

## АЛГОЛ-ПРОГРАММА ДЛЯ ЭФЕМЕРИД СПУТНИКОВ

*Любомир-Александр Садовски*

Описанный алгоритм вычисления эфемерид искусственного спутника Земли (ИСЗ) пользуется модифицированными элементами орбиты [1]:

$E_0$  — эпоха элементов (по Мировому времени), в этот момент спутник находится в перигее (средняя аномалия  $M=0$ );

$i$  — наклонение орбиты к экватору;

$L_0$  — долгота восходящего узла к западу от Гринвича;

$\Delta J$  — величина, при помощи которой определяются средние сутки плоскости орбиты;

$\omega_0$  — аргумент перигея;

$\Delta\omega$  — движение перигея на один оборот;

$P_0$  — аномалистический период;

$\Delta P$  — изменение периода за период;

$e$  — эксцентриситет;

$r_p$  — геоцентрическое расстояние перигея;

$\alpha_Q$  — прямое восхождение восходящего узла.

Входными данными являются также название и кодовый номер спутника NrS, число наблюдательных станций  $n_{\text{obs}}$ , их кодовые номера, географические координаты — широта  $\varphi_{\text{obs}}$ , долгота  $\lambda_{\text{obs}}$  и геоцентрическое расстояние  $\rho_{\text{obs}}$ . При вычислениях нужно ввести еще несколько величин:

init — календарная дата начала вычислений;

fin — календарная дата конца вычислений;

$h_{\text{max}}$  и  $h_{\text{min}}$  — данные, ограничивающие снизу топоцентрическую высоту двух пунктов трассы спутника, когда оптические наблюдения возможны;

$l_{\odot}$  — средняя долгота Солнца на календарную дату эпохи орбитальных элементов;

$k$  — число суток после эпохи  $E_0$ .

Программа позволяет вычислить одновременно эфемериды для 30 наблюдательных станций при условии, что они находятся в районе с диаметром 1500 km. Ее можно применять к спутникам с любыми наклонениями орбиты, небольшими эксцентриситетами ( $e < 0,3$ ) и к северным географическим широтам станций.

Данная программа делает возможным получать эфемериды спутника в виде готовых телеграмм, составленных по коду „Spacewarn — SATAT“, для двух пунктов спутниковой трассы:

1) пункта наибольшего приближения к станции ( $U^*$ ) при условии, что высота спутника должна быть по крайней мере  $h_{\max}$ ;

2) пункта, через который спутник пройдет на несколько минут раньше, чем через пункт  $U^*$  (этот интервал времени определяется величиной  $\Delta v_0$ ) и высота спутника должна быть по крайней мере  $h_{\min}$ .

Большую часть вычислений занимает проверка условий видимости:

1. Спутник должен находиться над горизонтом станции.  
2. Солнце должно находиться ниже горизонта больше чем на  $10^\circ$ , так как оптические наблюдения возможны только после наступления сумерек.

3. Спутник должен освещаться Солнцем (не находиться в тени Земли).

Основы численного метода даются в [2] и [3]. Здесь существенно вычисление величин:  $n$  — номер оборота спутника, когда т. наз. относительный пункт [4] (с координатами  $\varphi_0, \lambda_0$ ) находится в орбитальной плоскости;  $T_n$  — момент пересечения спутником экватора Земли;  $u^{(2)}$  — аргумент широты и  $T^{(2)}$  — момент топоцентрической кульминации спутника; координаты  $\varphi_s, \lambda_s$  подспутниковых точек и, наконец, топоцентрические горизонтальные координаты спутника ( $A$  — азимут,  $h$  — высота).

