

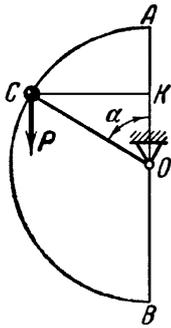
Задача 10

(Задачи 911, 912. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

Тяжелый шарик весом $P=mg$ находится на конце тонкого прута длиной l , могущего вращаться около другого своего конца. Пренебрегая массой прута, определить скорость v , какую приобретет шарик в самой низкой точке, падая с ничтожно малой начальной скоростью из положения неустойчивого равновесия, а также сжимающее и растягивающее усилия в пруте для обоих положений. Кроме того, определить угол поворота α , когда прут не будет испытывать ни растягивающего, ни сжимающего усилия.

Для решения задачи использовать следующие значения параметров:

$$m = 2 \text{ кг}, l = 0.5 \text{ м.}$$



Точный теоретический ответ

Скорость шарика в самой низкой точке определяется по формуле

$$v = 2 \cdot \sqrt{l \cdot g}.$$

Сжимающее усилие в пруте в положении неустойчивого равновесия определяется по формуле

$$F1 = P.$$

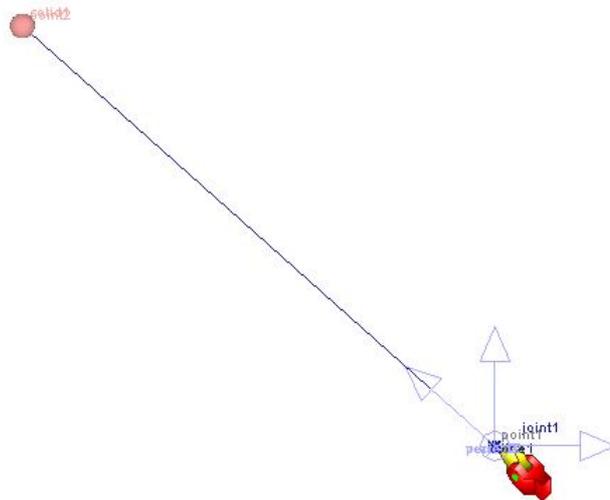
Растягивающее усилие в пруте в самой низкой точке определяется по формуле

$$F2 = 5P.$$

Угол поворота, когда прут не будет испытывать ни растягивающего, ни сжимающего усилия определяется по формуле

$$\alpha = \arccos(2/3).$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте звено не отображается.
- Шарик с прутом (body2). Шарик отображается точкой point2 и телом solid1 с массой m . Прут изображается линией line1 длиной l .

Прут соединен с инерциальным звеном шарниром joint1 типа «пара вращения». Для вывода из положения неустойчивого равновесия задано начальное условие condition1 с малой скоростью v_0 . В проекте задана гравитация с ускорением свободного падения $9.81 \text{ [m/s}^2\text{]}$.

Для определения скорости используется датчик скорости v. Для определения сжимающего и растягивающего усилий используется датчик сил F. Для определения угла поворота шарика с прутом используется внутренний датчик угла шарнира joint1.gamma. Сжимающее усилие в положении неустойчивого равновесия фиксируется в начальный момент времени с помощью события event1. Для определения угла поворота в момент нулевой продольной силы, действующей на прут используется событие event2.

Остановка расчета осуществляется с помощью события event3, когда прут с шариком повернется на угол 180° из положения неустойчивого равновесия в положение устойчивого равновесия. В этот же момент фиксируются значения растягивающего усилия в пруте и скорости шарика.

Для решения задачи используется команда command1 «расчет динамики движения» с временем интегрирования 5 [s] и заданным постоянным шагом.

Результаты моделирования

При получении точного теоретического решения предполагалось, что шарик начинает движение из положения неустойчивого равновесия с ничтожно малой скоростью и, что его радиус равен 0. При моделировании в EULER шарик должен иметь ненулевую начальную скорость и ненулевой радиус. Поэтому решение в EULER будет зависеть от заданной в проекте начальной скорости и радиуса шарика. Для постоянного шага интегрирования 0.001 [s] и для радиуса шарика $r=0.0001 \text{ [m]}$ зависимость от начальной скорости приведена в следующей таблице.

