°], 均時差 E_t [°] などの計算

次式によって視赤緯 δ と均時差 E_t を求める。また、大気の外縁における法線面日射量 IN_0 [W/m^2] は、太陽定数を $J_0 (=1,367 \text{ W}/\text{m}^2)$ として、式 (17) で求められる。式中の r は地心距離 [-, AU] であり、その計算は式 (18) の通り、表 1 右欄の値をとる係数 P'_i, Q'_i, R'_i を用いた多項式による。

$$\delta = \tan^{-1} \left(\frac{\sin \psi \sin \varepsilon}{\sqrt{1 - \sin^2 \psi \sin^2 \varepsilon}} \right) \quad (15)$$

$$E_t = \left\{ \sum_{i=1}^{18} P_i \cos(Q_i T + R_i) - 0^{\circ}.005\,7 \right\} \cos \varepsilon + \tan^{-1} \left(\frac{\tan \alpha'_m - \tan \psi \cos \varepsilon}{1 + \tan \alpha'_m \tan \psi \cos \varepsilon} \right) \quad (16)$$

$$IN_0 = J_0 / r^2 \quad (17)$$

$$r = \sum_{i=1}^8 P'_i \cos(Q'_i T + R'_i) + P'_9 T \cos(Q'_9 T + R'_9) \quad (18)$$

表 1 視黄経 ψ および地心距離 r の近似計算式の係数

i	P_i	Q_i	R_i	P'_i	Q'_i	R'_i
1	+1.9147	35,999.05	267.52	+1.000140	0.00	0.00
2	+0.0200	71,998.10	265.10	+0.016706	35,999.05	177.53
3	+0.0020	32,964.00	158.00	+0.000139	71,998.00	175.00
4	+0.0018	19.00	159.00	+0.000031	445,267.00	298.00
5	+0.0018	445,267.00	208.00	+0.000016	32,964.00	68.00
6	+0.0015	45,038.00	254.00	+0.000016	45,038.00	164.00
7	+0.0013	22,519.00	352.00	+0.000005	22,519.00	233.00
8	+0.0007	65,929.00	45.00	+0.000005	33,718.00	226.00
9	+0.0007	3,035.00	110.00	-0.000042	35,999.00	178.00
10	+0.0007	9,038.00	64.00			
11	+0.0006	33,718.00	316.00			
12	+0.0005	155.00	118.00			
13	+0.0005	2,281.00	221.00			
14	+0.0004	29,930.00	48.00			
15	+0.0004	31,557.00	161.00			
16	-0.0048	35,999.00	268.00			
17	+0.0048	1,934.00	145.00			
18	-0.0004	72,002.00	111.00			

