Решения на задачи за изпит по НМ

Камен Димитров Младенов, FN

DATE

Всички изчисления ще са с точност до 8 знака след десетичния знак.

Първо ще подготвим таблицата на орбиталните елементи. Уебсайтът на NASA [1] предоставя следните константи за началото на епохата (1 Януари 2000, 00:00 часа):

	a au, au/Cy	e rad, rad/Cy	I deg, deg/Cy	L deg, deg/Cy	long.peri. deg, deg/Cy	long.node. deg, deg/Cy
Mercury	0.38709927	0.20563593	7.00497902	252.25032350	77.45779628	48.33076593
	0.00000037	0.00001906	-0.00594749	149472.67411175	0.16047689	-0.12534081
Venus	0.72333566	0.00677672	3.39467605	181.97909950	131.60246718	76.67984255
	0.00000390	-0.00004107	-0.00078890	58517.81538729	0.00268329	-0.27769418
EM Bary	1.00000261	0.01671123	-0.00001531	100.46457166	102.93768193	0.0
	0.00000562	-0.00004392	-0.01294668	35999.37244981	0.32327364	0.0
Mars	1.52371034	0.09339410	1.84969142	-4.55343205	-23.94362959	49.55953891
	0.00001847	0.00007882	-0.00813131	19140.30268499	0.44441088	-0.29257343
Jupiter	5.20288700	0.04838624	1.30439695	34.39644051	14.72847983	100.47390909
	-0.00011607	-0.00013253	-0.00183714	3034.74612775	0.21252668	0.20469106
Saturn	9.53667594	0.05386179	2.48599187	49.95424423	92.59887831	113.66242448
	-0.00125060	-0.00050991	0.00193609	1222.49362201	-0.41897216	-0.28867794
Uranus	19.18916464	0.04725744	0.77263783	313.23810451	170.95427630	74.01692503
	-0.00196176	-0.00004397	-0.00242939	428.48202785	0.40805281	0.04240589
Neptune	30.06992276	0.00859048	1.77004347	-55.12002969	44.96476227	131.78422574
	0.00026291	0.00005105	0.00035372	218.45945325	-0.32241464	-0.00508664

Всички колони със стойности спрямо градуси ще бъдат преобразувани спрямо радиани, използвайки формулата $\text{rad} = \text{deg} \cdot \frac{\pi}{180}$ (с точност до 8 знака след десетичния знак).

Ще сменим наименованията на елементите: a $\to$ a_{2000}, e $\to$ e_{2000}, I $\to$ i_{2000}, L $\to$ λ_{2000}, long.peri. $\to \tilde{\omega}_{2000}$ и long.node. $\to \theta_{2000}$. Ще бележим измененията с Δ. Получаваме:

	$a_{2000}, \Delta a$ au, au/Cy	$e_{2000}, \Delta e$ rad, rad/Cy	$i_{2000}, \Delta i$ rad, rad/Cy	$\lambda_{2000}, \Delta\lambda$ rad, rad/Cy	$\tilde{\omega}_{2000}, \Delta\tilde{\omega}$ rad, rad/Cy	$\theta_{2000}, \Delta\theta$ rad, rad/Cy
Mercury	0.38709927	0.20563593	0.12225995	4.40259868	1.35189358	0.84353100
	0.00000037	0.00001906	-0.00010380	2608.79030501	0.00280085	-0.00218761
Venus	0.72333566	0.00677672	0.05924827	3.17613446	2.29689636	1.33831572
	0.00000390	-0.00004107	-0.00001377	1021.32854958	0.00004683	-0.00484668
EM Bary	1.00000261	0.01671123	-0.00000027	1.75343756	1.79660147	0.0
	0.00000562	-0.00004392	-0.00022596	628.30757790	0.00564219	0.0
Mars	1.52371034	0.09339410	0.03228321	-0.07947238	-0.41789517	0.86497713
	0.00001847	0.00007882	-0.00014192	334.06130168	0.00775643	-0.00510637
Jupiter	5.20288700	0.04838624	0.02276602	0.60033114	0.25706047	1.75360053
	-0.00011607	-0.00013253	-0.00003206	52.96631189	0.00370929	0.00357253
Saturn	9.53667594	0.05386179	0.04338874	0.87186604	1.61615531	1.98378354
	-0.00125060	-0.00050991	0.00003379	21.33653879	-0.00731244	-0.00503838
Uranus	19.18916464	0.04725744	0.01348507	5.46703627	2.98371499	1.29183904
	-0.00196176	-0.00004397	-0.00004240	7.47842217	0.00712187	0.00074012
Neptune	30.06992276	0.00859048	0.03089309	-0.96202600	0.78478315	2.30006864
	0.00026291	0.00005105	0.00000617	3.81283674	-0.00562720	-0.00008878