Начальная скорость шарика [m/s]	Относительное отличие от теоретического решения v_delta_rel [-]	Относительное отличие от теоретического решения $F1_delta_rel$ [-]	Относительное отличие от теоретического решения $F2_delta_rel$ [-]	Относительное отличие от теоретического решения a_delta_rel [-]
0.1	0.000 25	0.002 0	0.000 41	0.001 1
0.01	0.000 002 5	0.000 020	0.000 004 1	0.000 007 9
0.001	0.000 000 018	0.000 000 20	0.000 000 028	0.000 001 8

Для постоянного шага интегрирования 0.001 [s] и для начальной скорости шарика $V_0=0.001 \text{ [m/s]}$ зависимость от радиуса шарика приведена в следующей таблице.

Радиус шарика [m]	Относительное отличие от теоретического решения v_delta_rel [-]	Относительное отличие от теоретического решения $F1_delta_rel$ [-]	Относительное отличие от теоретического решения $F2_delta_rel$ [-]	Относительное отличие от теоретического решения a_delta_rel [-]
0.1	0.007 9	0.000 000 2	0.013	0.005 6
0.01	0.000 08	0.000 000 2	0.000 13	0.000 057
0.001	0.000 000 77	0.000 000 2	0.000 001 2	0.000 004 6

Для постоянного радиуса шарика $r=0.001$ [m] и для начальной скорости шарика $V_0=0.001$ [m/s] зависимость от шага интегрирования приведена в следующей таблице.

Шаг интегрирования [s]	Относительное отличие от теоретического решения v_delta_rel [-]	Относительное отличие от теоретического решения $F1_delta_rel$ [-]	Относительное отличие от теоретического решения $F2_delta_rel$ [-]	Относительное отличие от теоретического решения a_delta_rel [-]
0.01	0.000 000 3	0.000 000 2	0.000 000 48	0.000 001 2
0.001	0.000 000 77	0.000 000 2	0.000 001 2	0.000 004 6
0.000 1	0.000 000 77	0.000 000 2	0.000 001 2	0.000 002 1

Текст проекта в EULER

```

scalar m=2 [ kg ];
scalar l=0.5 [ m ];
scalar v0=0.001 [ m/ s ];
scalar g=9.81 [ m/ s2 ];
scalar P=m*g;
scalar r=0.001[m];
point point1=point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2=point( 0 [ m ], l, 0 [ m ] );
line line1=polyLine( list( point1, point2 ) );
solid solid1=sphere( point2, r, mass = m );
body body1=body( color = RGB( 229, 229, 229 ) );
set ground = body1;
body body2=body( color = RGB( 255, 153, 153 ) );
body body2 < ( point2, solid1 );
joint joint1=rotational( body1, body2, point1, projectZ );
gravity gravity1=parallel( reverse( projectY ), g = g );
condition condition1=transVelocity( body1, reverse( projectX ), body2, point2, v0 );
sensor v=velocity( point2 );
vector vector1=vectorPP( point1, point2 );
sensor F=jointForce( force:, body2, point2, vector1, joint1, fixing = unlock: );
event event1=reformsBySensor( list( ), time, 0 [ s ], gauge = list( F ) );
event event2=reformsBySensor( list( ), F, 0 [ N ], gauge = list( joint1.gamma ) );
event event3=reformsBySensor( list( stop( ) ), joint1.gamma, 180 [ deg ], gauge = list( F ) );
sensor v_theoretical=2*sqrt( l*g );
sensor v_delta_rel=abs( ( v - v_theoretical ) / v_theoretical );
sensor F1_theoretical=P;
sensor F1_delta_rel=abs( ( abs( event1.F ) - F1_theoretical ) / F1_theoretical );
sensor F2_theoretical=5*P;
sensor F2_delta_rel=abs( ( abs( event3.F ) - F2_theoretical ) / F2_theoretical );
sensor a_theoretical=arccos( 2/3 ) * 1 [ rad ];
sensor a_delta_rel=abs( ( event2.joint1_gamma - a_theoretical ) / a_theoretical );
command command1=constRK4( 5.00000e+000 [ s ], 1.00000e-004[ s ] );

```

Λ//

Λ Список главных команд;
set dynamics = command1;

Λ//

Λ Единицы измерения;
set units = SI;