2.2 計算プログラムと計算例

2.2.1 太陽位置の計算プログラム（松本による C++コード）

C++言語による計算プログラムを以下に示す。太陽位置は、関数 SolPos(const SInputparams* aInput, const SDate* aDate, const double aTime, SOutputParams* aOutput) を呼び出して求めることができる。

```
#ifndef SolPosH
#define SolPosH
// SolPos.h - 太陽位置計算関数 SolPos その他の
// 関数群ヘッダファイル
// =====
#define YEAR_MIN (1971) // 計算対象年の下限
//
typedef unsigned int uint;
typedef unsigned long ulong;
//
// 日付構造体
typedef struct {
    uint YYYY; // 年 (1971 <= YYYY)
    uint MM; // 月 (1 <= MM <= 12)
    uint DD; // 日 (1 <= DD <= 28|29|30|31)
} SDate;
//
// 時刻構造体
typedef struct {
    uint hh; // 時 (0 <= hh <= 23)
    uint mm; // 分 (0 <= mm <= 59)
    double ss; // 秒 実数であることに注意
} STime;
//
// 太陽位置計算用入力パラメータ構造体
typedef struct {
    double Latit; // 計算対象地点の緯度 (deg.), +は北緯
    double Longit; // 計算対象地点の経度 (deg.), +は東経
    double RefLongit; // 計算対象地点の標準時経度 (deg.),
    // +は東経 (日本: 135.0)
} SInparams;
//
// 太陽位置計算結果 (出力) パラメータ構造体
typedef struct {
    double SinH, CosH; // 太陽高度角の正弦, 余弦
    double SinA, CosA; // 太陽方位角の正弦, 余弦
    double SinD, CosD; // 視赤緯の正弦, 余弦
    double Et, hAngle; // 均時差 (deg.) と時角 (deg.)
    double R; // 動径 (地心距離) (AU)
} SOutparams;
//
typedef const int cint;
typedef const double cdouble;
typedef const SDate cSDate;
typedef const STime cSTime;

typedef const SInparams cSInparams;
//
// 2000 年 1 月 1.5 日のユリウス日 (J2000.0)
const ulong JC2000 = 2451545UL;
//
// =====
// SolPos に関する関数群の宣言
// =====
//
void SetSTime( STime* aSTime, cdouble aHour = 0.0 );
// 実数で与えられた時刻 (aHour) (h) (デフォルト 0 時) を
// 用いて時刻構造体 (aSTime) を初期化する
// ただし, 0.0 <= aHour < 24.0
//
void STtoTD( cSDate* aDate, cSTime* aTime,
             cdouble RefTime,
             SDate* TDDate, STime* TDTTime );
// 標準時刻系で表された年月日 (aDate), 時刻 (aTime) を
// TCG 時刻系の年月日 (TDDate), 時刻 (TDTTime) に変換する
// 標準時と世界時の時差は, RefTime で与える (日本: +9.0)
//
ulong JD( cSDate& aDate );
// SDate 構造体 (aDate) で与えた年月日のユリウス日を
// 計算して, unsigned long 値として返す
//
inline double JC( cdouble aJD )
{ return ( aJD - JC2000 ) / 36525.0; };
// ユリウス日 (aJD) (実数であり, 時刻に応じた端数有効) を
// J2000.0 を元期とするユリウス世紀数の換算して返す
//
double MAscens( cdouble aUTJC );
// UT 時刻系の J2000.0 元期のユリウス世紀数 (aJC) に対する
// 平均赤経を返す (返値は deg. 単位)
//
long DT( cSDate* aDate, cdouble aUTTime );
// SDate 構造体 (aDate) で与えた年月日のある時刻
// aUTTime (UT 時系) の地心座標時 TCG を求める際の補正値
// (Delta T_1 = TCG - UTC) を返す (返値は msec. 単位)
//
double CLongit( cdouble aJC, const bool IsAp = true );
// J2000.0 元期のユリウス世紀数 (aJC) に対する太陽黄経を
// 返す IsAp が true (デフォルト) のとき視黄経,
// false のとき平均黄経 (返値は deg. 単位)
//
double Obliq( cdouble aJC, const bool IsPrC = true );
// J2000.0 元期のユリウス世紀数 (aJC) に対する黄道傾斜角
// を返す IsPrC が true (デフォルト) のとき章動を含み,
// false のときは章動を含まない平均黄道傾斜角とする
```

```
// (返値はdeg.単位)
// -----
double Eq( cdouble aJC );
// J2000.0元期のユリウス世紀数(aJC)に対する分点差を
// 返す(返値はdeg.単位)
// -----
double DecEq( cSDate* aDate, cSTime* aTime,
              cdouble aMascens,
              cdouble Decl, double& Et );
// TCG時刻系で年月日(aDate)と時刻(aTime),および
// 平均赤経(aMascens)(deg.)を与えて,
// (視)赤緯Decl(deg.)と均時差(deg.)を参照形式で返す
// また,返値として動径(地心距離)(AU)を返す
// -----
double SolRadius( cdouble aJC );
// J2000.0元期のユリウス世紀数(aJC)に対する
// 動径(地心距離)を返す(返値はAU単位)
// -----
void SolPos( cSInparams* aInput, cSDate* aDate,
             cdouble aTime, SOutparams* aOutput );
// 計算用入力パラメータ構造体(aInput)で与えた地点に対して
// 標準時刻系で与えた年月日(aDate)と時刻(aTime)における
// 太陽位置を計算し,結果を計算結果(出力)パラメータ構造体
// (aOutput)に入れて返す
// -----
#endif // end of flag SolPosH

// =====
// SolPos.cpp - 太陽位置計算関数 SolPos その他の関数群
// =====
#include <math.h>
#include "SolPos.h" // 前頁のヘッダーファイルをインクルード
// -----
cdouble _rpd = M_PI / 180.0;
cdouble ZERO = 1.0e-7; // ゼロと判断する基準値
// -----
void SolPos( cSInparams* aInput, cSDate* aDate,
             cdouble aTime, SOutparams* aOutput )
{ // 計算用入力パラメータ構造体(aInput)で与えた地点に対して
  // 標準時刻系で与えた年月日(aDate)と時刻(aTime)における
  // 太陽位置を計算し,結果を計算結果(出力)パラメータ構造体
  // (aOutput)に入れて返す
  SDate TDDate;
  STime STTime, TDTime;
  double RefTime = aInput->RefLongit / 15.0;
  SetSTime( &STTime, aTime );
  STtoTD( aDate, &STTime, RefTime, &TDDate, &TDTime );
  //
  double jday = (double)JD( aDate );
  jday += (aTime - RefTime - 12.0) / 24.0;
  double Tu = JC( jday );
  double Alpha_m = Mascens( Tu );
  //
  double Decl, Et, t, SinD, CosD, SinT, CosT;
  aOutput->R = DecEq( &TDDate, &TDTime, Alpha_m,
                    Decl, Et );
  //
  Decl *= _rpd;
  SinD = sin( Decl ); CosD = cos( Decl );
  aOutput->SinD = SinD; aOutput->CosD = CosD;
  aOutput->Et = t = Et;
  t += 15.0 * (aTime - 12.0)
    + (aInput->Longit - aInput->RefLongit);
  aOutput->hAngle = t;
  t *= _rpd;
  SinT = sin( t ); CosT = cos( t );
  //
  double phi = aInput->Latit * _rpd;
  double SinH, CosH, SinA, CosA;
  SinH = sin( phi ) * SinD + cos( phi ) * CosD * CosT;
  CosH = sqrt( 1.0 - SinH * SinH );
  if ( ZERO > fabs( CosH ) ) {
    // H = PI/2(90deg.), A = 0とみなす
    SinA = 0.0; CosA = 1.0; SinH = 1.0; CosH = 0.0;
  }
  else {
    SinA = CosD * SinT / CosH;
    CosA = sqrt( 1.0 - SinA * SinA );
    // ZERO Div.回避のため,式(3)は使わない
  }
  aOutput->SinA = SinA; aOutput->CosA = CosA;
  aOutput->SinH = SinH; aOutput->CosH = CosH;
  return;
}
// -----
// 内部使用の関数群の定義
// =====
```

```
bool IsLeap( cint aYear )
{ // 西暦年(aYear)がうるう年のとき true を,
  // うるう年でないとき false を返す
  if ( 1583 > aYear ) {
    // グレゴリオ暦採用(1582年10月15日)以前は
    // うるう年を考慮しない
    return ( false );
  }
  if ( 0 == (aYear % 400) ) {
    // 400年に一度のうるう年である
    return ( true );
  }
  int ymod = aYear % 100; // 100で除した余り
  if ( 0 == ymod ) {
    // YY00年はうるう年ではない
    return ( false );
  }
  return ( (0 == (ymod % 4)) ? true : false );
}
// -----
int SerialDay( cSDate* aDate )
{ // 日付構造体(aDate)で与えた年月日から1月1日を1とする
  // 通日を計算して返す(1-366)
  // 引数aDateが不正な場合,0を返す
  int DaysofMonth[12] = {
    31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, };

  if ( !aDate ) return ( 0 );
  int y = aDate->YYYY;
  int m = aDate->MM;
  int d = aDate->DD;
  if ( YEAR_MIN > y || 1 > m || 12 < m ||
        1 > d ) return ( 0 );
  DaysofMonth[1] += (IsLeap( y ) ? 1 : 0);
  if ( DaysofMonth[m - 1] < d ) return ( 0 );
  int sd = 0;
  for ( int mi = 1; mi < m; mi++ ) {
    sd += DaysofMonth[mi - 1];
  }
  return ( sd + d );
}
// -----
void SetSDate( SDate* aDate, cint aYear,
              cint aSerialDay )
{ // ある年(aYear)の通日(aSerialDay)を与えて,
  // 年月日を求め,日付構造体(aDate)に入れて返す
  // (デフォルトは2006年の通日1)
  int DaysofMonth[12] = {
    31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, };

  if ( !aDate ) return;

  int y = aYear;
  int sd = aSerialDay, d, m;
  while ( 0 >= sd ) {
    sd += 365 + (IsLeap( -y ) ? 1 : 0);
  };
  while ( 365 + (IsLeap( y ) ? 1 : 0) < sd ) {
    sd -= 365 + (IsLeap( ++y ) ? 1 : 0);
  };
  if ( YEAR_MIN > y ) { // デフォルトの年を使用する
    y = 2006;
  }
  DaysofMonth[1] += (IsLeap( y ) ? 1 : 0);
  d = m = 0;
  for ( int mi = 1; mi <= 12; mi++ ) {
    d += DaysofMonth[mi - 1];
    if ( d > sd ) {
      m = mi;
      d -= DaysofMonth[mi - 1];
      d = sd - d;
      break;
    }
  }
  aDate->YYYY = y; aDate->MM = m; aDate->DD = d;
  return;
}
// -----
// 公開する関数群の定義
// =====
void SetSTime( STime* aSTime, cdouble aHour )
{ // 実数で与えられた時刻(aHour)(h)(デフォルト0時)を
  // 用いて時刻構造体(aSTime)を初期化する
  // ただし,0.0 <= aHour < 24.0
  if ( !aSTime ) return;
  double h = aHour;

  if ( 0.0 > h ) {
    aSTime->hh = 0U; aSTime->mm = 0U; aSTime->ss = 0.0;
  }
  else if ( 24.0 <= h ) {
    aSTime->hh = 23U; aSTime->mm = 59U;
```

```

    aTime->ss = 60.0 - ZERO;
    return;
}
else {
    // 引数が正常な場合
    int hh = int(h);
    double res = 60.0 * (h - double(hh));
    int mm = int(res);
    res = 60.0 * (res - double(mm));
    aTime->hh = (uint)hh;
    aTime->mm = (uint)mm;
    aTime->ss = res;
}
return;
}
// -----
void STtoTD( cSDate* aDate, cSTime* aTime,
             cdouble aRefTime,
             SDate* TDDate, STime* TDTIME )
{ // 標準時刻系で表された年月日 (aDate), 時刻 (aTime) を
  // TCG 時刻系の年月日 (UTDate), 時刻 (UTIME) に変換する
  // 標準時と世界時の時差は, aRefTime で与える (日本: +9.0)
  if ( !aDate || !aTime || !TDDate || !TDTIME ) {
    return;
  }
  double r = aRefTime;
  if ( 12.0 < fabs( r ) ) {
    r = ((aRefTime > 0) ? 12.0 : -12.0);
  }
  double uth = aTime->ss / 3600.0 + aTime->mm / 60.0
    + aTime->hh - r;
  if ( 0.0 > uth ) {
    // 標準時->UT1 時変換で日付が前日になる場合
    int sd = SerialDay( aDate ) - 1;
    SetSDate( TDDate, aDate->YYYY, sd );
    uth += 24.0;
    SetSTime( TDTIME, uth );
  }
  else if ( 24.0 <= uth ) {
    // 標準時->UT1 時変換で日付が翌日になる場合
    int sd = SerialDay( aDate ) + 1;
    SetSDate( TDDate, aDate->YYYY, sd );
    uth -= 24.0;
    SetSTime( TDTIME, uth );
  }
  else {
    *TDDate = *aDate;
    SetSTime( TDTIME, uth );
  }
  // ここまでは, 標準時 -> UT1 時変換
  //
  // dt [sec] を標準時正午について求める
  double dt = (double)DT( aDate, 12.0 - r );
  // UT1 時を dt で補正
  double tdh = (dt * 0.001 + TDTIME->ss) / 3600.0
    + TDTIME->mm / 60.0 + TDTIME->hh;
  if ( 0.0 > tdh ) {
    // UT1 時->TD 時変換で日付が前日になる場合
    int sd = SerialDay( TDDate ) - 1;
    SetSDate( TDDate, TDDate->YYYY, sd );
    tdh += 24.0;
    SetSTime( TDTIME, tdh );
  }
  else if ( 24.0 <= tdh ) {
    // UT1 時->TD 時変換で日付が翌日になる場合
    int sd = SerialDay( TDDate ) + 1;
    SetSDate( TDDate, TDDate->YYYY, sd );
    tdh -= 24.0;
    SetSTime( TDTIME, tdh );
  }
  else {
    *TDDate = *aDate;
    SetSTime( TDTIME, tdh );
  }
  return;
}
// -----
// グレゴリオ暦採用年月日参照用マクロ
#define LGREG(Y,M,D) ((D)+31L*((M)+12L*(Y)))
#define GREGO LGREG(1582,10,15)
//
ulong JD( cSDate& aDate )
{ // SDate 構造体 (aDate) で与えた年月日のユリウス日を
  // 計算して, unsigned long 値として返す
  int ja, jm, jy = aDate.YYYY;
  long jul;
  long GREG1 = LGREG(aDate.YY, aDate.MM, aDate.DD);
  if ( 0 > jy ) ++jy;
  if ( 2 < int(aDate.MM) ) jm = (int)aDate.MM + 1;
  else {
    --jy; jm = (int)aDate.MM + 13;
  }
  jul = long(floor(365.25*jy) + floor(30.6001*jm)
    + aDate.DD) + 1720995L;
  if ( GREGO <= GREG1 ) {
    ja = int(0.01*jy);
    jul += 2 - ja + int(0.25*ja);
  }
  return ( ulong(jul) );
}
// -----
double MAscens( cdouble aUTJC )
{ // UT 時刻系の JC2000.0 元期のユリウス世紀数 (aJC) に対する
  // 平均赤経を返す (返値は deg.単位)
  static cdouble A[4] = { 67310.54841,
    8640184.812866, 0.093104, -0.0000062, };

  double Tu = aUTJC;
  double asec = ((A[3] * Tu + A[2]) * Tu + A[1]) * Tu
    + A[0]; // in [s]
  return ( asec / 240.0 ); // 15/3600=1/240[deg./s]
}
// -----
long DT( cSDate* aDate, cdouble aUTIME )
{ // SDate 構造体 (aDate) で与えた年月日のある時刻
  // aUTIME(UT 時系) の力学時を求める際の 補正值
  // (Delta T_1 = TD - UT1) を返す (返値は msec.単位)
  double jday = (double)JD( *aDate ) // 12UT1 のとき
    + (aUTIME - 12.0) / 24.0;

  double Tu = JC( jday );
  double dt = - 0.311; // in [s]
  double dp = 1.0 + 0.2605601 * exp( -4.423790 * Tu );
  dt += 80.84308 / dp;
  dt *= 1000.0; // in [ms]
  dt += 0.5; // 四捨五入のため
  return ( long(dt) );
}
// -----
typedef struct {
  bool T;
  double P, Q, R;
} SCoef;
#define PCOEF_MAX (18)
const SCoef PCoef[PCOEF_MAX] = {
  { false, 1.9147, 35999.05, 267.52, }, // 1
  { false, 0.0200, 71998.10, 265.10, }, // 2
  { false, 0.0020, 32964.00, 158.00, }, // 3
  { false, 0.0018, 19.00, 159.00, }, // 4
  { false, 0.0018, 445267.00, 208.00, }, // 5
  { false, 0.0015, 45038.00, 254.00, }, // 6
  { false, 0.0013, 22519.00, 352.00, }, // 7
  { false, 0.0007, 65929.00, 45.00, }, // 8
  { false, 0.0007, 3035.00, 110.00, }, // 9
  { false, 0.0007, 9038.00, 64.00, }, // 10
  { false, 0.0006, 33718.00, 316.00, }, // 11
  { false, 0.0005, 155.00, 118.00, }, // 12
  { false, 0.0005, 2281.00, 221.00, }, // 13
  { false, 0.0004, 29930.00, 48.00, }, // 14
  { false, 0.0004, 31557.00, 161.00, }, // 15
  { true, -0.0048, 35999.00, 268.00, }, // 16*
  { false, 0.0048, 1934.00, 145.00, }, // 17
  { false, -0.0004, 72002.00, 111.00, }, // 18
};
//
double CLongit( cdouble aJC, const bool IsAp )
{ // JD2000.0 元期のユリウス世紀数 (aJC) に対する太陽黄経を
  // 返す IsAp が true(デフォルト) のとき視黄経,
  // false のとき平均黄経 (返値は deg.単位)
  double T = aJC;
  double psi = 0.0, qtr, p;
  int icmax = (IsAP ? PCOEF_MAX : PCOEF_MAX - 2);

  for ( int ic = icmax; ic >= 1; ic-- ) {
    qtr = fmod( PCoef[ic - 1].Q * T
      + PCoef[ic - 1].R, 360.0 ) * _rpd;
    p = PCoef[ic - 1].P;
    if ( PCoef[ic - 1].T ) p *= T;
    psi += p * cos( qtr );
  }
  if ( IsAP ) psi -= 0.00569;
  psi += 36000.7695 * T + 280.4602;
  return ( psi );
}
// -----
double Oblit( cdouble aJC, const bool IsPrc )
{ // JD2000.0 元期のユリウス世紀数 (aJC) に対する黄道傾斜角
  // 返す IsPrc が true(デフォルト) のとき章動を含み,
  // false のときは章動を含まない平均黄道傾斜角とする
  // (返値は deg.単位)
  static cdouble E[4] = { -89381.448000, 46.815000,
    0.000590, -0.001813, };

  double T = aJC;
  double eps = 0.0, qtr, epss;
  if ( IsPrc ) {
    qtr = fmod( 72002.0 * T + 201.0, 360.0 ) * _rpd;
    eps -= 0.00015 * cos( qtr );
  }

