

Московский авиационный институт
(национальный исследовательский университет)

Институт №7 «Робототехнические
и интеллектуальные системы»

Лабораторная работа №2
по курсу теоретической механики
Статика

Выполнил студент группы М70-106С-22
Мастерских Егор Александрович
Преподаватель: Шамин Александр Юрьевич

Оценка:

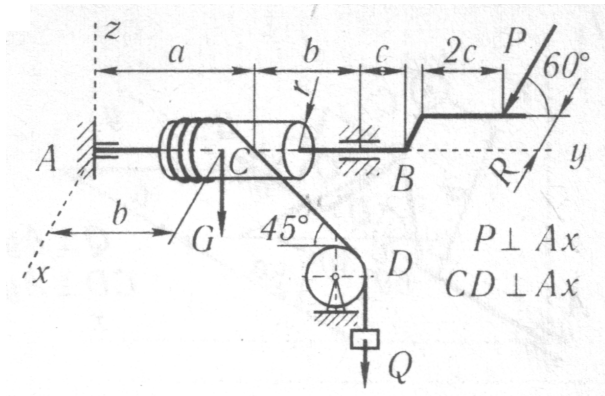
Дата: 9 мая 2023

Москва 2023

Вариант №12

Задание С.7. Определение реакций опор твёрдого тела

Найти реакции опор конструкции (R_A , R_B) и силу P .



Силы, кН		Размеры, см				
Q	G	a	b	c	R	r
4	1	25	20	8	15	10

Текст программы

```

1 from sympy import symbols, Matrix, solve
2 from math import sin, cos, pi
3 import numpy as np
4 from numpy.linalg import norm
5 from tabulate import tabulate
6
7 # модули сил
8 Q = 4
9 G = 1
10
11 # размеры
12 a = 25
13 b = 20
14 c = 8
15 R = 15
16 r = 10
17
18 P, R_Ax, R_Ay, R_Az, R_Bx, R_Bz = symbols('P,R_Ax,R_Ay,R_Az,R_Bx,R_Bz')
19
20 # радиус-векторы, проведённые из точки A
21 # в точку приложения соответствующей силы
22 r_Q = Matrix([r, a, 0])
23 r_G = Matrix([0, b, 0])
24 r_P = Matrix([-R, a + b + 3 * c, 0])
25 r_R_B = Matrix([0, a + b, 0])
26
27 # силы
28 Q_vec = Matrix([0, Q * cos(pi / 4), -Q * sin(pi / 4)])

```

```

29 G_vec = Matrix([0, 0, -G])
30 P_vec = Matrix([0, -P * cos(pi / 3), -P * sin(pi / 3)])
31 R_A_vec = Matrix([R_Ax, R_Ay, R_Az])
32 R_B_vec = Matrix([R_Bx, 0, R_Bz])
33
34 # равнодействующая сила
35 F_vec = Q_vec + G_vec + P_vec + R_A_vec + R_B_vec
36
37 # суммарный момент
38 M_vec = (r_Q.cross(Q_vec) + r_G.cross(G_vec)
39          + r_P.cross(P_vec) + r_R_B.cross(R_B_vec))
40
41 # словарь, где каждой искомой величине соответствует вычисленное значение
42 res = solve([F_vec, M_vec], [P, R_Ax, R_Ay, R_Az, R_Bx, R_Bz])
43
44 P, R_Ax, R_Ay, R_Az, R_Bx, R_Bz = (
45     res[P], res[R_Ax], res[R_Ay], res[R_Az], res[R_Bx], res[R_Bz]
46 )
47 R_A_vec = np.array([R_Ax, R_Ay, R_Az], dtype='float')
48 R_B_vec = np.array([R_Bx, 0, R_Bz], dtype='float')
49 R_A, R_B = norm(R_A_vec), norm(R_B_vec)
50
51 with open('C.7_res.txt', 'w') as out_file:
52     R_A_vec = tuple(R_A_vec.round(1))
53     R_B_vec = tuple(R_B_vec.round(1))
54     table = [['', 'Вектор, кН', 'Норма, кН'],
55              ['P', '', P],
56              ['R_A', R_A_vec, R_A],
57              ['R_B', R_B_vec, R_B]]
58     print(tabulate(
59         table, headers='firstrow', floatfmt='.2f',
60         colalign=['center', 'right', 'right']
61     ), file=out_file)

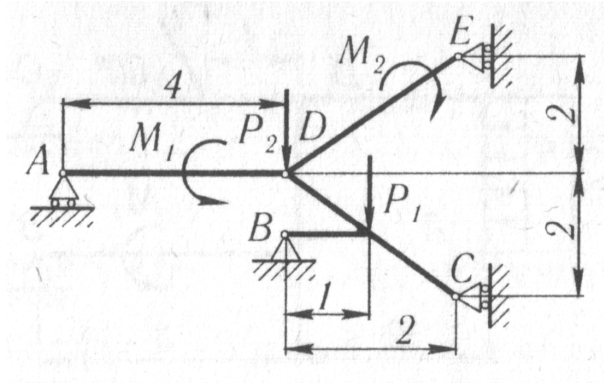
```