Основные формулы следующие:

$$\lambda' = \arcsin \left( \frac{\operatorname{tg} \varphi_0}{\operatorname{tg} i} \right); \quad \Delta L = \frac{2880 \pi}{1440 + \Delta J};$$

$$N_0 = \frac{\lambda_0 + \lambda' - L_0}{\Delta L P_0}; \quad N' = \frac{1}{P_{\text{cp}}}; \quad N = N_0 + N'k;$$

$$n = [N] - 1, [N], [N] + 1;$$

$$\omega = \omega_0 + \Delta \omega n; \quad P = P_0 + \Delta P n;$$

$$T_n = E_0 + \left( n - \frac{\omega + v - M}{2\pi} \right) P;$$

$$L = L_0 + \Delta L (T_n - E_0);$$

$$\operatorname{tg} u^{(2)} = \frac{\sin (L - \lambda_{\text{obs}}) \cos i + \operatorname{tg} \varphi_{\text{obs}} \sin i}{\cos (L - \lambda_{\text{obs}})};$$

$$\varphi_s = \arcsin (\sin i \sin u); \quad \lambda_s = L - \arcsin (\cos i \operatorname{tg} u);$$

$$\sin \psi = \sin \varphi_s \sin \varphi_{\text{obs}} + \cos \varphi_s \cos \varphi_{\text{obs}} \cos (\lambda_{\text{obs}} - \lambda_s);$$

$$\cos A = \frac{\sin \varphi_s - \sin \varphi_{\text{obs}} \sin \psi}{\cos \varphi_{\text{obs}} \cos \psi}; \quad \operatorname{tg} h = \operatorname{tg} \psi - \frac{\varrho_{\text{obs}}}{r \cos \psi}.$$

Программа написана для электронно-вычислительной машины „GIER“ на универсальном языке „АЛГОЛ 60“ [5, 6] и дается в приложении вместе с примером. Она применялась для вычисления эфемерид спутников „Эхо 2“ (64041) и „Пагеос 1“ (66561) во время осеннего сеанса синхронных фотографических наблюдений спутников на станциях Познань (1154) и София (1101).

Выражаю искреннюю благодарность д-ру В. Пахельскому (Вычислительный центр Польской академии наук) за обсуждение метода и программы.

## ЛИТЕРАТУРА

1. Shapley A., COSPAR Inform. Bull., No. 24, 1965, 38.
2. Садовски Л.-А., Бюлл. СОН ИСЗ, № 46, Москва, 1965, 17.
3. Sadowsky L.-A., Beobachtungen künstlicher Erdsatelliten, Nr. 3, Berlin, 1965, 114.
4. Pachelski W., Computatio, 2, Warszawa, 1964, 5.
5. Backus J. W. and al., Revised Report on the Algorithmic Language ALGOL 60, Copenhagen, 1962.
6. Naur P. (ed.), A Manual of GIER ALGOL III, Copenhagen, 1965.