```

```

    qtr = fmod( 1934.0 * T + 235.0, 360.0 ) * _rpd;
    eps -= 0.00256 * cos( qtr );
}
epss = ((E[3] * T + E[2]) * T + E[1]) * T + E[0];
eps += epss / 3600.0; // in [deg.]
return ( eps );
}
// -----
#define RCDEF_MAX (9)
const SCoeff RCoef[RCDEF_MAX] = {
{ false, 1.000140, 0.00, 0.00, }, // 1
{ false, 0.016706, 35,999.05, 177.53, }, // 2
{ false, 0.000139, 71,998.00, 175.00, }, // 3
{ false, 0.000031, 445,267.00, 298.00, }, // 4
{ false, 0.000016, 32,964.00, 68.00, }, // 5
{ false, 0.000016, 45,038.00, 164.00, }, // 6
{ false, 0.000005, 22,519.00, 233.00, }, // 7
{ false, 0.000005, 33,718.00, 226.00, }, // 8
{ true, -0.000042, 35,999.00, 178.00, }, // 9*
};
//
double SolRad( cdouble aJC )
{ // JC2000.0 元期のユリウス世紀数 (aJC) に対する
// 動径 (地心距離) を返す (返値は AU 単位)
double T = aJC;
double r = 0.0, qtr, p;
int icmax = RCDEF_MAX;

for ( int ic = icmax; ic >= 2; ic-- ) {
    qtr = fmod( RCoef[ic - 1].Q * T
        + RCoef[ic - 1].R, 360.0 ) * _rpd;
    p = RCoef[ic - 1].P;
    if ( RCoef[ic - 1].T ) p *= T;
    r += p * cos( qtr );
}
r += RCoef[0].P;
return ( r );
}
// -----
double Eq( cdouble aJC )
{ // JD2000.0 元期のユリウス世紀数 (aJC) に対する分点差を返す
// (返値は deg. 単位)
double eps = Obliq( aJC, true );
    eps = fmod( eps, 360.0 ) * _rpd;
    double dps = 0.0, qtr, p;
    qtr = fmod( SCoeff[SCDEF_MAX - 1].Q * T
        + SCoeff[SCDEF_MAX - 1].R, 360.0 ) * _rpd;
    p = SCoeff[SCDEF_MAX - 1].P;
    dps += p * cos( qtr );
    qtr = fmod( SCoeff[SCDEF_MAX - 2].Q * T
        + SCoeff[SCDEF_MAX - 2].R, 360.0 ) * _rpd;
    p = SCoeff[SCDEF_MAX - 2].P;
    dps += p * cos( qtr ) - 0.00569;
    return ( dps * cos( eps ) );
}
// -----
double DecEq( cDate* aDate, cTime* aTime,
    cdouble aMAscens,
    cdouble& Decl, double& Et )
{ // TD 時刻系で年月日 (aDate) と時刻 (aTime), および
// 平均赤経 (aMAscens) (deg.) を与えて,
// (視) 赤緯 Decl(deg.) と均時差 (deg.) を参照形式で返す
// また, 返値として動径 (地心距離) (AU) を返す
double jday = (double)JD( *aDate );
double hour = aTime->hh + aTime->mm / 60.0
    + aTime->ss / 3600.0;
jday += (hour - 12.0) / 24.0;
double T = JC( jday );
double eps = Obliq( T, true );
    eps = fmod( eps, 360.0 ) * _rpd;
    double psi = CLongit( T, true );
    psi = fmod( psi, 360.0 ) * _rpd;
    double eq = Eq( T );
    double sind = sin( psi ) * sin( eps );
    double cosd = sqrt( 1.0 - sind * sind );
    double tanam = tan( aMAscens * _rpd );
    double tpce = tan( psi ) * cos( eps );
    double dAsc = atan( (tanam - tpce)
        / (1.0 + tanam * tpce) ) / _rpd;
    Et = fmod( eq + dAsc, 360.0 );
    Decl = atan( sind / cosd ) / _rpd; // rad.->deg.
    return ( SolRadius( T ) );
}

```

2.2.2 均時差の計算例

理科年表 CD-ROM[12]に収録された 2003 年 1 月 1 日から 12 月 31 日までの 0UTC 時の均時差 $E_{t,m}$ [min.]^{注4} を真値と見なし, 松本の式 (式 (16)) による計算結果 (上に示したプログラムによる) と比較した結果を図 3 に示す。図によれば, 真値に対して計算結果は $\pm 0.6s$ 以内の差に収まっており, 妥当な計算結果が得られることが判る。

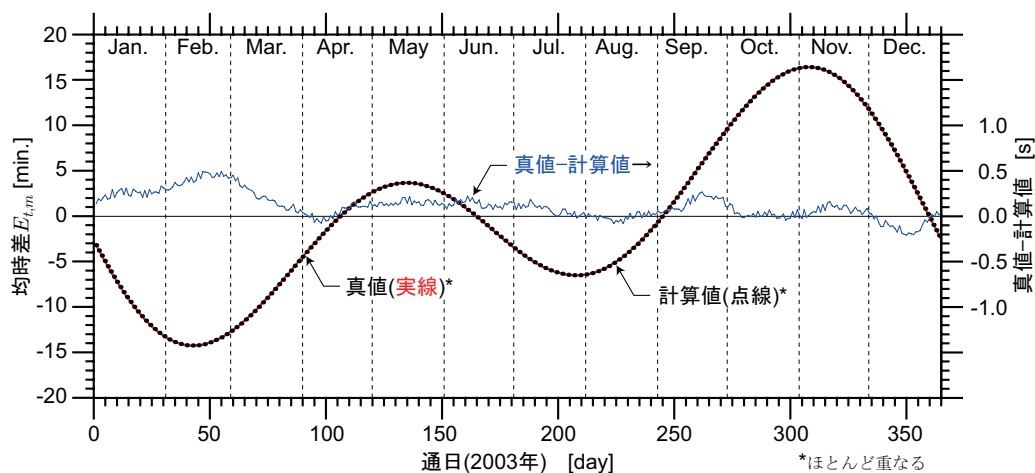


図 3 計算した均時差 $E_{t,m}$ [min.] の経時変化の真値との比較

注4 ここでは, E_t [°] を 4 倍 (=60min./15°) して分単位にした値を用いる。

3 赤緯と均時差の計算 (その 2)—赤坂の式による

3.1 計算式

赤坂の式によれば，太陽の視赤緯 δ [°] と均時差 E_t [°] は以下のように計算される (文献 7]~[10])^{注5}。

$$\sin \delta = \cos (v + \varepsilon) \sin \delta_0 \quad (19)$$

$$E_t = E_{t1} - E_{t2} \quad (20)$$

ここに， δ_0 : 北半球の冬至 (南半球の夏至) の日赤緯 ($=-23.4393^\circ$)

v : 真近点離角 [°]

ε : 近日点と冬至点の角度 [°]

E_{t1} : 中心差による時差 [°]

E_{t2} : 太陽赤経と太陽黄経の差 [°]

式 (20) における中心差による時差 E_{t1} は，1 日の地球の自転角 (約 361°) が公転軌道面上の地球の位置によって多少変化することにより生じる時差のことで，1 年周期である。また，太陽赤経と太陽黄経の差 E_{t2} は，地軸の傾きによって生じる差による時差で，半年周期である。

近日点と冬至点の角度 ε ，真近点離角 v ，時差 E_{t1} ， E_{t2} は次式で計算する。

$$\varepsilon = 12.3901 + 0.0172(n + \frac{M}{360}) \quad [^\circ] \quad (21)$$

$$v = M + 1.914 \sin M + 0.02 \sin (2M) \quad [^\circ] \quad (22)$$

$$E_{t1} = M - v \quad [^\circ] \quad (23)$$

$$\tan E_{t2} = \frac{0.043 \sin 2(v + \varepsilon)}{1 - 0.043 \cos 2(v + \varepsilon)} \quad (24)$$

式 (21) ~ (23) に現れる M [°] は平均近点離角で次式による。

$$M = 360 \frac{D - d_0}{365.2596} = 0.9856(D - d_0) \quad [^\circ] \quad (25)$$

$$d_0 = 3.71 + 0.2596n - \text{INT} \left[\frac{n+3}{4} \right] \quad [\text{日}] \quad (26)$$

$$n = \text{西暦年 (Y)} - 1968 \quad [\text{年}] \quad (27)$$

ここに，INT [] : 小数点以下切捨て

365.2596 : 近点年 (近日点基準の公転周期日数)

d_0 : 平均軌道上の近日点通過日

(暦表時による 1968 年 1 月 1 日正午基準の日差)

D : 1 月 1 日を初日 ($D = 1$) とする計算対象日の年間通日

大気の外縁における法線面日射量 IN_0 [W/m^2] は，太陽定数を J_0 ($= 1,367 \text{W}/\text{m}^2$) として，次式で求められる。