Без долен индекс и Δ ще обозначаваме величина след време t от началото на епохата. Тоест, обозначаваме $a = a_{2000} + t \cdot \Delta a$, $e = e_{2000} + t \cdot \Delta e$, ...

Финално, ще изчислим рожденната ми дата във векове. Аз съм роден на 27 Април 2002 година. Това е 117-ят ден в годината, което е $\frac{117}{365.25} = 0.32032854$ Юлиански години. Общо излиза 2.32032854 Юлиански години след началото на епохата, което са $\frac{2.32032854}{100} = 0.02320328$ Юлиански века.

1. Решение Ще напишем програма на езика `R`, която намира координатите и скоростите на планетите в деня на моето раждане. Входните данни на програмата ще бъдат орбиталните елементи в началото на епохата и рожденната ми дата (в Юлиански векове от началото на епохата).

Дефинираме функция `atDay`, която пресмята новата стойност на величина след време t:

```
atDay <- function(epochConstant, rate, t) {
  epochConstant + t * rate
}
```

От теорема знаем, че векторът координат r и векторът скорост v се пресмятат чрез:

$$r = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = Q \begin{pmatrix} a(\cos u - e) \\ a\sqrt{1-e^2}\sin u \\ 0 \end{pmatrix} \qquad v = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = Q \begin{pmatrix} -\sin u \\ \sqrt{1-e^2}\cos u \\ 0 \end{pmatrix} \frac{an}{1 - e\cos u}$$

Където Q е матрицата на въртения около нулата:

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{pmatrix} \begin{pmatrix} \cos g & -\sin g & 0 \\ \sin g & \cos g & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Дефинираме помощни функции `m3x3` и `p`. Първата приема редица елементи и ги преобразува в 3×3 матрица, втората изкарва подаденото на екрана, като всички числа са с 8 цифри прецизност:

```
m3x3 <- function(...) {
  matrix(c(...), nrow = 3, ncol = 3, byrow = TRUE)
}
p <- function(x) {
  formatC(x, format="f", digits = 8)
}
```

Дефинираме функция `Q`, която пресмята матрица на въртене по подадени стойности θ, i, g (операторът `%*%` умножава матрици и вектори):

```
Q <- function(theta, i, g) {
  m3x3(
    cos(theta), -sin(theta), 0,
    sin(theta),  cos(theta), 0,
             0,           0, 1
  ) %*% m3x3(
    1,      0,       0,
    0, cos(i), -sin(i),
    0, sin(i),  cos(i)
  ) %*% m3x3(
    cos(g), -sin(g), 0,
    sin(g),  cos(g), 0,
         0,       0, 1
  )
}
```

Дефинираме функция `coordinates`, която изчислява стойността на r вектора по горната формула:

```
coordinates <- function(theta, i, g, a, u, e) {
   Q(theta, i, g) %*% c(
    a * (cos(u) - e),
    a * sqrt(1 - e^2) * sin(u),
    0
  )
}
```

Дефинираме функция `speed`, която изчислява стойността на v вектора по горната формула:

```
speed <- function(theta, i, g, a, u, e, n) {
   Q(theta, i, g) %*% c(
    -sin(u),
    sqrt(1 - e^2) * cos(u),
    0
  ) * a * n / (1 - e * cos(u))
}
```

