

チャレンジしよう日食計算 (2)

山本 威一郎

10. 接触時刻の求め方 (別法)

前号では、第1接触時刻から第4接触時刻の計算において、いくつかのQの値から階差を求め、逆補間公式を用いました。今回は、Bessel要素多項式(前号8項参照)をうまく利用し、階差を全く使用しないで接触時刻を求める方法について説明します。

接触時刻は $Q_i = 0$ になる t を求めることに集約されますから、これを Q_i の関数の根を求めることに置き換えて考えます。一般に連続関数の根を求めるには、

イ. はさみ打ち法

ロ. $x_i = f(x_i)$

ハ. Newton-Raphson法

等がありますが、ここでは収束の早いハの方法の変形を利用します。

任意の時刻 t における Q の値が、Bessel要素多項式を用いると計算出来ますから(前号①～⑦参照)、 Q は t の連続関数と見なせます。一般に、 $y = Q(t)$ の根の初期値を t_0 とすると、

$$t_1 = t_0 - \frac{Q(t_0)}{Q'(t_0)} \dots\dots\dots(14)$$

⋮

$$t_i = t_{i-1} - \frac{Q(t_{i-1})}{Q'(t_{i-1})} \dots\dots\dots(15)$$

$$|t_i - t_{i-1}| < \varepsilon \quad (\varepsilon: \text{微小一定値})$$

となる t_i を根としていますが、問題は $Q'(t_{i-1})$ を求めることです。そこで、近似的に以下の式で処理することにします。

$$Q'(t_{i-1}) \doteq \frac{Q(t_{i-1} + \Delta t) - Q(t_{i-1})}{\Delta t} \quad (\Delta t \doteq 0)$$

この式を用いると

$$t_i = t_{i-1} - \Delta t \left\{ \frac{Q(t_{i-1})}{Q(t_{i-1} + \Delta t) - Q(t_{i-1})} \right\} \dots\dots\dots(16)$$

ここで Δt をいくらに選ぶかが問題となりますが、Bessel要素多項式の各誤差が 10^{-5} 程度ですから、一応 $\Delta t = 10^{-4}$ 時

としておきましょう。この方法で、いろいろな接触時刻を計算してみました。一秒程度の精度で求まるようです。また、 t_0 として相当へだたえた値から始めても、10回以下のくり返して収束すると思います。これは、電卓とかコンピュータ向けの方法ですので、自動的に、第1～第4までの接触時刻を求まるには、 t_0 として、地球における食始めの世界時(日食図等から読みとれます)を選び、⑩式を用いて $|t_i - t_{i-1}| < 10^{-4}$ 時になったら、この t_i を第1接触時

とします。第4接触時刻の際の t_0 は、第1接触時刻に3時間を加えた値から始めれば良いでしょう。又、第2、第3接触時刻の際も同様な方法で求めますが、第2接触時刻から第3接触時刻を計算する時の初期値は、第2接触時刻に8分を加えた値とします。

11. 食分の求め方

食分は、太陽に対する月の侵入具合を示す量として、以下の式で定義されています。

$$D = \frac{BC}{AB}$$

$$= \frac{L_1 - m}{2L_1 - 0.5459}$$

.....⑰

これは略式ですが、0.01の位の精度では充分です。ただし⑰式において、

$$L_1 = \ell_1 - \zeta \tan f_1$$

$$m = \{ (x - \xi)^2 + (y - \eta)^2 \}^{1/2}$$

から求めます。

⑰式からも判りますが、第1及び第4接触時では、 $D = 0$ 、第2及び第3接触時では、 $D = 1$ 、皆既時では $D > 1$ となります。Dの大きさが、日食の大きさのひとつの目安となるわけです。