Результат работы программы

	Вектор, кН	Норма, кН
P		2.18
R_A	(-1.0, -1.7, 0.8)	2.16
R_B	(1.0, 0.0, 4.9)	5.01

Задание С.4. Определение реакций опор составной конструкции (система трёх тел)

Найти реакции R_A, R_B, R_C, R_D, R_E для конструкции, состоящей из трёх тел, соединённых в одной точке шарниром.



P_1	P_2	M_1	M_2
кН		кН · м	
12	14	36	28

Текст программы

```

1 from sympy import symbols, Matrix, solve
2 import numpy as np
3 from numpy.linalg import norm
4 from tabulate import tabulate
5
6 P1 = 12
7 P2 = 14
8
9 M1 = 36
10 M2 = 28
11
12 R_Ay, R_Bx, R_By, R_Cx, R_Dx, R_Dy, R_Ex = symbols(
13     'R_Ay,R_Bx,R_By,R_Cx,R_Dx,R_Dy,R_Ex'
14 )
15
16 P1_vec = Matrix([0, -P1, 0])
17 P2_vec = Matrix([0, -P2, 0])
18
19 R_A_vec = Matrix([0, R_Ay, 0])
20 R_B_vec = Matrix([R_Bx, R_By, 0])
21 R_C_vec = Matrix([R_Cx, 0, 0])
22 R_D_vec = Matrix([R_Dx, R_Dy, 0])
23 R_E_vec = Matrix([R_Ex, 0, 0])
24
25 M1_vec = Matrix([0, 0, M1])
26 M2_vec = Matrix([0, 0, -M2])
27
28 # радиус-векторы отсчитываются от точки D

```

```

29
30 r_P1 = Matrix([1, -1, 0])
31
32 r_R_A = Matrix([-4, 0, 0])
33 r_R_B = Matrix([0, -1, 0])
34 r_R_C = Matrix([2, -2, 0])
35 r_R_E = Matrix([2, 2, 0])
36
37 M_P1_vec = r_P1.cross(P1_vec)
38
39 M_A_vec = r_R_A.cross(R_A_vec)
40 M_B_vec = r_R_B.cross(R_B_vec)
41 M_C_vec = r_R_C.cross(R_C_vec)
42 M_E_vec = r_R_E.cross(R_E_vec)
43
44 # для балки AD
45 M_res_AD = M_A_vec + M1_vec
46
47 # для балки DE
48 M_res_DE = M_E_vec + M2_vec
49
50 # для конструкции ADE
51 F_res_ADE = R_A_vec + R_D_vec + R_E_vec + P2_vec
52
53 # для конструкции DBC
54 M_res_DBC = M_B_vec + M_C_vec + M_P1_vec
55 F_res_DBC = R_B_vec + R_C_vec - R_D_vec + P1_vec + P2_vec
56
57 res = solve(
58     [M_res_AD, M_res_DE, F_res_ADE, M_res_DBC, F_res_DBC],
59     [R_Ay, R_Bx, R_By, R_Cx, R_Dx, R_Dy, R_Ex]
60 )
61
62 R_Ay, R_Bx, R_By, R_Cx, R_Dx, R_Dy, R_Ex = (
63     res[R_Ay], res[R_Bx], res[R_By], res[R_Cx],
64     res[R_Dx], res[R_Dy], res[R_Ex]
65 )
66
67 # нормы реакций
68 R_A = abs(R_Ay)
69 R_B = norm(np.array([R_Bx, R_By], dtype='float'))
70 R_C = abs(R_Cx)
71 R_D = norm(np.array([R_Dx, R_Dy], dtype='float'))
72 R_E = abs(R_Ex)
73

```

```

74 with open('C.4_res.txt', 'w') as out_file:
75     table = [['', 'Вектор, кН', 'Норма, кН'],
76               ['R_A', (0, R_Ay), R_A],
77               ['R_B', (R_Bx, R_By), R_B],
78               ['R_C', (R_Cx, 0), R_C],
79               ['R_D', (R_Dx, R_Dy), R_D],
80               ['R_E', (R_Ex, 0), R_E]]
81     print(tabulate(
82         table, headers='firstrow', floatfmt='.2f', colalign=['right'] * 3
83     ), file=out_file)

```

Результат работы программы

1	Вектор, кН	Норма, кН
2	---	-----
3	R_A (0, 9)	9.00
4	R_B (16, 31)	34.89
5	R_C (-2, 0)	2.00
6	R_D (14, 5)	14.87
7	R_E (-14, 0)	14.00