По дефиниция знаем, че $\tilde{\omega} = g + \theta$ и $\lambda = l + g + \theta = l + \tilde{\omega}$. То, можем да намерим стойностите на g и l чрез формулите:

$$g = \tilde{\omega} - \theta \qquad l = \lambda - \tilde{\omega}$$

Имплементираме ги във функции `g` и `l`:

```
g <- function(omega, theta) { omega - theta }
l <- function(lambda, omega) { lambda - omega }
```

Един от начините да дефинираме средното движение n е чрез средната аномалия l:

$$l = l_0 + n(t - t_0) \qquad n = \frac{l - l_0}{t - t_0}$$

В нашата задача, $l_0 = l_{2000} = \lambda_{2000} - \tilde{\omega}_{2000}$. Нека да зачетем t_0 за началото на епохата и t за време от това начало (във векове), тоест свеждаме $t - t_0$ до t. Тогава намираме, че n е изменението на средната аномалия l:

$$n = \frac{l - l_{2000}}{t} = \frac{(l_{2000} + t \cdot \Delta l) - l_{2000}}{t} = \frac{t \cdot \Delta l}{t} = \Delta l$$

Ще покажем, че изменението на средната аномалия е измението на λ минус изменението на $\tilde{\omega}$:

$$\Delta l = n = \frac{l - l_{2000}}{t} = \frac{(\lambda - \tilde{\omega}) - (\lambda_{2000} - \tilde{\omega}_{2000})}{t} = \frac{(\lambda_{2000} + t \cdot \Delta\lambda) - (\tilde{\omega}_{2000} + t \cdot \Delta\tilde{\omega}) - \lambda_{2000} + \tilde{\omega}_{2000}}{t}$$
$$= \frac{t \cdot \Delta\lambda - t \cdot \Delta\tilde{\omega}}{t} = \Delta\lambda - \Delta\tilde{\omega}$$

Под тази формула, n е в единици радиани за век, обаче на нас ни трябва безмерната ѝ стойност. Следователно, ще разделим n на 100 и 2π. Имплементираме формулата във функция `n`:

```
n <- function(dlambda, domega) {
  (dlambda - domega) / 100 / (2*pi)
}
```

Ще дефинираме стойността u използвайки вторият метод за решаване на уравнението на Кеплер:

$$u = l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(\cdots)))$$

Нуждаем се от 8 знака прецизност, затова ще използваме следното приближение:

$$u = l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(l + e \cdot \sin(l)))))))$$

Дефинираме функция `u`, която пресмята стойността на u чрез рекурсивен алгоритъм:

```
u <- function(l, e, depth=8) {
  if (depth <= 0) { l }
  else            { l + e * sin(u(l, e, depth - 1)) }
}
```

Дължината на векторите r и v са част от решението. Дефинираме функция `length`, която приема 3-мерен вектор и връща евклидовата му дължина:

```
length <- function(vec) { sqrt(vec[1]^2 + vec[2]^2 + vec[3]^2) }
```

Вече сме готови да имплементираме кода, който обединява всичко това. Дефинираме функция `planetData`, която приема датата t във векове от началото на епохата, орбиталните елементи в началото на епохата и тяхното изменение. Функцията пресмята координатите x, y, z; радиуса $|r|$; покомпонентните скорости v_x, v_y, v_z и скоростта на планетата v:

```
planetData <- function(t, a2000, da, e2000, de, i2000, di, lambda2000, dlambda, omega2000,
    domega, theta2000, dtheta)
{
  a <- atDay(a2000, da, t)
  e <- atDay(e2000, de, t)
  i <- atDay(i2000, di, t)
  lambda <- atDay(lambda2000, dlambda, t)
  omega <- atDay(omega2000, domega, t)
  theta <- atDay(theta2000, dtheta, t)

  g <- g(omega, theta)
  l <- l(lambda, omega)
  n <- n(dlambda, domega)
  u <- u(l, e)

  r <- coordinates(theta, i, g, a, u, e)
  v <- speed(theta, i, g, a, u, e, n)

  c(r, length(r), v, length(v))
}
```