$$IN_0 = (1 + 0.033 \cos v) J_0 \quad (28)$$

なお，本節で用いている術語の定義は，3.3 節にまとめてあるので適宜参照されたい。

^{注5} 赤坂の式においては，視赤緯・均時差ともグリニッチ標準時の 0UTC(理科年表 13 参照，日本標準時の 9JST) の値である。

3.2 計算プログラム（赤坂による FORTRAN コード）

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C 太陽位置を求めるサブルーチン
C  PHI:計算対象地点の緯度(度),
C  LON:計算対象地点の経度(度)
C  TM:標準時(時),
C  LONS:標準時の地点の経度(度)(日本では135.0)
C  SINH:太陽高度のサイン, COSH:太陽高度のコサイン
C  SINA:太陽方位角のサイン, COSA:太陽方位角のコサイン
C  ET:均時差(度), T:時角(度)
C  SINDLT:赤緯のサイン, COSDLT:赤緯のコサイン
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  SUBROUTINE SUNLHA(PHI,LON,TM,LONS,SINDLT,COSDLT,
+  ET,SINH,COSH,SINA,COSA)
C  REAL PHI, LON, TM, LONS, SINDLT, COSDLT,
+  ET, SINH, COSH, SINA, COSA
C  PARAMETER (RAD=3.141592*2./360.)
C
C  T = 15. * (TM - 12.) + (LON - LONS) + ET
C  PHIRAD = PHI * RAD
C  TRAD = T * RAD
C
C  SINH = SIN(PHIRAD) * SINDLT
+  +COS(PHIRAD) * COSDLT * COS(TRAD)
C  COSH = SQRT(1.0 - SINH**2.0)
C  SINA = COSDLT*SIN(TRAD)/COSH
C  COSA = (SINH*SIN(PHIRAD)-SINDLT)
+  / (COSH*COS(PHIRAD))
C
C  END SUBROUTINE
C
C
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C 赤坂の式により, 赤緯と均時差を求めるサブルーチン
C  YEAR:西暦年, NDAY:年間通日
C  SINDLT:赤緯のサイン, COSDLT:赤緯のコサイン
C  ET:均時差(度)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  SUBROUTINE SUNLD(YEAR,NDAY,SINDLT,COSDLT,ET)
C  REAL YEAR, NDAY, SINDLT, COSDLT, ET
C  REAL M, DO, N
C  PARAMETER (RAD=3.141592*2./360.,
+  DLTO=-23.4393*RAD)
C
C  N = YEAR - 1968.
C
C  DO = 3.71 + 0.2596*N - INT((N+3.)/4.)
C
C  M = 0.9856 * (NDAY-DO)
C  EPS = 12.3901 + 0.0172 * (N + M/360.)
C  V = M + 1.914 * SIN(M*RAD)
+  + 0.02 * SIN(2.*M*RAD)
C  VEPS=(V+EPS)*RAD
C  VE2 = 2. * VEPS
C
C  ET = (M - V) -
+  + ATAN(0.043*SIN(VE2)/(1.-0.043*COS(VE2)))
+  /RAD
C
C  SINDLT = COS(VEPS) * SIN(DLTO)
C  COSDLT = SQRT(ABS(1. - SINDLT**2.))
C
C  END SUBROUTINE

```

3.3 用語の解説

本章で用いている術語の定義 [7], [14] を以下に示す。

- 1) 冬至点: 地球の公転軌道面における冬至の位置 (図 4)。
- 2) 近日点: 公転軌道面で太陽と地球が最も近づく位置 (図 4)。公転軌道上の冬至点を少し通過した位置にある。
- 3) 真近点離角 (v): 公転軌道面上で, ある日に, 近日点から公転方向に地球まで計った角度。
- 4) 平均近点離角 (M): 地球が公転軌道上を一定の角速度で移動すると仮定したとき, 近日点から地球まで計った角度。
- 5) 太陽赤経, 太陽黄経: それぞれ天球の赤道と黄道を基準面にとる赤道座標系と黄道座標系の経度である。天球の赤道と黄道は春分点と秋分点で交わっており, 黄道は赤道に対して約 23.5° 傾いている。太陽赤経, 太陽黄経は春分点から東回りの角距離で表す。
- 6) 均時差 (E_t): 図 5 において, 平均太陽とは天の赤道を 1 太陽年 (太陽が春分点を通過してから春分点に戻ってくるまでの時間) で一定速度で一周する仮想の天体である。平均太陽が子午線を通過してから次に子午線を通過するまでの時間が, 時計の 1 日の長さに対応する。視太陽と平均太陽のずれが均時差 E_t であり, 式 (23) に示す E_{t1} と式 (24) に示す E_{t2} の 2 つの項より成っている。視太陽は図 5 の黄道上を一定でない速度で移動しているが, E_{t1} は黄道上を一定速度で移動する仮想天体と視太陽とのずれを角距離で表したもので, E_{t2} はこの仮想天体と平均太陽の赤経のずれを角距離で表したものである。

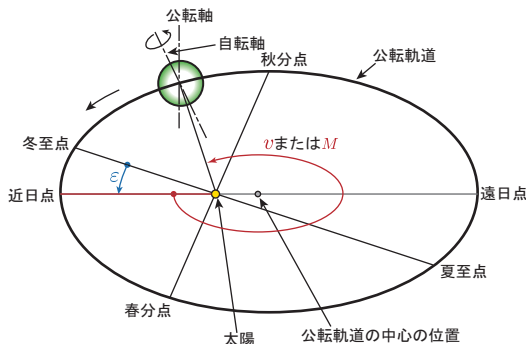


図 4 地球の公転と自転 [7]

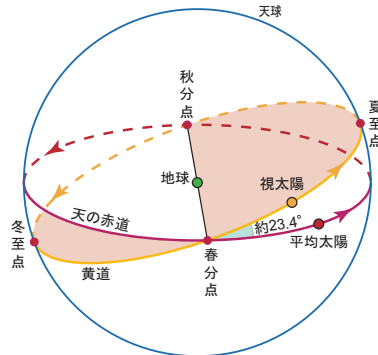


図 5 黄道と赤道 [14]

付録： 解説～松本の式による赤緯・均時差計算について

A 既往の計算式と松本の計算式提案の背景

A.1 既往の計算式

太陽の視赤緯 δ および式 (1) に現れる均時差 E_t の計算法はいくつかあるが、我が国の建築環境工学分野でよく知られているのは以下の 4 つであろう。この付録では、最後に列挙した松本の式 (文献 2]~[5]) について詳述する。

- 松尾・滝沢の式 [15] ある時期の理科年表に記載された値を調和解析により、一年の通日 Y の関数として表したもの。HASP などの熱負荷計算プログラムに組み込まれている。
- 山崎の式 [16] 位置天文学分野における Newcomb の理論を実用的な範囲で展開したもの。プログラムリストが公開されている [16]
- 赤坂の式 [7] 上述の山崎の式を簡略化し、地球の軌道要素の長期変動項を省略したもの。旧版の拡張アメダス気象データ [8]~[10]で提供されている斜面日射計算プログラムなどに組み込まれている。新しい拡張アメダス気象データにおけるプログラム類では、この式はオプションとして選択できる。
- 松本の式 [2]~[5] 海上保安庁水路部 (現 海洋情報部) による太陽系天体位置略算式 [17]に基づくもの。新版の拡張アメダス気象データ DVD で提供される斜面日射計算プログラムなどに、デフォルトの計算式として組み込まれている。

A.2 計算式提案の背景

3 章に示した赤坂の式 [7], あるいは建築環境工学分野でよく知られている山崎の式 [16]は基本的に、1895 年に発表された Newcomb の理論に基礎を置くものであり、以下のような問題がある。

- ① Newcomb の理論は、暦計算の元期を J1900.0 (1900 年 1 月 0.5 日) としている。しかし、1984 年より天文学の分野では、国際天文学連合 (IAU) の 1976 年決議に従い、元期を J2000.0 (2000 年 1 月 1.5 日) とし、さらには、Newcomb の理論とは異なる暦計算システム (理論) を採用しており ([18] pp.227~241), 理科年表 [13] もこれに従っている。また、新しく採用された暦計算システムにより、1984 年年初に春分点補正 (Newcomb の歳差式から Lieske の歳差式への移行があり、この時点で年々の春分点移動が不連続 ($0^s.063899$) になった ([18] pp.233~235)。

これらのことから、山崎の式や赤坂の式による δ と E_t の計算値と理科年表の記載値の間に乖離が懸念される。

- ② Newcomb は自論において時刻系を UT (Universal Time, 世界時) を用いて表したつもりであったが、今日では Newcomb の用いた時刻系は ET (Ephemeris Time, 暦表時) であると理解されている ([19] pp.79~80)。
しかし、松尾・滝沢の式 [15]はもちろん、赤坂の式でも UT と ET は区別して扱われていない。山崎の式では、UT から ET への変換を単純に扱っており^{注6}、この部分の誤差が精度に影響を及ぼすことが懸念される。
- ③ 1984 年以降、時刻系として TD (Dynamical Time, 力学時) を用いることになった。さらに、1991 年に TD は TT (Terrestrial Time, 地球時) として再定義され、今日では TT が浸透しつつある。しかし、②で述べたとおり、建築環境工学分野ではこのことに対応していない。

松本による太陽視赤緯 δ と均時差 E_t の計算式は、このような背景から提案されたものである。

^{注6} ET と UT の差が 1 年あたり 1s ずつ拡大するという設定で、 $ET = UT + 1^s \times (Y - 1930)$ としている。この式で $Y = 2000$ とすると西暦 2000 年には、ET と UT の差が 70s となるが、後述するように現実には約 64s である。

B 時刻系について

我々は日本標準時 (JST) に従っており, JST と協定世界時 (UTC) との間に 9h の時差があることは言うまでもない ($UTC = JST - 9^h$)。しかし, 前述のように, 天文学分野では現在, 時刻系として地球時 TT[時]を用いることになっており, 太陽位置計算の場合には, さらに TT と以下の関係にある TCG (Geocentric Coordinate Time, 地心座標時)[時]を用いるのがよい。

$$TCG = TT + \frac{24^h}{6.969\,290\,134 \times 10^{10}} \times (JD - 2,443,144.500\,372\,5) \quad (29)$$

ここに, JD は計算対象時の, 2,443,144.500 372 5 は 1977 年 1 月 1 日 0TAI のユリウス日である^{注7}。

太陽位置の計算においては, ①JST→時角変換や②その逆変換が必要となる。JST や UTC はいわゆる原子時系に属し, TT や TCG は力学時系に属する。その間をつなぐのがいわゆる恒星時系であり, 時角を扱うには恒星時系について知っておく必要がある。

B.1 恒星時系

B.1.1 世界時 UT1 と UT2

(1) 世界時 UT1 の定義

UT1 は, グリニッチ (Greenwich) における平均太陽時である。1984 年より, UT1 はグリニッチ平均恒星時 (Greenwich Mean Sidereal Time, GMST) $\bar{\Theta}_G[h]$ と次の関係があるものとして定義されている。

$$UT1 = \bar{\Theta}_G - \alpha_m + 12^h \quad (30)$$

ここに, $\alpha_m[h]$ は平均太陽の赤経で, 今考えている時 (UT1 時) のユリウス日 (Julian Day) を JD_u , それに対応する J2000.0 (2000 年 1 月 1 日 12UT1) を元期とするユリウス世紀数を式 (31) で表すとき, 式 (32) で定義される ([20], p.147)^{注8}。

$$T_u = \frac{JD_u - 2,451,545.0}{36,525} \quad (31)$$

$$\alpha_m = 18^h 41^m 50^s.548\,41 + 8,640,184^s.812\,866\,T_u + 0^s.093\,104\,T_u^2 - 0^s.000\,006\,2\,T_u^3 \quad (32)$$

なお, $INT[]$ を小数点以下切り捨ての関数として,

$$T' = \frac{INT[JD_u - 2,451,545.0] - 0.5}{36,525} \quad (33)$$

で定義される T' を用いて, 0UT1 におけるグリニッチ平均恒星時 $\bar{\Theta}_{G0}$ は,

