

Лабораторная работа 2 ИТ в физике

Круглов Иван, ИВТ 1/2

1. Математическая модель

Введем две системы координат:

1. Гелиоцентрическую ХСУ (с центром в точке С).

2. Геоцентрическую Х'ЗУ' (с центром в точке З).

Система ХСУ неподвижная, а система Х'ЗУ' – подвижная.

$$\vec{ЗМ} = \vec{СМ} - \vec{СЗ},$$

$\vec{СЗ}$ – расстояние от Солнца до Земли,

$\vec{СМ}$ – расстояние от Солнца до Марса.

Для построения траектории движения Марса относительно Земли необходимо найти координаты вектора $\vec{ЗМ}$.

$$x = x_M - x_3$$

$$y = y_M - y_3$$

Солнце, Марс и Землю рассматриваем как материальные точки. Движение материальной точки по окружности описывается следующим уравнением:

$$x = r * \cos(w*t + \varphi).$$

Для нашей задачи уравнения движения Марса в системе координат ХСУ имеют вид:

$$x_M = r_1 * \cos(w_1 * t + \varphi)$$

$$y_M = r_1 * \sin(w_1 * t + \varphi)$$

уравнения движения Земли в системе координат Х'ЗУ' имеют вид: $x_3 = r_2 \cos(w_2 t + \varphi)$

$$y_3 = r_2 * \sin(w_2 * t + \varphi)$$

Окончательно уравнения движения Марса относительно Земли имеют вид:

$$x = r_1 * \cos(w_1 * t + \varphi) - r_2 * \cos(w_2 * t + \varphi)$$

$$y = r_1 * \sin(w_1 * t + \varphi) - r_2 * \sin(w_2 * t + \varphi)$$

Где $w = 2\pi/T$ (Т – период обращения планеты вокруг Солнца).

2. Код программы для визуализации данных на языке python

```

1. from numpy import*
2. from matplotlib.pyplot import*
3.
4. # EARTH
5. rEarth = 1.496 * 10**8
6. tEarth = 365.2
7. wEarth = 2 * pi / tEarth
8. #MARS
9. rMars = 2.279 * 10**8
10. tMars = 687.0
11. wMars = 2 * pi / tMars
12. # JUPITER
13. rJupiter = 7.786 * 10**8
14. tJupiter = 4331
15. wJupiter = 2 * pi / tJupiter
16. # SATURN
17. rSaturn = 14.335 * 10**8
18. tSaturn = 10747
19. wSaturn = 2 * pi / tSaturn
20.
21. N = 10**4
22.
23. def getX(r, w, i):
24.     return r * cos(w * i)
25. def getY(r, w, i):
26.     return r * sin(w * i)
27.
28. xMars = array([getX(rMars, wMars, i) for i in arange(0, N, 1)])
29. yMars = array([getY(rMars, wMars, i) for i in arange(0, N, 1)])
30. xEarth = array([getX(rEarth, wEarth, i) for i in arange(0, N, 1)])
31. yEarth = array([getY(rEarth, wEarth, i) for i in arange(0, N, 1)])
32. xJupiter = array([getX(rJupiter, wJupiter, i) for i in arange(0, N, 1)])
33. yJupiter = array([getY(rJupiter, wJupiter, i) for i in arange(0, N, 1)])
34.
35. # Earth -> Mars
36. figure()
37. title("Положение Марса относительно Земли")
38. xlabel('x')
39. ylabel('y')
40. x = (xMars - xEarth)
41. y = (yMars - yEarth)
42. plot(x, y)
43. savefig('earth mars.jpg')
44.
45. # Earth -> Jupiter
46. figure()
47. title("Положение Юпитера относительно Земли")
48. xlabel('x')

```

```

49.ylabel('y')
50.x = (xJupiter - xEarth)
51.y = (yJupiter - yEarth)
52.plot(x, y)
53.savefig('earth jupiter.jpg')
54.
55.N = 2 * 10**5
56.xJupiter = array([getX(rJupiter, wJupiter, i) for i in arange(0, N, 1)])
57.yJupiter = array([getY(rJupiter, wJupiter, i) for i in arange(0, N, 1)])
58.xSaturn = array([getX(rSaturn, wSaturn, i) for i in arange(0, N, 1)])
59.ySaturn = array([getY(rSaturn, wSaturn, i) for i in arange(0, N, 1)])
60.
61.# Saturn -> Jupiter
62.figure()
63.title("Положение Юпитера относительно Сатурна")
64.xlabel('x')
65.ylabel('y')
66.x = (xJupiter - xSaturn)
67.y = (yJupiter - ySaturn)
68.plot(x, y)
69.savefig('saturn jupiter.jpg')
70.
71.show()

```

3. Результат работы программы (графики)