Финално дефинираме функцията `solarSystemAtBirth`, която извиква `planetData` за всяка планета с подадени константи:

```
solarSystemAtBirth <- function() {
  birthDate <- 0.02320328
  rbind(
    planetData(birthDate,    # Меркурии
               0.38709927,    0.00000037, 0.20563593, 0.00001906, 0.12225995, -0.00010380,
               4.40259868, 2608.79030501, 1.35189358, 0.00280085, 0.84353100, -0.00218761),
    planetData(birthDate,    # Венера
               0.72333566,    0.00000390, 0.00677672, -0.00004107, 0.05924827, -0.00001377,
               3.17613446, 1021.32854958, 2.29689636,  0.00004683, 1.33831572, -0.00484668),
    planetData(birthDate,    # EM Bary
               1.00000261,   0.00000562, 0.01671123, -0.00004392, -0.00000027, -0.00022596,
               1.75343756, 628.30757790, 1.79660147,  0.00564219,  0.0,         0.0),
    planetData(birthDate,    # Марс
                1.52371034,   0.00001847,  0.09339410, 0.00007882, 0.03228321, -0.00014192,
               -0.07947238, 334.06130168, -0.41789517, 0.00775643, 0.86497713, -0.00510637),
    planetData(birthDate,    # Юпитер
               5.20288700, -0.00011607, 0.04838624, -0.00013253, 0.02276602, -0.00003206,
               0.60033114, 52.96631189, 0.25706047,  0.00370929, 1.75360053,  0.00357253),
    planetData(birthDate,    # Сатурн
               9.53667594, -0.00125060, 0.05386179, -0.00050991, 0.04338874,  0.00003379,
               0.87186604, 21.33653879, 1.61615531, -0.00731244, 1.98378354, -0.00503838),
    planetData(birthDate,    # Уран
               19.18916464, -0.00196176, 0.04725744, -0.00004397, 0.01348507, -0.00004240,
                5.46703627,  7.47842217, 2.98371499,  0.00712187, 1.29183904,  0.00074012),
    planetData(birthDate,    # Нептун
               30.06992276, 0.00026291, 0.00859048,  0.00005105, 0.03089309,  0.00000617,
               -0.96202600, 3.81283674, 0.78478315, -0.00562720, 2.30006864, -0.00008878)
  )
}
```

Извикваме функцията:

```
p(solarSystemAtBirth())
```

И получаваме следния резултат:

	x	y	z	$\|r\|$	v_x	v_y	v_z	$\|v\|$
Mercury	-0.25698453	0.21643004	0.04126659	0.33850545	-1.38648452	-1.18390004	0.03053503	1.82342837
Venus	-0.11664062	0.70946840	0.01643289	0.71918043	-1.16420814	-0.19720358	0.06450187	1.18255247
EM Bary	-0.79848552	-0.61311058	0.00000338	1.00671928	0.59280163	-0.79699190	0.00000439	0.99328237
Mars	0.01018262	1.56865936	0.03261130	1.56903135	-0.78255752	0.07431329	0.02078708	0.78635288
Jupiter	-1.81325044	4.88999123	0.02034819	5.21539121	-0.41697241	-0.13210905	0.00988286	0.43751167
Saturn	2.07612673	8.79599349	-0.23579568	9.04076343	-0.33293384	0.07398502	0.01195145	0.34126466
Uranus	16.49642718	-11.30522661	-0.25582952	20.00014268	0.12742638	0.17777408	-0.00099186	0.21872835
Neptune	18.92712374	-23.39768921	0.04565204	30.09468321	0.14091098	0.11598482	-0.00563534	0.18259282

2. Решение Ще напишем програма на езика `R`, която намира елементите на Делоне и Поанкаре (от първи и втори вид) в деня на моето раждане. За целта се нуждаем и от масите на всички планети.