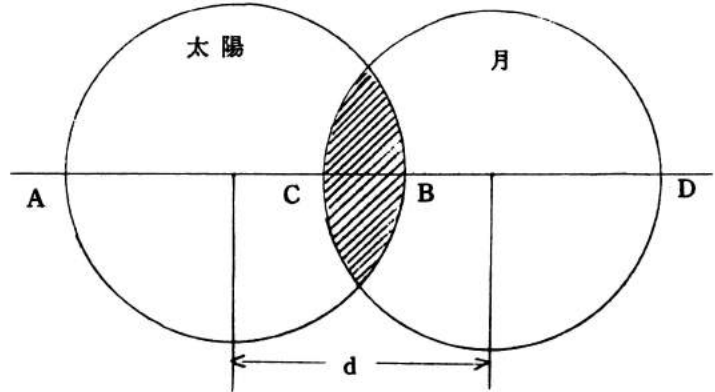


図 - 5

12. 月と太陽の視距離、月の視半径の求め方

天球上における月と太陽の視距離を d 、月の視半径を A とするとそれぞれの量は、次式から求めることができます。

$$d = \frac{m}{L_1 - 0.273} \text{⑱}$$

$$A = \frac{CD}{AB}$$

$$= \frac{0.5459}{2L_1 - 0.5459} \text{⑲}$$

⑨、⑩、⑪(前号)及び⑰、⑱、⑲から食の進行状況を図で示すことができます。図-6を参考にしてください。

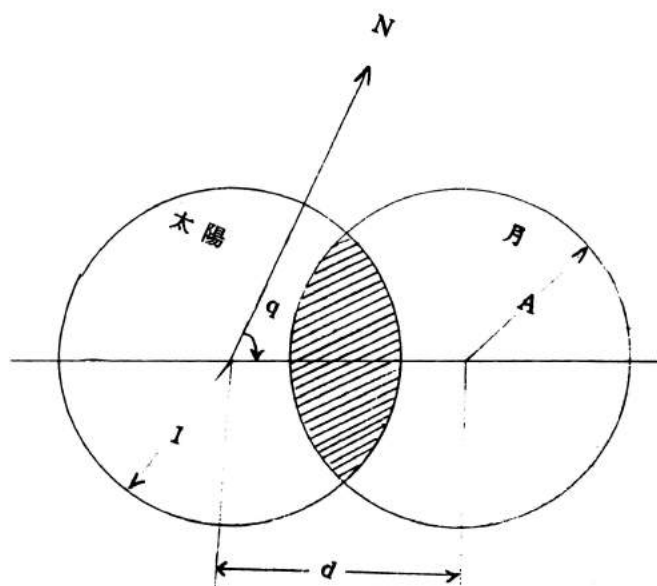


図 - 6

13. 電卓によるプログラミング例

近年の電子技術には著しい進歩があり、10年前には大型計算機を使わなければ解けなかった問題も、ちょっとした関数電卓で答が求まってしまうまでに到って来ました。パーソナルコンピュータとか高級プログラム電卓をお持ちの読者も多いと思いますので、今まで説明して来たアルゴリズムのプログラム例を示しておきます。

なお、著者はSHARP PC-1211(ポケットコンピュータ)のBASIC言語で組んでみました。この電卓はBASICで走らせることが出来ますが、処理速度が遅く、日食計算では、4つの時刻を求めるのに20分近くを要しました。

このプログラムを利用して公の場で発表される際は、筆者の承認を必要といたしますから、御注意願います。

14. 前号の一部訂正

前号に一部ミスがありましたので、修正される様願います。

- | | | | | | |
|------|---------|--------------------------------------|------|---------|--------------------|
| P 10 | 上から6行目； | 時間→時角 | P 10 | 下から7行目； | 1568h→1568h |
| P 11 | 図2が逆さ； | | P 11 | 図3 | ； ↑→↑ _g |
| | ⑥式 | ； tau _f →tan _f | | | |
| P 13 | ⑩、⑪の式 | ； C →小文字のc | | | |
| P 14 | 上から3行目； | h →H | | | |
| | ⑫式 | ； sinh→sinH | | | |