*Поступила 31. XII. 1966 г.*

```

The programme SATEPHCO [sator]
begin real Eo, incl, Lo, dL, omega0, domega, Po, dP, e, rp, So, lo,
    dvo, max, min, init, fin, pi, twopi, rad, dl, dS, R;
    integer NrS, year, dy, mn, nobS;
integer procedure DAY(D, M); value D, M; integer D, M;
begin integer m; for m:=1 step 1 until M-1 do
    D:=D+(if m=4  $\vee$  m=6  $\vee$  m=9  $\vee$  m=11 then 30 else if m=2 then 28
    else 31); DAY:=if year $\div$ 4 $\times$ 4=year  $\wedge$  M > 2 then D+1 else D
end DAY;

procedure RED (a, per); value per; real a, per;
L: if a < 0  $\vee$  a > per then begin a:=a-sign(a) $\times$ per; go to L end;

real procedure IN; begin real a; a:=inone;
    IN:=(inone+inone/60)/60 $\times$ sign(a)+a)rad end IN;

real procedure VtM(v, e); value v, e; real v, e;
begin real sinv, sin2v, sin3v, sin4v; sinv:=sin(v);
    sin2v:=sin(2 $\times$ v); sin3v:=sin(3 $\times$ v); sin4v:=sin(4 $\times$ v);
    VtM:=v-2 $\times$ e $\times$ sinv+(0.75 $\times$ e $\uparrow$ 2+0.125 $\times$ e $\uparrow$ 4) $\times$ sin2v-e $\uparrow$ 3 $\times$ sin3v/3+
    5 $\times$ e $\uparrow$ 4 $\times$ sin4v/32 end VtM;

procedure TnodeP(Eo, omega0, domega, e, n, Po, dP, P, Tn);
value Eo, omega0, domega, e, n, Po, dP; integer n;
real Eo, omega0, domega, e, P, Tn, Po, dP, begin real omega, v, eta;
    omega:=omega0+domega $\times$ n; v:=twopi-omega; eta:=v-VtM(v, e);
    P:=dP $\times$ n+Po; Tn:=(n-(omega+eta)/twopi) $\times$ P+Eo end TnodeP;

real procedure ARC(x, y); value x, y; real x, y; begin real arc;
    arc:=if abs(x) > abs(y) then arctan(y/x)+(if x < 0 then pi
    else if y  $\geq$  0 then 0 else twopi) else arctan(-x/y)+(if y  $\geq$  0
    then pi/2 else 3 $\times$ pi/2); ARC:=if arc > pi then arc-twopi
    else arc end ARC;

procedure Ahtop(v, rp, e, incl, u, OMEGA, fiobs, lobs, robs, r, fis, ls,
A, h); value v, rp, e, incl, u, OMEGA, fiobs, lobs, robs; real v, rp, e,
incl, u, OMEGA, fiobs, lobs, r, fis, ls, A, h; begin real psi;
    r:=rp $\times$ (1+e)/(1+e $\times$ cos(v)); fis:=arctan((sin(incl) $\times$ sin(u))/
    sqrt(1-(sin(incl) $\times$ sin(u)) $\uparrow$ 2)); if u=pi/2 then
    ls:=sign(cos(incl)) $\times$ pi/2 else ls:=arctan(cos(incl) $\times$ sin(u)/
    cos(u)); if u $\leq$ pi/2 then ls:=ls else ls:=pi+ls;
    ls:=OMEGA-ls; RED(ls, twopi); psi:=sin(fis) $\times$ sin(fiobs)+
    cos(fis) $\times$ cos(fiobs) $\times$ cos(lobs-ls); psi:=arctan(psi/sqrt
    (1-psi $\uparrow$ 2)); A:=sin(fis)-sin(fiobs) $\times$ sin(psi)/(cos(fiobs) $\times$ 
    cos(psi)); if abs(A) > 1 then begin A:=pi; h:=0 end
else begin A:=arctan(sqrt(1-A $\uparrow$ 2)/A); if A $\geq$ 0 then
    A:=if lobs $\geq$ ls  $\wedge$  lobs $\leq$ ls+pi  $\vee$  lobs < ls  $\wedge$  lobs < ls-pi
    then A else twopi-A else
    A:=if lobs $\geq$ ls  $\wedge$  lobs $\leq$ ls+pi  $\vee$  lobs < ls  $\wedge$  lobs < ls-pi
    then pi+A else pi-A;
    h:=arctan(sin(psi)/cos(psi)-robs/(r $\times$ cos(psi)))
end end Ahtop;