$$\bar{\Theta}_{G0} = 6^h 41^m 50^s.548\,41 + 8,640,184^s.812\,866\,T' + 0^s.093\,104\,T'^2 - 0^s.000\,006\,2\,T'^3 \quad (34)$$

と表され ([18], pp.233–235), $\bar{\Theta}_G$ は, $\bar{\Theta}_{G0}$ と UT1 を用いて

$$\bar{\Theta}_G = \bar{\Theta}_{G0} + 1.002\,737\,909\,UT1 \quad (35)$$

となる ([19], pp.65–71)。

^{注7} TAI は国際原子時 (International Atomic Time) で, 後述の式 (39) の通り, TT と 32.184s の差がある。32^s.184/(3,600^s × 24) = 0.000 372 5 日であるから, 小数点以下 2 桁目以降の数値は 32.184s の補正分であることが判る。

^{注8} 厳密にいうと, FK5 星表による平均太陽の瞬時ににおける平均春分点を基準にして測った赤経のことである。

(2) UT1 と協定世界時 UTC の関係, UT1 と UT2 の関係

我々が通常用いている UTC と UT1 の間には $DUT1(=UT1 - UTC)$ と呼ぶ差があるが, UTC に対してうるう秒の補正を施すことにより, 1972 年以降, $DUT1$ が $\pm 0.9s$ 以内となるように調整されている。従って, $1s$ 以下の厳密さを求めない限り, $UT1 \approx UTC(= JST - 9^h)$ と考えてよい。

一方, $UT2$ は, $UT1$ に地球の自転速度変動のうち季節変化を加味したもので, その差 $\Delta T_s(= UT2 - UT1)$ は, $10ms$ オーダーの値である ([20], p.147)。

(3) UT1, $UT2$ と日本標準時 JST の関係

従って, $UT1 \approx UTC$ とする限り, $UT2 \approx UT1$ と考えてよい。すなわち,

$$UT2 \approx UT1 \approx UTC = JST - 9^h. \quad (36)$$

B.1.2 地方視恒星時 Θ とグリニッチ視恒星時 Θ_G

ある地点における (地方) 恒星時とは, その地における瞬時の真春分点^{注9}の時角のことで, $\Theta[h]$ で表される。特に, 地点を天文東経 $\lambda_s[h]$ がゼロのグリニッチに特定した恒星時をグリニッチ視恒星時 $\Theta_G[h]$ という。

天文東経 $\lambda_s[h]$ の地の (地方) 恒星時 Θ は, グリニッチ視恒星時 Θ_G を用いて $\Theta = \Theta_G + \lambda_s$ と表される。また, 通常東経 $\lambda[h]$ と天文東経 λ_s の間の差を $\Delta\lambda(= \lambda - \lambda_s)[h]$ とすると^{注10}, 次式で表される。

$$\Theta = \Theta_G + \lambda_s = \Theta_G + \lambda - \Delta\lambda \quad (37)$$

B.1.3 地方平均恒星時 $\bar{\Theta}$ とグリニッチ平均恒星時 $\bar{\Theta}_G$

ある地点における (地方) 平均恒星時とは, その地における瞬時の平均春分点の時角のことで, $\bar{\Theta}[h]$ で表される。地点をグリニッチに特定した場合の平均恒星時がグリニッチ平均恒星時 $\bar{\Theta}_G[h]$ である。 $\bar{\Theta}_G$ は式 (34) および式 (35) を用いて計算することができる。また, 天文東経 $\lambda_s[h]$, 東経 $\lambda[h]$ の地の (地方) 平均恒星時 $\bar{\Theta}$ は, グリニッチ平均恒星時 $\bar{\Theta}_G$ を用いて, 次式で表される。

$$\bar{\Theta} = \bar{\Theta}_G + \lambda_s = \bar{\Theta}_G + \lambda - \Delta\lambda \quad (38)$$

B.2 力学時系

B.2.1 暦表時 ET, 力学時 TD と地球時 TT

Newcomb が太陽位置の計算理論において UT1 のつもりで用いていた時刻系は, 現在では暦表時 ET であると考えられている ([19], pp.79–80)^{注11}。また後述するように, ET は TD と等しいと考えてよいと

^{注9} 春分点は変化しないものではなく, 地球の歳差運動と章動の影響を受けて変化する。歳差と章動の両方の影響の表れの結果としての春分点がある瞬間においていうとき, つまり視たままの春分点 (視差と光行差の影響も含める場合が多い) を与えるものを「瞬時の真春分点」といい, 歳差の影響だけを考慮した平均的な春分点を「瞬時の平均春分点」という。

^{注10} 地球の極運動の影響による経度 λ の変化に相当する補正で, 観測地の経緯度を $\varphi[^\circ]$, $\lambda[^\circ]$ とすると, IERS 基準極原点に準拠した瞬時の極位置座標 (単位は角度の $''$) (x, y) を用いて, $\Delta\lambda = -\frac{1}{3,600}(x \sin \lambda + y \cos \lambda) \tan(\varphi/15)$ と表される ([20], p.151)。

^{注11} ET は, 以下のように定義される (IAU 1958)。

① 1900 年初近くで太陽の幾何的平均黄経が $279^\circ 41' 48'' .04$ となった時刻を起点と考え, 1900 年 1 月 0 日 12ET と表す。
② 1 秒の長さを 1900 年 1 月 0 日 12ET における 1 回帰年の $1/31,556,925.9747$ とする。

されている ([17] p.144)。さらに、TT は TD を再定義したもので、TAI と次式で関係付けられている。

$$TT = TAI + 32^s.184/86,400^s \text{ [h]} \quad (39)$$

結局、ET、TD、TT は天文力学理論に合うように (再) 定義された時刻系であって、恒星時系である UT とは異なるものである。しかしながら、このことを明示的に考慮した計算式は、これまで建築環境工学の分野にはないように思われる。

B.2.2 地心座標時 TCG

先に TT は TD の再定義であると述べたが、この文脈でいう TD は、正確には地球力学時 TDT (Terrestrial Dynamical Time) のことであり、一方、太陽位置や暦の計算で用いてきた TD は、正確には太陽系力学時 TDB (Barycentric Dynamical Time) のことである。ただし、TT と TDB の差は最大でも 10ms 程度であり、多くの用途では無視できる。しかしながら、TT の再定義とともに、新たに地心座標時 TCG と太陽系座標時 TCB (Barycentric Coordinate Time) が導入された。TCG は TDB を置き換えたものであり、太陽位置計算の用途の場合、厳密には TT ではなく TCG を用いるのが適切であり、松本の計算式では基本的な時刻系として TCG を採用している。

(1) ΔT_1 (TCG と UTC の差) の定義 TT と TCG の関係は、式 (29) に示した通りであるが、この式の右辺第 2 項の小ささを考えると、それに掛かる JD は、UT1 あるいは UTC に基づくユリウス日 JD_u に置き換えても実用上十分であろう。 JD_u のユリウス世紀表現 $T_u[-]$ (式 (31) 参照) を用いると、式 (29) は、以下のよう書ける。

$$TCG = TT + \frac{24^h}{6.969\,290\,134 \times 10^{10}} \times (36,525T_u + 8,611.999\,627\,5) \quad (40)$$

さらに式 (39) より、

$$TCG = TAI + \frac{32^s.184}{3,600^s} + \frac{24^h}{6.969\,290\,134 \times 10^{10}} \times (36,525T_u + 8,611.999\,627\,5) \quad (41)$$

UTC と TAI は常に整数秒の差があるため、これを $\Delta T' = 3,600 \times (TAI - UTC)$ [s] と書けば、

$$TCG = UTC + \left[\frac{\Delta T' + 32^s.184}{3,600^s} + \frac{24^h}{6.969\,290\,134 \times 10^{10}} \times (36,525T_u + 8,611.999\,627\,5) \right] \quad (42)$$

と表される。ここで上式右辺の [] 内の 3,600 倍、すなわち秒表現を ΔT_1 [s] とおけば (式 (43))、TCG と UTC の関係は式 (29) のように簡単に表記できる。

$$\Delta T_1 = \Delta T' + 32^s.184 + \frac{86,400^s}{6.969\,290\,134 \times 10^{10}} \times (36,525T_u + 8,611.999\,627\,5) \quad (43)$$

$$TCG = UTC + \Delta T_1/3,600^s \text{ [h]} \quad (44)$$

したがって、式 (43) で定義される ΔT_1 [s] を用いると、地心座標時 TCG と日本標準時 JST との関係は以下のように表現できる (最右辺は式 (36) による)。

$$TCG = JST - 9^h + \frac{\Delta T_1}{3,600} \approx UT1 + \frac{\Delta T_1}{3,600} \quad (45)$$

(2) ΔT_1 の実際の値 ΔT_1 は現在では 60s を超えており [12]、無視できない。表 2 は、文献 [12] に示されている TD - UT2 の値と ΔT_s の計算式を元に ΔT_1 [s] を計算したものである。

表 2 地心座標時 TCG と協定世界時 UTC の差 ($\Delta T_1 = \text{TCG} - \text{UTC}$)

年月日 *	ユリウス日	T_u	$\Delta T_1 [\text{s}]$	$\Delta T' [\text{s}]$	年月日 *	ユリウス日	T_u	$\Delta T_1 [\text{s}]$	$\Delta T' [\text{s}]$
1962 7 1	2,437,846.5	-0.375 044 5	34.266		1982 7 1	2,445,151.5	-0.175 044 5	52.598	20.413
1963 7 1	2,438,211.5	-0.365 051 3	34.755		1983 7 1	2,445,516.5	-0.165 051 3	53.458	21.273
1964 7 1	2,438,577.5	-0.355 030 8	35.423		1984 7 1	2,445,882.5	-0.155 030 8	54.110	21.925
1965 7 1	2,438,942.5	-0.345 037 6	36.171		1985 7 1	2,446,247.5	-0.145 037 6	54.660	22.475
1966 7 1	2,439,307.5	-0.335 044 5	37.020		1986 7 1	2,446,612.5	-0.135 044 5	55.137	22.952
1967 7 1	2,439,672.5	-0.325 051 3	37.907		1987 7 1	2,446,977.5	-0.125 051 3	55.605	23.420
1968 7 1	2,440,038.5	-0.315 030 8	38.776		1988 7 1	2,447,343.5	-0.115 030 8	56.117	23.933
1969 7 1	2,440,403.5	-0.305 037 6	39.728	7.543	1989 7 1	2,447,708.5	-0.105 037 6	56.594	24.409
1970 7 1	2,440,768.5	-0.295 044 5	40.730	8.545	1990 7 1	2,448,073.5	-0.095 044 5	57.247	25.062
1971 7 1	2,441,133.5	-0.285 051 3	41.709	9.524	1991 7 1	2,448,438.5	-0.085 051 3	57.982	25.797