В допълнението към астрономическия алманах, глава 8, страници 19-20 [2], намираме:

GMs/GM1	6023600.0	$mass^{-1}$ for Mercury
GMs/GM2	408523.71	$mass^{-1}$ for Venus
GMs/GMB	328900.5614	$mass^{-1}$ for EM-Bary
GMs/GM4	3098708.0	$mass^{-1}$ for Mars
GMs/GM5	1047.3486	$mass^{-1}$ for Jupiter
GMs/GM6	3497.898	$mass^{-1}$ for Saturn
GMs/GM7	22902.98	$mass^{-1}$ for Uranus
GMs/GM8	19412.24	$mass^{-1}$ for Neptune

Ще се нуждаем от $\mu = \frac{m_{\text{планета}}}{m_{\text{слънце}}}$. Преобразуваме таблицата по формулата $\mu = \frac{1}{\text{GMs/GMX}}$:

	μ
Mercury	0.00000016
Venus	0.00000244
EM Bary	0.00000304
Mars	0.00000032
Jupiter	0.00095479
Saturn	0.00028588
Uranus	0.00004366
Neptune	0.00005151

Входните данни на програмата ще бъдат орбиталните елементи в началото на епохата, рожденната ми дата (в Юлиански векове от началото на епохата) и съотношенията на масите μ.

Първо ще разгледаме елементите на Делоне: L, G, Θ, l, g, θ.

Искаме да обновим величините от началото на епохата спрямо време t и да изкарваме на екрана стойности с 8 цифри прецизност. Копираме функциите `atDay` и `p` от 1 задача:

```
1 atDay <- function(epochConstant, rate, t) { epochConstant + t * rate }
2 p <- function(x) { formatC(x, format="f", digits = 8) }
```

Величината θ е орбитален елемент. За l и g копираме функциите `l` и `g` от задача 1:

```
1 l <- function(lambda, omega) { lambda - omega }
2 g <- function(omega, theta) { omega - theta }
```

За величините L, G и Θ знаем следните формули по теорема:

$$L = m\sqrt{\gamma}\sqrt{a} \qquad G = L\sqrt{1-e^2} \qquad \Theta = G\cos i$$

Където a, e и i са орбитални елементи. За останалите величини (m и γ) имаме формулите:

$$\gamma = 1 + \mu \qquad m = \frac{\mu}{\gamma} = \frac{\mu}{1+\mu}$$

Имплементираме функции по формулите за всяка велична:

```
1 gamma <- function(mu) { 1 + mu }
2 m <- function(mu) { mu / (1 + mu) }
3 L <- function(m, gamma, a) {
4   m * sqrt(gamma) * sqrt(a)
5 }
6 G <- function(L, e) {
7   L * sqrt(1 - e^2)
8 }
9 Theta <- function(G, i) { G * cos(i) }
```

Обединявайки всичко, получаваме следната функция за намиране на елементите на Делоне по подадени време t, съотношение на маси μ, орбитални елементи в началото на епохата и тяхното изменение за век:

```
delaunay <- function(t, mu, a2000, da, e2000, de, i2000, di, lambda2000, dlambda, omega2000,
    domega, theta2000, dtheta) {
  a <- atDay(a2000, da, t)
  e <- atDay(e2000, de, t)
  i <- atDay(i2000, di, t)
  lambda <- atDay(lambda2000, dlambda, t)
  omega <- atDay(omega2000, domega, t)
  theta <- atDay(theta2000, dtheta, t)

  m <- m(mu)
  gamma <- gamma(mu)

  L <- L(m, gamma, a)
  G <- G(L, e)
  Theta <- Theta(G, i)
  l <- l(lambda, omega)
  g <- g(omega, theta)

  c(L, G, Theta, l, g, theta)
}
```