日食局地予報 for PC-1211

```

1 : 未使用
2 : " A " : CLEAR : I = 27          入力 ; 経度 ( ° )
3 : INPUT " LONG. = " ; R          入力 ; 経度 ( ° )
4 : INPUT " LAT. = " ; P           入力 ; 標高 ( m )
5 : INPUT " ALT. = " ; H
6 : S = 0.99497418 - 0.00167082 * COS(2 * P) + 0.0000021 * COS(4 * P) + 0.0000001568 * H
7 : C = 1.00167997 - 0.00168208 * COS(2 * P) + 0.00000212 * COS(4 * P) + 0.0000001568 * H
8 : INPUT " APPR. T = " ; D ; J = 0  入力 ; 第1接触概略世界時
9 : T = D : GOSUB 50
10 : 未使用
11 : A ( 31 ) = Q
12 : T = D + 0.0001
13 : GOSUB 50
14 : A ( 32 ) = Q : GOTO 70
15 : D = T - ( T - D ) * A ( 32 ) / ( A ( 32 ) - A ( 31 ) ) : BEEP 1 : PAUSE T, D
16 : IF ABS ( D - T + 0.0001 ) < 0.000001 GOTO 18
17 : GOTO 9
18 : A ( I ) = D : BEEP 3 : J = 0
19 : IF I = 27 LET I = I + 1 : D = D + 3 : GO TO 9
20 : IF I = 28 LET I = I + 1 : D = A ( 27 ) : GO TO 9
21 : IF I = 29 LET I = I + 1 : D = D + 8 / 60 : GO TO 9
22 : BEEP 4 : GOTO 30
23 : BEEP 5 : A ( 27 ) = DMS A ( 27 ) : PRINT " T1 = " ; A ( 27 )      出力 ; 第1接触時刻 ( U.T. 時. 分. 秒 )
24 : A ( 29 ) = DMS A ( 29 ) : PRINT " T2 = " ; A ( 29 )              出力 ; 第2 " ( " " " )
25 : D = DMS ( D ) : PRINT " TM = " ; D                             出力 ; 食最大時刻 ( " " " )

```

```

26 : A(30)=DMS A(30):PRINT "T3=";A(30)      出力;第3接触時刻(U. T. 時. 分. 秒)
27 : A(28)=DMS A(28):PRINT "T4=";A(28)      出力;第4      (      )
28 : END
29 : 未使用
30 : D=(A(29)+A(30))/2
31 : T=D
32 : GOSUB 50
33 : A(31)=Q
34 : T=D+0.0001
35 : GOSUB 50
36 : A(32)=Q
37 : A(33)=A(32)-A(31)
38 : T=D+0.0002
39 : GOSUB 50
40 : A(34)=Q
41 : A(35)=A(34)-A(32)
42 : D=(T-0.0002)-0.0001*A(33)/(A(35)-A(33)):BEEP 1
43 : IF ABS (D-T+0.0002)<0.000001 GOTO 23
44 : GOTO 31
45 : }
46 : }
47 : } 未使用
48 : }
49 : }
50 : X=-1.96874328+0.54790626*T+0.00005309*T*T-0.00000822*T*T*T

```

```

51 : Y=0.90489610-0.08798356*T-0.00015733*T*T+0.00000134*T*T*T
52 : A=0.31470985-0.00016630*T
53 : B=0.94918811+0.00005501*T
54 : M=178.4128692+15.0018573*T+0.0000021*T*T
55 : F=0.54236538+0.00020355*T-0.00001156*T*T
56 : O=-0.00396130+0.00020268*T-0.00001153*T*T
57 : G=C*COS(P)*SIN(M+R)
58 : E=S*SIN(P)*B-C*COS(P)*A*COS(M+R)
59 : Z=S*SIN(P)*A+C*COS(P)*B*COS(M+R)
60 : IF A(27)*A(28)=0 GOTO 62
61 : L=Q-Z*0.004583: GOTO 63
62 : L=F-Z*0.004606
63 : O=L*L-((X-G)*X-G)+(Y-E)*Y-E):RETURN
64 : }
65 : }
66 : } 未使用
67 : }
68 : }
69 : }
70 : K=A(32)-A(31)
71 : IF J*K<0 GOTO 73
72 : J=K:GOTO 15
73 : IF A(29)*A(30)=0 GOTO 75
74 : GOTO 23
75 : D=(A(27)+A(28))/2:GOTO 31

```