1972 7 1	2,441,499.5	-0.275 030 8	42.845	10.660	1992 7 1	2,448,804.5	-0.075 030 8	58.564	26.380
1973 7 1	2,441,864.5	-0.265 037 6	43.981	11.796	1993 7 1	2,449,169.5	-0.065 037 6	59.610	27.425
1974 7 1	2,442,229.5	-0.255 044 5	45.018	12.833	1994 7 1	2,449,534.5	-0.055 044 5	60.426	28.241
1975 7 1	2,442,594.5	-0.245 051 3	46.006	13.821	1995 7 1	2,449,899.5	-0.045 051 3	61.270	29.085
1976 7 1	2,442,960.5	-0.235 030 8	47.020	14.835	1996 7 1	2,450,265.5	-0.035 030 8	61.999	29.815
1977 7 1	2,443,325.5	-0.225 037 6	48.058	15.873	1997 7 1	2,450,630.5	-0.025 037 6	62.658	30.473
1978 7 1	2,443,690.5	-0.215 044 5	49.125	16.940	1998 7 1	2,450,995.5	-0.015 044 5	63.287	31.102
1979 7 1	2,444,055.5	-0.205 051 3	50.125	17.940	1999 7 1	2,451,360.5	-0.005 051 3	63.664	31.479
1980 7 1	2,444,421.5	-0.195 030 8	51.001	18.816	2000 7 1	2,451,726.5	0.004 969 2	63.980	31.796
1981 7 1	2,444,786.5	-0.185 037 6	51.837	19.652	*: 0UT1 時刻の値, T_u : ユリウス世紀数				

(3) ΔT_1 の採用値に関する提案 ΔT_1 は本来、観測によって確定する値であって、予測が困難であるとされている ([18], pp.125–126, [17], pp.144–146) が、太陽位置を予測計算する上で重要である。

松本は、時期によって以下のように場合分けをして得られる値を採用することを提案している。ただし、 ΔT_1 は 1 日ごとに一定の値を採用するものとする (小数点以下 3 桁まで)。

- 1800 年 1 月 1 日以前: $\Delta T_1 = 7.427\text{s}$ とする。
- 1800 年 1 月 1 日 ~ 1970 年 12 月 31 日: Schmadel and Zech の式による ΔT_1 を用いる。

$$\begin{aligned}
-\Delta T_1 = & 987^{\text{s}}.5520 + 20,781^{\text{s}}.6192 T_u + 176,498^{\text{s}}.5248 T_u^2 + 844,973^{\text{s}}.0784 T_u^3 \\
& + 2,557,073^{\text{s}}.9232 T_u^4 + 5,167,425^{\text{s}}.7152 T_u^5 + 7,169,822^{\text{s}}.6976 T_u^6 \\
& + 6,905,686^{\text{s}}.4928 T_u^7 + 4,601,064^{\text{s}}.3840 T_u^8 + 2,077,236^{\text{s}}.7488 T_u^9 \\
& + 605,853^{\text{s}}.7344 T_u^{10} + 102,926^{\text{s}}.6784 T_u^{11} + 7,732^{\text{s}}.0224 T_u^{12} \quad (46)
\end{aligned}$$

- 1971 年 1 月 1 日 ~ 2010 年 12 月 31 日: 以下の松本の提案式 I による ΔT_1 を用いる。

$$\Delta T_1 = \frac{80^{\text{s}}.84308}{1 + 0.2605601 \exp(-4.423790 T_u)} - 0^{\text{s}}.311 \quad (47)$$

- 2011 年 1 月 1 日以降: 以下の松本の提案式 II による ΔT_1 を用いる。

$$\begin{aligned}
\Delta T_1 = & \frac{35^{\text{s}}.88950}{1 + 0.1494554 \exp(-9.796888 T_u)} + 32^{\text{s}}.184 \\
& + \frac{86,400^{\text{s}}}{6.969290134 \times 10^{10}} \times (36,525 T_u + 8,611.9996275) \quad (48)
\end{aligned}$$

(4) ΔT_1 の提案式に関する補足

Schmadel and Zech の式 (46) Schmadel and Zech は 1800 年 ~ 1975 年の範囲で適用可能な ΔT_1 を与える実験式 (精度は、平均誤差 0.94s, 最大誤差 2.76s) を示している ([17], pp.144–146)。オリジナルの式は J1900.0 元期のユリウス世紀数によっているため、松本は、それを J2000.0 元期のユリウス世紀数 T_u (UT1 時系) に直し、式 (46) を提案している。

松本の提案式 I 式 (47) は表 2 に示したデータの回帰分析結果によるもので、それを補外して 2010 年末まで使うという提案である。分析に用いた 1962 年 ~ 2000 年の範囲における予測値の平均誤差は 0.650s, 最大誤差は 1.33s である。1s 以下の精度を問題にしないならば、式 (47) は十分実用的な予測式である。

松本の提案式 II 提案式 I による予測値と最近公表された観測結果の間に乖離があることが気になってきたため、提案式 I を再検討したのが、提案式 II である。

理科年表 CD-ROM (2003 年) [12] には、式 (44) で用いている $\Delta T'$ ($= \text{TAI} - \text{UTC}$) とは若干異なるが、 $\text{TAI} - \text{UT2}$ の実数データが 1969 年～2000 年の範囲で掲載されている。そこで、 UT2 に msec オーダーの補正^{注12}を施して UT1 (UTC と見なす) を推定した上で、 $\Delta T'$ の観測値と見なし、表 2 の $\Delta T'$ 欄のデータを得た。これに基づき、Levenberg-Marquardt 法による回帰分析を適用した結果が式 (48) である。最終的な ΔT_1 の推定値に対して、決定係数 $R^2=0.992$, $\text{MBE}=0.020\text{s}$, $\text{RMSE}=0.690\text{s}$ であり、1s 以下の精度を問題にしない実用的な立場からほぼ満足できる予測式である^{注13}。(図 6 に式 (47) と式 (48) による予測値と表 2 に示した観測値の関係を示した。)

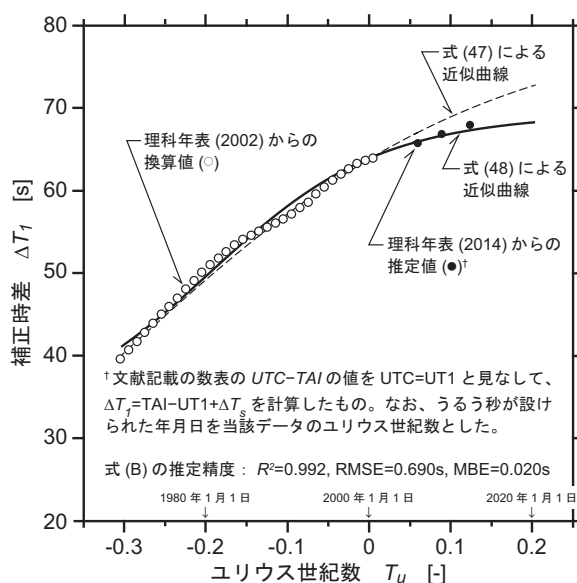


図 6 補正項 ΔT_1 の観測値と予測曲線

B.2.3 時刻系のまとめ

UT0 など述べていない時刻系もあるが、以上の時刻系相互の関係を図 7 にまとめて示す。図中、 $\Delta\lambda[\text{h}]$ は、地球の極運動の影響による経度 λ の変化に相当する補正である (p.12 の注 10 参照)。

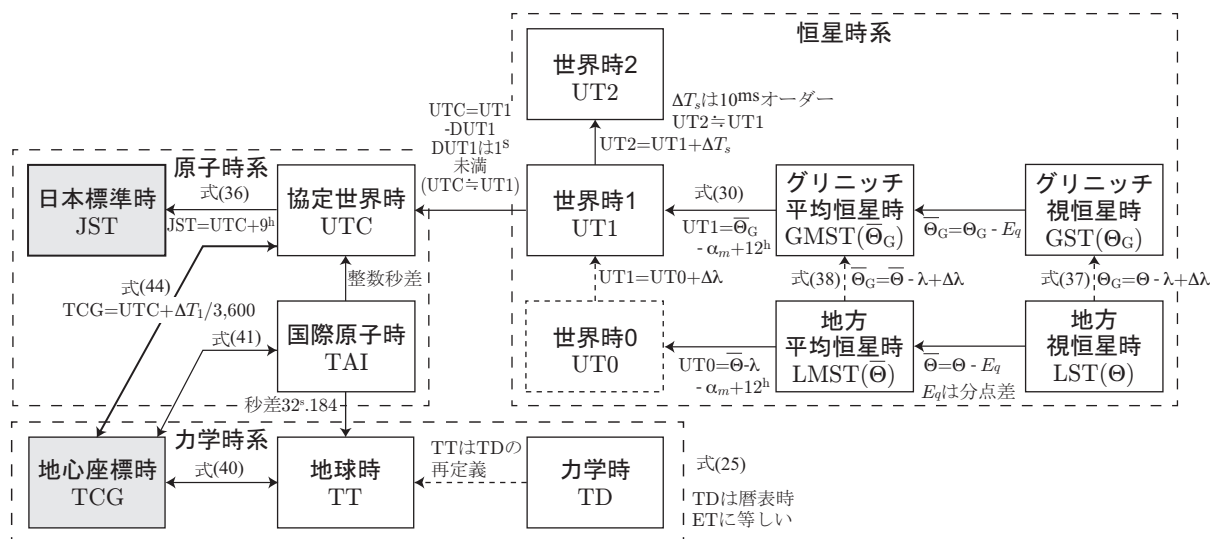


図 7 時刻系相互の関係

注13 t を太陽年で表した (周期的な) 時間とすると、以下のような慣習的な補正式 (出典不明) がある。

$$\Delta T_s = \text{UT2} - \text{UT1} = 0^{\text{s}}.0220 \sin(2\pi t) - 0^{\text{s}}.0120 \cos(2\pi t) - 0^{\text{s}}.0060 \sin(4\pi t) + 0^{\text{s}}.0070 \cos(4\pi t)$$

この式に基づき補正を施したが、値そのものは毎年 7 月 1 日に -0.029s 程度となる小数である。

注13 式 (48) を式 (47) の代わりに用いて、1971 年 1 月 1 日以降ならば全て式 (48) を用いることでもよいが、この式が提案されたのは最近であり [4] [5], 拡張アメダス気象データ (2013 年 4 月刊行) の操作プログラム類には既に式 (47) が適用されている。混乱を避けるため、このような使い分けを推奨している。式 (48) は次期バージョンの操作プログラムに適用される。

C 真太陽時 θ , 平均太陽時 $\bar{\theta}$ と均時差 E_t について

本章以降, 変数 T がしばしば用いられるが, これは J2000.0(2000 年 1 月 1 日 12TT) を元期とする (TT 時系) のユリウス世紀数のことであり, 前節の T_u とは区別すべきものであることに注意されたい。

C.1 真太陽時 θ と平均太陽時 $\bar{\theta}$

天文東経 λ の地点における地方真太陽時 (Local Apparent Solar Time) θ [h] は, その地の視太陽の時角で決定する時刻系で, 南中の瞬間を 12 時とし, 翌日の南中の瞬間までを 1 太陽日とするものであるから,

$$\theta = \Theta - \alpha + 12^{\text{h}} = \Theta_G + \lambda_s - \alpha + 12^{\text{h}} \quad (49)$$

と表される (第 3 辺は式 (37) 参照)。

恒星時が瞬時の平均春分点を基準とする (地方) 平均恒星時 $\bar{\Theta}$ [h] で表されるとき, 平均太陽の赤経 α_m を用いて, 平均太陽の時角は $\bar{\Theta} - \alpha_m$ であるから, 地方平均太陽時 (Local Mean Solar Time) $\bar{\theta}$ は,

$$\bar{\theta} = \bar{\Theta} - \alpha_m + 12^{\text{h}} = \bar{\Theta}_G + \lambda_s - \alpha_m + 12^{\text{h}} \quad (50)$$

で表される (第 3 辺は式 (38) 参照)。グリニッチにおける平均太陽時は UT1 そのものであり, 式 (30) で表されるから, 以下の式が成り立つ。

$$\bar{\theta} = \text{UT1} + \lambda_s \quad (51)$$

C.2 均時差 E_t

均時差 E_t [°] は, 「視太陽の時角から平均太陽の時角を減じたもの」 ([20] p.1) と定義されるから, それを時刻単位で T_e [h] と表すことにすると,

$$T_e = (\Theta - \alpha) - (\bar{\Theta} - \alpha_m) = (\Theta_G - \bar{\Theta}_G) + (\alpha_m - \alpha) = \theta - \bar{\theta} \quad (52)$$

となり, 当然のことながら, 地方真太陽時 θ から地方平均太陽時 $\bar{\theta}$ を減じたものに等しく, ある瞬間にはどの地点においても等しい。上式第 3 辺第 1 項の $(\Theta_G - \bar{\Theta}_G)$ を分点差 E_q [h] という。

$$E_q = \Theta_G - \bar{\Theta}_G \quad (53)$$

E_q の絶対値は, 最大 1s を少し超える程度である。 E_q を角度で表した e_q [°] は, 黄経の章動^{注14} $\Delta\psi$ [°], 黄道傾斜角の章動 $\Delta\epsilon$ [°] および平均黄道傾斜角 ϵ_m [°] が既知ならば,

$$e_q = \Delta\psi \cos(\epsilon_m + \Delta\epsilon) \quad (54)$$

と計算される。 $\Delta\psi$ および $\Delta\epsilon$ を求めるための章動の計算は現在, ‘1980 IAU Theory of Nutation’ と名付けられた Wahr の理論を用いて行うのが通例のようである ([18] pp.229–233)。