```

```

procedure twlgh(k, lo, fiop, lop, So Tbeg, Tend);
value k, lo, fiop, lop, So; integer k; real lo fiop, lop, So, Tbeg, Tend;
begin real l, delta, alfa, s, s1, s2; l:=k×dl+lo;
delta:=sin(0.4091698)×sin(1); delta:=arctan(delta/sqrt
(1-delta↑2)); alfa:=ARC(cos(1), cos(0.4091698)×sin(1));
s:=(-sin(10/rad)-sin(delta)×sin(fiop))/(cos(delta)×cos(fiop));
s:=arctan(sqrt(1-s↑2)/s); if s<0 then s:=pi+s;
s1:=alfa+s; s2:=alfa-s; s:=So+twopi×dS×k;
Tbeg:=(s1+lop-s)×0.99726957/twopi; RED(Tbeg, 1);
Tend:=(s2+lop-s)×0.99726957/twopi; RED(Tend, 1) end twlgh;

boolean procedure IISun(Eo, T, lo, dl, So, R, r, fis, ls);
value Eo, T, lo, dl, So, R, r, fis, ls; real Eo, T, lo, dl, So, R, r, fis, ls;
begin real l, delta, alfa, tS, sigma, sigma0;
l:=(T-entier(Eo))×dl+lo; delta:=sin(0.4091698)×sin(1);
delta:=arctan(delta/sqrt(1-delta↑2));
alfa:=ARC(cos(1), cos(0.4091698)×sin(1));
tS:=(T-entier(Eo))×twopi×1.00273791+So;
cos(fis)×cos(delta)×cos(tS-ls); if sigma=0 then sigma:=pi/2
else sigma:=arctan(sqrt(1-sigma↑2)/sigma);
if sigma<0 then sigma:=pi+sigma; sigma0:=R/r;
sigma0:=arctan(sqrt(1-sigma0↑2)/sigma0);
sigma0:=sigma0+pi/2; IISun:=sigma<sigma0 end IISun;

G0: writetext(⌘< Ephemerides for the satellite ⌘); writecopy(⌘<[ ]⌘);
pi:=3.14159265; twopi:=6.28318531; rad:=57.2957795;
dl:=0.98564734/rad; R:=6378.245; dS:=236.55536/86400;
input(NrS, year, dy, mn); Eo:=DAY(dy, mn)+inone/24+inone/1440;
incl:=inone/rad; Lo:=inone/rad; dL:=2880×pi/(1440+inone);
omega0:=inone/rad; domega:=inone/rad; Po:=inone/1440
dP:=inone/1440; e:=inone; rp:=inone×1.609344;
So:=inone/rad+Lo-(Eo-entier(Eo))×twopi×1.00273791;
RED(So, twopi); lo:=inone/rad; dvo:=inone×twopi/(1440×Po);
max:=inone/rad; min:=inone/rad; writetext(⌘< for the period
from ⌘); init:=DAY(typein, typein); writetext(⌘< to ⌘);
fin:=DAY(typein, typein); writetext
(⌘< based on the SATOR elements of ⌘);
write(⌘nd⌘, dy, writetext(⌘</⌘), mn); writecr; mn:=typechar;

G1: nob:=inone; outcr; begin array OBS [1:nob, 1:4];
real lop, tn, n, fiop, fim, lam, No1, No2, Np, NN, Tn, P, OMEGA, omega, rL,
M, dt, dv, u1, v1, T1, A1, h1, u2, v2, T2, A2, h2, r1, fis1, ls1, r2, fis1, ls2,
Tbeg, Tend; integer i, j, k, w1, w2; boolean one, B;
lop:=0; fim:=0; for r:=1 step 1 until nob do
begin OBS [i, 1]:=inone; OBS [i, 2]:=IN;
OBS [i, 3]:=IN; OBS [i, 4]:=inone;
lop:=lop+OBS [i, 3]; fim:=fim+OBS [i, 2] end;
lop:=lop/nob; fim:=fim/nob; fiop:=fim;
one:=fiop≥incl ∨ fiop≥pi-incl; if one then
begin if incl<pi/2 then fiop:=incl
else fiop:=pi-incl; lam:=pi/2;
No1:=(lop+lam-Lo)/(dL×Po); No2:=No1 end
else begin lam:=sin(fiop)×cos(incl)/(cos(fiop)×sin(incl));
lam:=arctan(lam/sqrt(1-lam↑2)); No1:=lam;
No1:=(lop+No1-Lo)/(dL×Po); if lam≥0 then
begin No2:=pi-lam; No2:=(lop+No2-Lo)/(dL×Po) end else
begin No2:=-pi-lam; No2:=(lop+No2-Lo)/(dL×Po) end end;
w1:=entier(init-Eo); w2:=entier(fin-Eo);
Np:=((w1+w2)×dP)/(2×Po)+Po; Np:=1/Np; NN:=No1;
L1: for k:=w1 step 1 until w2 do
begin twlgh(k, lo, fiop, lop, So, Tbeg, Tend);
fim:=NN+Np×k; n:=entier(fim+0.5)-2;
for j:=1, 1, 1 do begin n:=n+j;

