

Задача 8

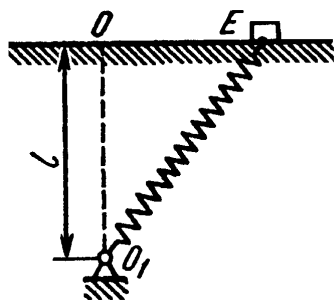
(Задача 30.2. Задачи по теоретической механике. И.В. Мещерский. СПб.: Издательство «Лань», 2001. – 448 с.)

Тело E , масса которого равна m , находится на шероховатой горизонтальной плоскости. Коэффициент трения скольжения равен f . К телу прикреплена пружина жесткости c , второй конец которой прикреплен к шарниру O_1 . Длина недеформированной пружины равна l_0 ; $OO_1 = l$. В начальный момент тело E отклонено от положения равновесия O на конечную величину $OE = a$ и отпущено без начальной скорости.

Определить скорость тела E в момент прохождения положения равновесия O .

Для решения задачи использовать следующие значения параметров:

$$m = 20 \text{ кг}, l_0 = 0.3 \text{ м}, l = 0.5 \text{ м}, a = 0.4 \text{ м}, f = 0.15.$$

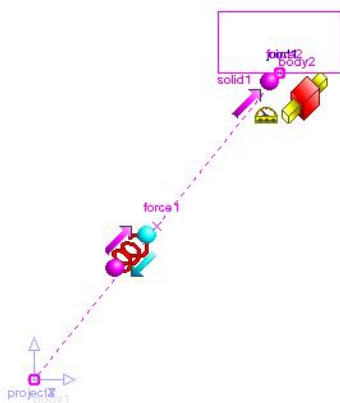


Точный теоретический ответ

Скорость тела определяется по формуле

$$v = \sqrt{\frac{2}{m} \cdot \left\{ c \cdot \left[\frac{a^2}{2} + l_0 \cdot (l - \sqrt{l^2 + a^2}) \right] - f \cdot \left[a \cdot (m \cdot g + c \cdot l) + c \cdot l_0 \cdot l \cdot \ln \frac{l}{a + \sqrt{l^2 + a^2}} \right] \right\}}$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте звено не отображается.
- Тело (body2). Звено отображается телом выдавливания solid1 и имеет массу m .

Пружина моделируется силовым элементом force1 типа «поступательная пружина с коэффициентом жесткости». В проекте задана гравитация с ускорением свободного падения $9.81 \text{ [m/s}^2]$. Перемещение тела по плоскости моделируется шарниром joint1 типа «поступательная пара».

Для определения силы реакции, действующей со стороны плоскости на тело, используется датчик силы N. Для моделирования силы трения используется силовой элемент force2 типа «сила по датчикам на одну точку» с функцией силы function1. Аргументами функции силы трения function1 заданы сила реакции и скорость. В качестве силы реакции используется датчик N1 типа «значение датчика на предыдущем шаге».

интегрирования». Для определения скорости используется датчик v. Для определения расстояния используется датчик s. Для остановки расчета используется событие event1.

Для решения задачи используется команда command1 «расчет динамики движения» с временем интегрирования 1 [s] и заданным постоянным шагом.

Результаты моделирования

Точность решения задачи в EULER в зависимости от шага численного интегрирования (использовался постоянный шаг интегрирования) представлено в следующей таблице.

Шаг интегрирования [s]	Относительное отличие от теоретического решения V_delta_rel [-]
0.001	0.000 28
0.000 1	0.000 029
0.000 01	0.000 004 2
0.000 001	0.000 002 9

Текст проекта в EULER

```

scalar m=20 [ kg ];
scalar c=1000 [ N/ m ];
scalar a=0.4 [ m ];
scalar l0=0.3 [ m ];
scalar l=0.5 [ m ];
scalar f=0.15;
scalar g=9.81 [ m/ s2 ];
point point1=point( 0 [ m ], 0 [ m ], 0 [ m ] );
body body1=body( color = RGB( 229, 229, 229 ) );
set ground = body1;
point point2=point( a-0.1 [ m ], l, 0 [ m ] );
point point3=point( a-0.1 [ m ], l+0.1 [ m ], 0 [ m ] );
point point4=point( a+0.1 [ m ], l+0.1 [ m ], 0 [ m ] );
point point5=point( a+0.1 [ m ], l, 0 [ m ] );
point point6=point( a, l, 0.05 [ m ] );
point point7=point( a, l, -0.05 [ m ] );
point point8=point( a, l, 0 [ m ] );
line line2=polyLine( list( point2, point3, point4, point5, point2 ) );
solid solid1=extrude( line2, point6, point7, mass = m );
body body2=body( color = RGB( 229, 0, 229 ) );
body body2 < ( solid1 );
gravity gravity1=parallel( reverse( projectY ), g = g );
joint joint1=translational( body1, body2, point8, projectX );
force force1=spring( body1, point1, body2, point8, c, s0 = l0 );
sensor v=transVelocity( body1, reverse( projectX ), body2, point8 );
sensor s=bodyDisplacement( body1, point1, projectX, body2, point8 );
sensor N=jointForce( force:, body2, point8, projectY, joint1 );
sensor N1=prevValue( N );
function function1(N [ N ], v [ m/ s ])=f*#N*sign(#v);
force force2=force( body2, point8, projectX, function1, list( N1, v ) );
event event1=reformsBySensor( list( stop( ) ), s, 0 [ m ] );
sensor v_theoretical=sqrt( 2/m*( c*( 0.5*a*a+l0*( l-sqrt( l*l+a*a ) ) )-f*( a*( m*g+c*l )+c*l0*ln( l/( a+sqrt( l*l+a*a ) ) ) ) ) );
sensor v_delta_rel=abs( ( v - v_theoretical ) / v_theoretical );
command command1=constRK4( 1.00000e+001 [ s ], 1.00000e-004 [ s ] );

```

Λ////////////////////////////////////////////////////////////////

Λ Список главных команд;
set dynamics = command1;

Λ////////////////////////////////////////////////////////////////

Λ Единицы измерения;
set units = SI;