しかし, 海上保安庁水路部 (現 海洋情報部) は, J2000.0 を元期とするユリウス世紀数 T を変数とする

$$E_q = 0^{\text{h}}.000\,29 \sin(1,934^\circ T + 235^\circ) \quad (55)$$

^{注14} 地球の自転軸のいわゆる「みそすり運動」あるいは「コマの首振り運動」は一般によく知られている。このような大きな動き (歳差という) とは別に, 自転軸が向きを変える短周期の小さな運動があり, これを章動という。

に相当する近似式 ([17], p.150) を示している^{注15}。

分点差 E_q [h] を用いれば, 式 (52) は次のように書ける。

$$T_e = E_q + (\alpha_m - \alpha) \quad (56)$$

C.3 真太陽時 θ と UT1 の関係

今, T_e を既知とすれば, 真太陽時 θ と平均太陽時 $\bar{\theta}$ の関係は, $\theta = \bar{\theta} + T_e$ と書け, さらに式 (51) により UT1 と関係づけることもできる。ここで天文経度 λ_s [h] と経度 λ [h] の差 $\Delta\lambda$ [h] を無視すれば, θ と $\bar{\theta}$ の関係は次式で表される。

$$\left. \begin{aligned} \theta &= \bar{\theta} + T_e = \text{UT1} + \lambda + T_e \\ \text{UT1} &= \theta - \lambda - T_e \end{aligned} \right\} \quad (57)$$

D 視赤経 α , 視赤緯 δ と均時差 E_t の計算式

D.1 海上保安庁水路部の近似計算式

海上保安庁水路部 (現 海洋情報部) は, J2000.0 を元期とするユリウス世紀数 T を変数とする視赤経 α [h] と視赤緯 δ [°] の計算式を示している。計算式は瞬時の真春分点および瞬時の真赤道に準拠したもので, 精度は角度にして 6'' (時間にして 0.4s) であるという ([17], pp.148–157)。

D.1.1 視赤経 α [h] の計算式

[h] の単位で次式で表される。

$$\alpha = \sum_{i=1}^{27} A_i \cos(B_i T + C_i) + \sum_{i=28}^{32} A_i T \cos(B_i T + C_i) \quad (58)$$

係数 A_i , B_i , C_i ($i = 1, 2, \dots, 32$) は, 後掲の表 3 に示す値をとる。

D.1.2 視赤緯 δ [°] の計算式

[°] を単位として

$$\delta = \sum_{i=1}^{19} D_i \cos(E_i T + F_i) + \sum_{i=20}^{23} D_i T \cos(E_i T + F_i) \quad (59)$$

と表される。係数 D_i , E_i , F_i ($i = 1, 2, \dots, 23$) の値は, 表 3 に示すとおりである。

D.1.3 均時差 T_e [h] の計算

式 (56) において, E_q が式 (55) で, α が式 (58) でそれぞれ近似でき, α_m が式 (32) で定義されるのであるから, 均時差 T_e [h] は, 水路部による式のみで計算することができることになる。

^{注15} $\Delta\psi$ および $\Delta\epsilon$ の近似式も次のように示されている ([17], p.152)。

$$\begin{aligned} \Delta\psi &= 0^\circ.0048 \sin(1,934^\circ T + 235^\circ) - 0^\circ.0004 \sin(72,002^\circ T + 201^\circ) & (\text{式 (63) 参照}) \\ \Delta\epsilon &= 0^\circ.00256 \cos(1,934^\circ T + 235^\circ) - 0^\circ.00015 \cos(72,002^\circ T + 201^\circ) & (\text{式 (66) 参照}) \end{aligned}$$

D.1.4 その他の計算式

水路部は、視赤経 α と視赤緯 δ の計算式のほか、平均黄経 ψ_m [°] と地心距離 r [AU] の計算式を示している ([17] pp.148–157)。式中の係数 $P_i, Q_i, R_i (i = 1, 2, \dots, 16)$ 、および $P'_i, Q'_i, R'_i (i = 1, 2, \dots, 9)$ の値は、それぞれ表 3 と表 4 に示すとおりである。

$$\psi_m = \sum_{i=1}^{15} P_i \cos(Q_i T + R_i) + P_{16} T \cos(Q_{16} T + R_{16}) + 36,000.7695 T + 280.4659 \quad (60)$$

$$r = \sum_{i=1}^8 P'_i \cos(Q'_i T + R'_i) + P'_9 T \cos(Q'_9 T + R'_9) \quad (61)$$

式 (61) を用いれば、大気の外縁における法線面日射量 IN_0 [W/m²] は、太陽定数を J_0 (= 1367W/m²) として次式で求められる。

$$IN_0 = J_0 / r^2 \quad (62)$$

表 3 視赤経 α 、視赤緯 δ 、平均黄経 ψ_m 、視黄経 ψ の近似計算式の係数

i	A_i	B_i	C_i	D_i	E_i	F_i	P_i	Q_i	R_i
1	+18.69735	0.000	0.00	+23.2643	36,000.7696	190.4602	+1.9147	35,999.05	267.52
2	+0.16419	72,001.539	290.92	+0.3888	1.7200	12.9400	+0.0200	71,998.10	265.10
3	+0.12764	35,999.050	267.52	+0.3886	71,999.8200	187.9900	+0.0020	32,964.00	158.00
4	+0.00549	36,002.500	113.40	+0.1646	108,002.3000	211.4000	+0.0018	19.00	159.00
5	+0.00549	108,000.600	288.50	+0.0082	72,003.0000	34.0000	+0.0018	445,267.00	208.00
6	+0.00353	144,003.100	311.90	+0.0082	144,001.0000	209.0000	+0.0015	45,038.00	254.00
7	+0.00133	71,998.100	265.10	+0.0073	107,999.0000	186.0000	+0.0013	22,519.00	352.00
8	+0.00032	1,934.100	145.00	+0.0031	180,004.0000	232.0000	+0.0007	65,929.00	45.00
9	+0.00024	108,004.000	134.00	+0.0022	37,935.0000	65.0000	+0.0007	3,035.00	110.00
10	+0.00024	180,002.000	309.00	+0.0008	35,997.0000	345.0000	+0.0007	9,038.00	64.00
11	+0.00015	144,000.000	286.00	+0.0004	68,965.0000	78.0000	+0.0006	33,718.00	316.00
12	+0.00013	32,964.000	158.00	+0.0004	3,036.0000	123.0000	+0.0005	155.00	118.00
13	+0.00012	445,267.000	208.00	+0.0003	481,268.0000	128.0000	+0.0005	2,281.00	221.00
14	+0.00012	19.000	159.00	+0.0003	35,982.0000	121.0000	+0.0004	29,930.00	48.00
15	+0.00010	45,038.000	254.00	+0.0003	36,020.0000	80.0000	+0.0004	31,557.00	161.00
16	+0.00010	216,005.000	333.00	+0.0003	409,266.0000	287.0000	-0.0048	35,999.00	268.00
17	+0.00009	22,519.000	352.00	+0.0003	13,482.0000	293.0000	+0.0048	1,934.00	145.00
18	+0.00005	9,038.000	64.00	+0.0003	9,037.0000	332.0000	-0.0004	72,002.00	111.00
19	+0.00005	65,929.000	45.00	+0.0003	180,000.0000	206.0000			
20	+0.00005	3,035.000	110.00	-0.0127	36,001.0000	190.0000			
21	+0.00004	33,718.000	316.00	-0.0012	72,000.0000	188.0000			
22	+0.00003	155.000	118.00	-0.0012	2.0000	13.0000			
23	+0.00003	73,936.000	166.00	-0.0003	108,002.0000	211.0000			
24	+0.00003	3.000	296.00						
25	+0.00003	29,930.000	48.00						
26	+0.00003	2,281.000	221.00						
27	+0.00003	31,557.000	161.00						
28	+2,400.05130	0.000	0.00						
29	-0.00012	35,999.000	268.00						
30	-0.00019	72,002.000	291.00						
31	-0.00002	108,001.000	289.00						
32	-0.00002	36,003.000	113.00						

表 4 地心距離 r の近似計算式の係数

i	P'_i	Q'_i	R'_i
1	+1.000140	0.00	0.00
2	+0.016706	35,999.05	177.53
3	+0.000139	71,998.00	175.00
4	+0.000031	445,267.00	298.00
5	+0.000016	32,964.00	68.00
6	+0.000016	45,038.00	164.00
7	+0.000005	22,519.00	233.00
8	+0.000005	33,718.00	226.00
9	-0.000042	35,999.00	178.00

D.2 松本の提案式

前節に示した海上保安庁水路部の式による視赤緯 (式 (59)) および均時差 (式 (55), 式 (58) および式 (32)) の近似精度は, 次章で考察するように優れた精度を持ったものと評価できるが, ①表 3 のとおり, 全般的に膨大な係数を要する点, ②均時差を間接的に視赤経の近似値 (近似に用いる係数が特に多い) を用いて求めなければならない点で不満が残る。そこで松本は, 一部に水路部の式を取り入れた, 多少簡便な以下の諸式を提案している。

D.2.1 視黄経 ψ [°] と黄道傾斜角 ε [°] の計算式

前節で, その他の式として紹介した水路部による平均黄経の近似計算式 (60) に章動の影響 $\Delta\psi_1$ と光行差 $\Delta\psi_2$ を加味すれば, 視黄経 ψ [°] が得られる。水路部によれば,

$$\left. \begin{aligned} \Delta\psi_1 &= 0^\circ.0048 \cos(1,934^\circ T + 145^\circ) - 0^\circ.0004 \cos(72,002^\circ T + 111^\circ) \\ \Delta\psi_2 &= 0^\circ.00569 \end{aligned} \right\} \quad (63)$$

であるから ([17], pp.148–157),

$$\begin{aligned} \psi &= \psi_m + \Delta\psi_1 - \Delta\psi_2 \\ &= \sum_{i=1}^{15} P_i \cos(Q_i T + R_i) + P_{16} T \cos(Q_{16} T + R_{16}) + 36,000.7695 T + 280.4602 \\ &\quad + \sum_{i=17}^{18} P_i \cos(Q_i T + R_i) - 0^\circ.00569 \end{aligned} \quad (64)$$