```

```

TnodeP(Eo, omega0, domega, e, n, Po, dP, P, Tn);
tn := Tn - entier(Tn); if tn + P/4 ≥ Tbeg ∨ tn + P/4 < Tend then
begin for i := 1 step 1 until nobs do begin T1 := Tn;
L2: OMEGA := (T1 - Eo) × dL + Lo; RED(OMEGA, twopi);
rL := OMEGA - OBS [i, 3]; RED(rL, twopi); if cos(rL) = 0
then u2 := pi/2 else u2 := arctan((sin(rL) × cos(incl) +
sin(OBS [i, 2]) × sin(incl) / cos(OBS [i, 2])) / cos(rL));
if rL > pi then rL := twopi - rL else rL := rL;
if rL ≤ pi/2 then u2 := abs(u2) else u2 := pi - abs(u2);
omega := omega0 + domega × (T1 - Eo) / P; RED(omega, twopi);
v2 := u2 - omega; M := VtM(omega, e);
dt := M + VtM(v2, e); dt := dt × P / twopi; T2 := Tn + dt;
if abs(T2 - T1) > 0.00005 then begin T1 := T2; go to L2 end;
Ahtop(v2, rp, e, incl, u2, OMEGA, OBS [i, 2], OBS [i, 3], OBS [i, 4],
r2, fis2, ls2, A2, h2); if h2 ≥ max then begin
if IISun(Eo, T2, lo, dl, So, R, r2, fis2, ls2) then
begin B := false; dv := dvo;
L3: v1 := v2 - dv; tn := dt - (M + VtM(v1, e)) × P / twopi; T1 := T2 - tn;
OMEGA := (T1 - Eo) × dL + Lo; RED(OMEGA, twopi); ul := omega + v1;
Ahtop(v1, rp, e, incl, ul, OMEGA, OBS [i, 2], OBS [i, 3], OBS [i, 4],
r1, fis1, ls1, A1, h1); if h1 ≥ min ∧ (B ∨
IISun(Eo, T1, lo, dl, So, R, r1, fis1, ls1)) then
begin real p; integer m, d, sm, g1, m1, g2, m2;
integer array D [1:12];
integer procedure sumdig(f); value f; integer f;
begin integer kf; kf := f ÷ 10; sumdig := if f < 10 then f
else f - kf × 10 + sumdig(kf) end sumdig;
procedure o2(a); value a; integer a;
if a < 10 then begin outchar(16);
output(⊗ n ⊗, a) end else output(⊗ nd ⊗, a);
procedure o3(a); value a; integer a;
if a < 100 then begin outchar(16);
o2(a) end else output(⊗ ndd ⊗, a);
d := entier(T1); p := (T1 - d) × 24; g1 := entier(p); m1 := (p - g1) × 60;
p := (T2 - entier(T2)) × 24; g2 := entier(p); m2 := (p - g2) × 60;
A1 := A1 × rad; A2 := A2 × rad; h1 := h1 × rad; h2 := h2 × rad;
for m := 1 step 1 until 12 do
D[m] := if m = 4 ∨ m = 6 ∨ m = 9 ∨ m = 11 then 30 else if m = 2 then 28 else 31;
if year ÷ 4 × 4 = year then D[2] := 29; m := 0;
for m := m + 1 while d > D[m] do d := d - D[m]; sm := 0;
for m := d, g1, m1, A1, h1, g2, m2, A2, h2 do
sm := sm + sumdig(m); if sm ≥ 100 then sm := sm - 100;
outtext(⊗ < satat ⊗);
output(⊗ ddddd ⊗, OBS [i, 1], outchar(23), outchar(0), NrS);
outchar(0); o2(sm); outchar(23); o2(d); outchar(0); o2(g1);
o2(m1); outchar(23); outchar(0); o3(A1); o2(h1); outchar(0); o2(g2);
o2(m2); outchar(23); outchar(0); o3(A2); o2(h2); outchar(0);
outtext(⊗ < + ⊗); outcr end
else begin dv := dv/2; if h1 ≥ min then B := true; go to L3
end end end end end end;
if No1 ≠ No2 then begin No1 := NN := No2; go to L1 end
else begin writetext(⊗ < group stations ⊗) end end;
if typechar = 39 then go to G1
else begin writetext(⊗ < END ⊗);
if typechar = 41 then go to G0 end
end programme;