Дефинираме функцията `delaunayAtBirth`, която връща всички елементи на Делоне за всички планети с подадени константи:

```
delaunayAtBirth <- function() {
  birthDate <- 0.02320328
  rbind(
    delaunay(birthDate, 0.00000016,    # Меркурии
               0.38709927,    0.00000037, 0.20563593, 0.00001906, 0.12225995, -0.00010380,
               4.40259868, 2608.79030501, 1.35189358, 0.00280085, 0.84353100, -0.00218761),
    delaunay(birthDate, 0.00000244,    # Венера
               0.72333566,    0.00000390, 0.00677672, -0.00004107, 0.05924827, -0.00001377,
               3.17613446, 1021.32854958, 2.29689636,  0.00004683, 1.33831572, -0.00484668),
    delaunay(birthDate, 0.00000304,    # EM Bary
               1.00000261,   0.00000562, 0.01671123, -0.00004392, -0.00000027, -0.00022596,
               1.75343756, 628.30757790, 1.79660147,  0.00564219,  0.0,         0.0),
    delaunay(birthDate, 0.00000032,    # Марс
                1.52371034,   0.00001847,  0.09339410, 0.00007882, 0.03228321, -0.00014192,
               -0.07947238, 334.06130168, -0.41789517, 0.00775643, 0.86497713, -0.00510637),
    delaunay(birthDate, 0.00095479,    # Юпитер
               5.20288700, -0.00011607, 0.04838624, -0.00013253, 0.02276602, -0.00003206,
               0.60033114, 52.96631189, 0.25706047,  0.00370929, 1.75360053,  0.00357253),
    delaunay(birthDate, 0.00028588,    # Сатурн
               9.53667594, -0.00125060, 0.05386179, -0.00050991, 0.04338874,  0.00003379,
               0.87186604, 21.33653879, 1.61615531, -0.00731244, 1.98378354, -0.00503838),
    delaunay(birthDate, 0.00004366,    # Уран
               19.18916464, -0.00196176, 0.04725744, -0.00004397, 0.01348507, -0.00004240,
                5.46703627,  7.47842217, 2.98371499,  0.00712187, 1.29183904,  0.00074012),
    delaunay(birthDate, 0.00005151,    # Нептун
               30.06992276, 0.00026291, 0.00859048,  0.00005105, 0.03089309,  0.00000617,
               -0.96202600, 3.81283674, 0.78478315, -0.00562720, 2.30006864, -0.00008878)
  )
}
```

Елементите на Поанкаре от първи вид са следната редица стойности по дефиниция:

$$L, \quad L - G, \quad G - \Theta, \quad l + g + \theta, \quad -g - \theta, \quad -\theta$$

Използвайки вече дефинираната `delaunayAtBirth`, можем итеративно да конвертираме всеки ред с елементи на Делоне към елементите на Поанкаре:

```
poincare1 <- function() {
  data <- delaunayAtBirth()
  result <- c()

  for (planet in 1:8) {
    L <- data[planet,1]; G <- data[planet,2]; Theta <- data[planet,3]
    l <- data[planet,4]; g <- data[planet,5]; theta <- data[planet,6]

    result <- rbind(
      result,
      c(L, L - G, G - Theta, l + g + theta, - g - theta, - theta)
    )
  }

  result
}
```

Елементите на Поанкаре от втори вид са следната редица стойности по дефиниция:

$$\begin{array}{ccccc} L, & & \xi = \sqrt{2(L-G)}\cos(g+\theta), & & p = \sqrt{2(G-\Theta)}\cos(\theta), \\ l+g+\theta, & & \eta = -\sqrt{2(L-G)}\sin(g+\theta), & & q = \sqrt{2(G-\Theta)}\sin(\theta) \end{array}$$