と計算できる。係数 P_i , Q_i , R_i ($i = 1, 2, \dots, 18$) の値は, 前出の表 3 のとおりである。

一方, 平均黄道傾斜角 ε_m [°] は, Lieske らの式,

$$\varepsilon_m = -23^\circ 26' 21''.448 + 46''.8150 T + 0''.00059 T^2 - 0''.001813 T^3 \quad (65)$$

で計算できる ([18], pp.228–229)。黄道傾斜角の章動の影響項として水路部の式 ([17], pp.148–157) を採用すれば, 真黄道傾斜角 ε [°] は次式で表される。

$$\begin{aligned} \varepsilon &= -23^\circ 26' 21''.448 + 46''.8150 T + 0''.00059 T^2 - 0''.001813 T^3 \\ &\quad - 0^\circ.00256 \cos(1,934^\circ T + 235^\circ) - 0^\circ.00015 \cos(72,002^\circ T + 201^\circ) \end{aligned} \quad (66)$$

視赤経 α ([h] 単位を [°] 単位に変換したものを α' とする), 視赤緯 δ , 視黄経 ψ , 真黄道傾斜角 ε の間には以下の関係があるから, ψ と ε が既知のとき α と δ が計算できることになる。

$$\left. \begin{aligned} \cos \delta \cos \alpha' &= \cos \psi, \\ \cos \delta \sin \alpha' &= \sin \psi \cos \varepsilon, \\ \sin \delta &= \sin \psi \sin \varepsilon. \end{aligned} \right\} \quad (67)$$

D.2.2 視赤緯 δ [°] の計算式

式 (67) 第 3 式のとおり, $\sin \delta = \sin \psi \sin \varepsilon$ であるから,

$$\delta = \tan^{-1} \left(\frac{\sin \delta}{\cos \delta} \right) = \tan^{-1} \left(\frac{\sin \psi \sin \varepsilon}{\sqrt{1 - \sin^2 \psi \sin^2 \varepsilon}} \right) \quad (68)$$

と, 式 (64) および式 (66) に基づく ψ と ε を用いて計算すればよい。

D.2.3 均時差 E_t [°] の計算式

式 (54) で示した分点差 e_q [°] は, $\varepsilon_m + \Delta\varepsilon = \varepsilon$, $\Delta\psi = \Delta\psi_1 - \Delta\psi_2$ であるから,

$$e_q = (\Delta\psi_1 - \Delta\psi_2) \cos \varepsilon \quad (69)$$

と書ける。

一方, 平均太陽の赤経 α'_m (α_m を [°] 単位に換算したもの) と視赤経 α' の差について, $\tan(\alpha'_m - \alpha') = \frac{\tan \alpha'_m - \tan \alpha'}{1 + \tan \alpha'_m \tan \alpha'}$ であるから, 式 (67) 第 1 式と第 2 式より $\tan \alpha' = \tan \psi \cos \varepsilon$ を代入すれば,

$$\alpha'_m - \alpha' = \tan^{-1} \left(\frac{\tan \alpha'_m - \tan \psi \cos \varepsilon}{1 + \tan \alpha'_m \tan \psi \cos \varepsilon} \right)$$

と計算できる。したがって, 均時差 E_t [°] は,

$$E_t = e_q + (\alpha'_m - \alpha') = (\Delta\psi_1 - \Delta\psi_2) \cos \varepsilon + \tan^{-1} \left(\frac{\tan \alpha'_m - \tan \psi \cos \varepsilon}{1 + \tan \alpha'_m \tan \psi \cos \varepsilon} \right) \quad (70)$$

のように書ける。

すなわち, 式 (32), 式 (63), 式 (64) および式 (66) に基づいて計算される α'_m , $\Delta\psi_1$, $\Delta\psi_2$, ψ , ε を右辺に用いれば, 均時差 E_t が求められる。

E 各種計算式による計算結果の比較

1 章で取り上げた, ①松尾・滝沢の式 [15], ②山崎の式 [16], ③赤坂の式 [7] (2 章で解説), ④松本の式 [2], [3] (3 章で解説) に加え, ⑤海上保安庁水路部の式 [17] による計算結果を比較した。

比較に際して参照したのは, 理科年表 CD-ROM [12] の 1974 年から 2003 年までの 30 年間の視赤緯 δ [°] と均時差 $E_{t,s}$ [s] のデータ (10,957 サンプル) である。このデータを真値と見なしたときの各計算式 (① ~ ⑤) の精度 (誤差 = |計算値 - 真値|) を表 5(a), (b) に示す。

表によれば, 松本の式による視赤緯と均時差の精度が比較的高いことが判る。計算の目的によって必要な精度は異なるであろうが, 松本の式は, より厳密な太陽位置計算が必要となる場面では有用な式といえる。

最終的に式 (2) ~ 式 (4) を適用して得られる太陽高度角 h と方位角 A について, 松本の式による結果を基準として比較した結果を表 5(c) に示す。太陽の視直径は 2,000" 程度 であるから, これらの比較値は太陽位置計算の目的 (必要精度) に応じて計算式を選択する目安になるであろう。

赤坂の式によるものと松本の式によるものは, 0.1° 程度の差が生じることが判る。EZ DataNavi Ver.1 および EA DataNavi Ver.4a/4b では, 太陽高度角と方位角は 0.1° の桁までユーザーファイルに保存される。したがって, これらのプログラムにおける太陽位置計算のオプション設定如何で, 高度角と方位角データの最後の桁が異なることがあり得る。それに伴い, 日射の直散分離データや 2 次データとして計算される斜面日射量データなどの計算結果も異なることになる。

もっとも, 日射量に関して差が生じる最大の要因は, 大気の外縁における法線面日射量 IN_0 [W/m²] の計算値の違いである。松本の式では式 (62) (p.18) が用いられ, 赤坂の式では式 (28) (p.8) が採用されており, 松本の式による場合の方が常に大きめに計算される。その差は最大で 1.8W/m² [=0.006MJ/(m²·h)] であり, 日射関係の計算結果に若干の影響を及ぼす。

図 8 は, 2000 年の各日 12 時 (日本標準時) における両式による IN_0 の計算結果を比較したものである。松本の方法では, 地軸の章動などの影響も考慮されていることも見てとれる。

表 5 各種の計算式による視赤緯 δ , 均時差 $E_{t,s}$ および太陽位置

(a) 視赤緯 δ の誤差 ["]

誤差 * の指標	松尾・ 滝沢の式	赤坂の式	山崎の式	海保・水 路部の式	松本の式
最大誤差 (XBE)	2079.0	33.0	19.0	8.0	4.0
平均誤差 (RMSE)	936.7	12.7	6.9	2.1	1.2

(b) 均時差 $E_{t,s}$ の誤差 [s]

誤差 * の指標	松尾・ 滝沢の式	赤坂の式	山崎の式	海保・水 路部の式	松本の式
最大誤差 (XBE)	39.80	3.10	2.50	0.60	0.60
平均誤差 (RMSE)	14.52	0.88	0.76	0.15	0.15

※: (誤差)=(計算値)-(真値), 真値は理科年表 CD-ROM[12]のデータ

(c) 太陽位置の最大偏差 ["]

太陽位置の パラメーター	松尾・ 滝沢の式	赤坂の式	山崎の式	海保・水 路部の式	松本の式
太陽高度角 h	2294.0	327.4	33.2	6.7	(基準)
太陽方位角 A	2730.0	404.6	148.0	7.6	(基準)

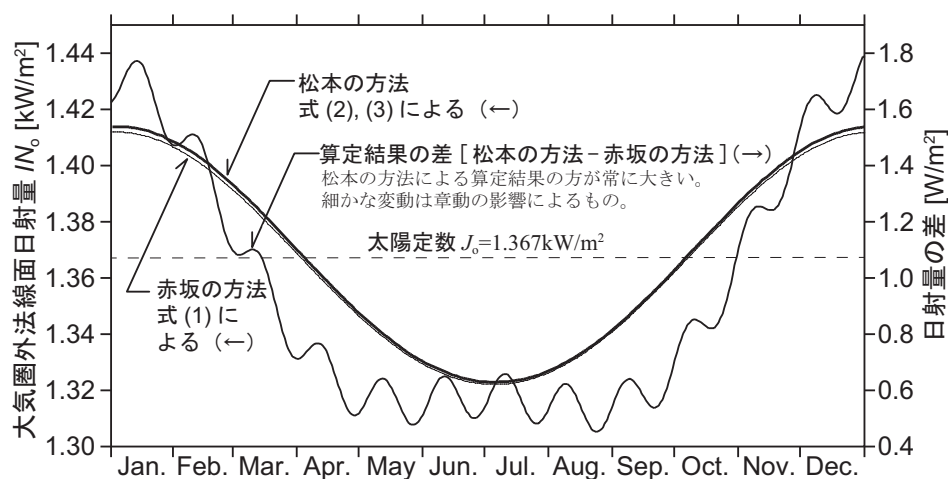


図 8 大気の外縁における法線面日射量の計算結果の差異 (2000 年各日の 12:00JST)

参考文献

- [1] 例えば, 小島武男: 日照, 『建築計画原論 I』, 渡辺 要 編 第 2 章, pp.120–128, 丸善 (東京), 1962.
- [2] 松本真一: 太陽視赤緯と均時差計算に関する一考察, 日本建築学会東北支部研究報告集, 計画系, 第 68 号, pp.89–96, 2005.6.
- [3] 松本真一: 太陽視赤緯と均時差の計算精度の検討, 日本建築学会大会学術講演梗概集, D-2 (環境工学 II), pp.7–8, 2006.8.
- [4] 松本真一: 太陽視赤緯と均時差の計算法に関する補遺, 日本建築学会東北支部研究報告集, 計画系, 第 77 号, pp.49–56, 2014.6.
- [5] 松本真一: 太陽視赤緯・均時差計算に関する筆者の方法の精度検証, 日本建築学会大会学術講演梗概集, D-2 (環境工学 II), pp.25–26, 2014.9.
- [6] Shin-ichi Matsumoto, Norio Igawa, Hideyo Nimiya, Hiroshi Akasaka, Kazuo Emura, Kazuhiro Takeda, Masaki Kubota: Recent Development Of Weather Data And Their Navigation Tools For Optimizing Building Environmental Design In Japan, Proc. of Building Simulation and Optimization (BSO2014), Second IBPSA - England Conf., Int'l Building Performance Simulation Assoc., pp.(N/A), 2014.6.
- [7] 赤坂 裕: 日射量の計算と日射直散分離・合成, 『建築環境学 1』木村建一 編 第 1 章, pp.13–17, 丸善 (東京), 1992.
- [8] 赤坂 裕他: 拡張アメダス気象データ, 日本建築学会 (丸善, 東京), 2000.
- [9] H. Akasaka et al.: Expanded AMeDAS Weather Data, Architectural Institute of Japan (Maruzen, Tokyo), 2003.
- [10] 赤坂 裕他: 拡張アメダス気象データ 1981–2000, 日本建築学会 (鹿児島 TLO, 鹿児島), 2005.
- [11] W.H. Press 他 (丹慶勝市他訳): Numerical Recipes in C(日本語版)–C 言語による数値計算のレシピ, 技術評論社 (東京), 1993.
- [12] 国立天文台 (編): 理科年表 CD-ROM 2003(平成 15 年版), 丸善 (東京), 2002.
- [13] 例えば, 国立天文台 (編): 理科年表 (平成 26 年 2014), 第 87 冊, 丸善 (東京), 2013.
- [14] 片山信夫: 地学精義 三訂版, 培風館 (東京), 1975.
- [15] 例えば, 斎藤平蔵: 建築気候, pp.121–127, 共立出版 (東京), 1974.
- [16] 山崎 均: 日照環境の基礎計算式 IV(対象地域の太陽視赤緯及び均時差を正確に計算するプログラム), 日本建築学会大会学術講演梗概集, 計画系, , pp.407–408, 1980.9.
- [17] 暦計算研究会 (編): 新こよみ便利帳 天文現象・暦計算のすべて, 恒星社厚生閣 (東京), 1991.
- [18] 長澤 工: 天体の位置計算 増補版, 地人書館 (東京), 1985.
- [19] 長谷川一郎: 天文計算入門 新装改訂版, 恒星社厚生閣 (東京), 1996.
- [20] 国立天文台 (編): 理科年表 (平成 17 年 2005), 第 76 冊, 丸善 (東京), 2004.