```

### Пример

Modified orbital elements			
Name	[PAGEOS A]	Object	66561
Epoch	1966/27/8,		0h 46.25 m
Inclination	86.93 g	Node	14.92 g
DeltaJ	-4.29 m	Perigee	219.75 g
Change	-0.01335 g	Period	181.367 m
Change	-0.00046 m	Eccentricity	0.05058
Perigee radius	6262 miles	Ascending node	331.24 g
Sun longitude	154.7884 g	Delta Vo	12 m
Hmax	25 g	Hmin	20 g
Init	1/9	Fin	10/9
Tracking stations: 2			
Sofia	1101,	42 g 41 m 2 s	336 g 39 m 10.5 s 6369.137 km
Belogradchik	1103,	43 g 37 m 23 s	337 g 19 m 24.6 s 6368.821 km

### Резултаты:

satat	1101 x	66561	61 x 01	2120 x	16338	2127 x	08671+
satat	1103 x	66561	62 x 01	2121 x	16238	2128 x	08671+
satat	1101 x	66561	74 x 02	2131 x	17139	2137 x	08977+
satat	1103 x	66561	72 x 02	2131 x	17039	2138 x	08876+
satat	1101 x	66561	71 x 03	1845 x	08125	1902 x	05826+
satat	1103 x	66561	71 x 03	1846 x	08026	1903 x	05726+
satat	1101 x	66561	68 x 03	2141 x	17940	2148 x	09183+
satat	1103 x	66561	68 x 03	2142 x	17840	2149 x	09182+
.							
.							
.							

## АЛГОЛ-ПРОГРАМА ЗА ЕФЕМЕРИДИ НА СПЪТНИЦИ

Л.-А. Садовски

(Резюме)

Описва се алгоритъм за изчисляване на ефемериди на ИСЗ въз основа на модифицираните орбитни елементи. Същият може да бъде прилаган към спътници с произволен наклон на орбитите, неголеми эксцентритети и за 30 наблюдателни станции със северна географска ширина в район с диаметър 1500 km.

Дават се работните формули на метода. Прилага се програма за електронна изчислителна машина, написана на универсалния език АЛГОЛ 60, заедно с пример. Същата е използвана за изчисляване на ефемериди на спътниците „Ехо 2“ и „Пагеос 1“ по време на есенния сеанс на синхронните наблюдения на станциите Познан (Полша) и София (България).

## ALGOL-PROGRAMME FOR EPHEMERIDES OF SATELLITES

*L.-A. Sadowsky*

(Summary)

An algorithm for the calculation of ephemerides of artificial satellites of the Earth on the basis of the modified orbital elements is described. It can be applied to satellites with an arbitrary inclination of orbits, moderate eccentricities and for 30 tracking stations with a northern geographical latitude in an area with a diameter of 1,500 km.

The working formulae of the method are given. The programme for an electronic computer, written in the universal language ALGOL 60, with an example as illustration, is supplied. The same was used for the calculation of ephemerides of the satellites "Echo 2" and "Pageos 1" during the autumn period of simultaneous observations of the stations in Poznan (Poland) and in Sofia (Bulgaria).