Напълно аналогично на `poincare1`, дефинираме `poincare2`:

```
poincare2 <- function() {
  data <- delaunayAtBirth()
  result <- c()

  for (planet in 1:8) {
    L <- data[planet,1]; G <- data[planet,2]; Theta <- data[planet,3]
    l <- data[planet,4]; g <- data[planet,5]; theta <- data[planet,6]

    result <- rbind(
      result,
      c(L,              sqrt(2*(L-G)) * cos(g + theta), sqrt(2*(G-Theta)) * cos(theta),
        l + g + theta, -sqrt(2*(L-G)) * sin(g + theta), sqrt(2*(G-Theta)) * sin(theta))
    )
  }

  result
}
```

Извикваме всяка от тези функции:

```
p(delaunayAtBirth())
p(poincare1())
p(poincare2())
```

И получаваме следните резултати:

Делоне	L	G	Θ	l	g	θ
Mercury	0.00000010	0.00000010	0.00000010	63.58313202	0.50847833	0.84348024
Venus	0.00000208	0.00000208	0.00000207	24.57740932	0.95869419	1.33820326
EM Bary	0.00000304	0.00000304	0.00000304	14.53550183	1.79673239	0.00000000
Mars	0.00000040	0.00000039	0.00000039	8.08956074	-1.28257384	0.86485865
Jupiter	0.00217682	0.00217427	0.00217371	1.57217677	-1.49653689	1.75368342
Saturn	0.00088271	0.00088143	0.00088060	-0.24904191	-0.36768100	1.98366663
Uranus	0.00019125	0.00019104	0.00019102	2.65667995	1.69202403	1.29185621
Neptune	0.00028245	0.00028244	0.00028231	-1.65820826	-1.51541400	2.30006658

Поанкаре 1 вид	L	$L-G$	$G-\Theta$	$l+g+\theta$	$-g-\theta$	$-\theta$
Mercury	0.00000010	0.00000000	0.00000000	64.93509059	-1.35195857	-0.84348024
Venus	0.00000208	0.00000000	0.00000000	26.87430677	-2.29689745	-1.33820326
EM Bary	0.00000304	0.00000000	0.00000000	16.33223422	-1.79673239	-0.00000000
Mars	0.00000040	0.00000000	0.00000000	7.67184554	0.41771520	-0.86485865
Jupiter	0.00217682	0.00000255	0.00000056	1.82932331	-0.25714654	-1.75368342
Saturn	0.00088271	0.00000128	0.00000083	1.36694372	-1.61598564	-1.98366663
Uranus	0.00019125	0.00000021	0.00000002	5.64056019	-2.98388024	-1.29185621
Neptune	0.00028245	0.00000001	0.00000013	-0.87355568	-0.78465258	-2.30006658

Поанкаре 2 вид	L	ξ	p	$l+g+\theta$	η	q
Mercury	0.00000010	0.00001416	0.00002536	64.93509059	-0.00006367	0.00002849
Venus	0.00000208	-0.00000648	0.00001967	26.87430677	-0.00000730	0.00008304
EM Bary	0.00000304	-0.00000653	0.00000001	16.33223422	-0.00002840	0.00000000
Mars	0.00000040	0.00005371	0.00001313	7.67184554	0.00002384	0.00001540
Jupiter	0.00217682	0.00218380	-0.00019305	1.82932331	-0.00057427	0.00104380
Saturn	0.00088271	-0.00007230	-0.00051683	1.36694372	-0.00159886	0.00117985
Uranus	0.00019125	-0.00064559	0.00005131	5.64056019	-0.00010267	0.00017917
Neptune	0.00028245	0.00010218	-0.00034594	-0.87355568	-0.00010203	0.00038713

References

[1] Solar System Dynamics. (2025, Май 1). *Approximate Positions of the Planets.* Jet Propultion Laboratory, National Aeronautics and Space Administration. https://ssd.jpl.nasa.gov/planets/approx_pos.html

[2] Williams, James G., & Standish, Myles E. (2006). Orbital Ephemerides of the Sun, Moon, and Planets. In Seidelmann, P. K. (Eds.), *Explanatory Supplement to the Astronomical Almanac.* https://www.researchgate.net/publication/232203657_Orbital_Ephemerides_of_the_Sun_Moon_and_Planets