

## Задача 11

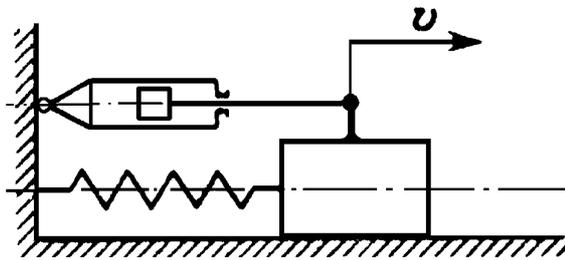
( Задача 8.7. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с. )

Тело массы  $m$ , прикрепленное к концу недеформированной пружины, приводится в прямолинейное поступательное движение по гладкой горизонтальной плоскости с начальной скоростью  $v_0$ , направленной по оси пружины. Величина силы упругости пропорциональна деформации пружины  $\lambda$ , т. е.  $F = c \cdot \lambda$ , где  $c = const > 0$ . Кроме пружины, к телу прикреплен шток с поршнем, помещенным в цилиндр, заполненный жидкостью. При движении поршня возникает сила сопротивления, величина которой  $R = \mu \cdot v^2$ , где  $v$  – скорость поршня,  $\mu = const > 0$ .

Найти значение начальной скорости тела, при котором оно остановится, пройдя путь, равный  $l$ .

Для решения задачи использовать следующие значения параметров:

$m = 4 \text{ кг}$ ,  $c = 10 \text{ Н/м}$ ,  $\mu = 2 \text{ кг/м}$ ,  $l = 0.1 \text{ м}$ .

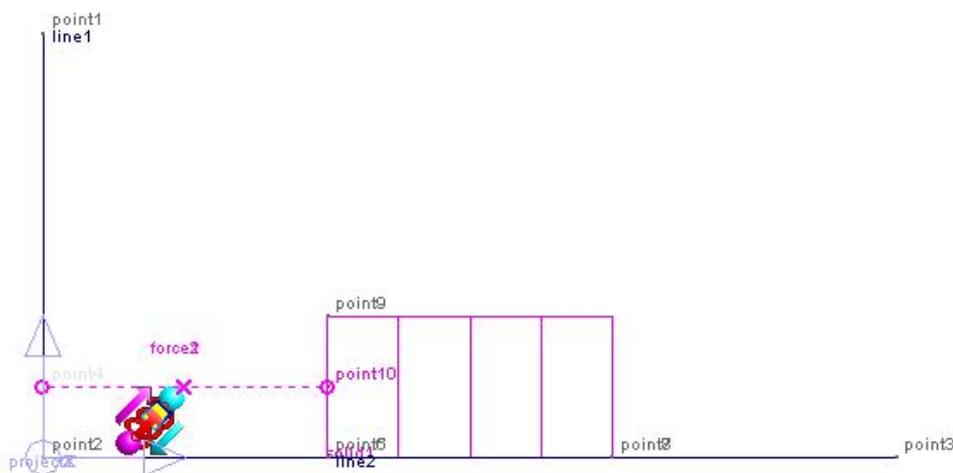


### Точный теоретический ответ

Начальная скорость определяется по формуле

$$v_0 = \sqrt{\frac{c \cdot m}{2 \cdot \mu^2} \cdot \left[ 1 + \left( \frac{2 \cdot \mu \cdot l}{m} - 1 \right) \cdot e^{\frac{2 \cdot \mu \cdot l}{m}} \right]}$$

### Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте инерциальное звено отображается линией line1.
- Тело (body2). Тело отображается параллелепипедом solid1 и имеет массу  $m$ .

Пружина моделируется силовым элементом force1 типа «поступательная пружина с коэффициентом жесткости». Демпфер моделируется силовым элементом force2 типа «поступательный демпфер между двумя точками» с функцией силы function1. Начальная скорость телу сообщается с помощью объекта «условие состояния механизма» condition1.

Для определения перемещения и скорости тела используются соответственно датчики s и v. События event1 останавливает расчет, когда скорость тела станет равна нулю.

Для решения задачи создана команда «Краевая задача» command1. Эта команда позволяет найти начальную скорость тела, при которой оно остановится, пройдя заданный путь. Интегрирование движения системы при решении краевой задачи в command1 выполняется стандартной командой comDynamics.

### **Результаты моделирования**

Решение краевой задачи зависит от заданных абсолютной и относительной погрешностей расчета. Зависимость результатов моделирования в EULER от заданных погрешностей решения краевой задачи приведена в следующей таблице.

Задаваемые значения абс. и отн. погрешностей решения краевой задачи [-]	Относительное отличие от теоретического решения v0_delta_rel [-]
0.001	0.000 2
0.000 1	0.000 22
0.000 01	0.000 000 18

### **Текст проекта в EULER**

```

scalar m=4 [ kg ];
scalar v0=0.15 [ m/ s ];
scalar c=10 [ N/ m ];
scalar mu=2 [ N s2/ m2 ];
scalar l=0.1 [ m ];
point point1=point( 0 [ m ], 0.3 [ m ], 0 [ m ] );
point point2=point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point3=point( 0.6 [ m ], 0 [ m ], 0 [ m ] );
point point4=point( 0 [ m ], 0.05 [ m ], 0 [ m ] );
line line1=polyLine( list( point1, point2, point3 ) );
body body1=body( color = RGB( 229, 229, 229 ) );
set ground = body1;
body body1 < ( point4 );
point point5=point( 0.2 [ m ], 0 [ m ], 0.05 [ m ] );
point point6=point( 0.2 [ m ], 0 [ m ], -0.05 [ m ] );
point point7=point( 0.4 [ m ], 0 [ m ], -0.05 [ m ] );
point point8=point( 0.4 [ m ], 0 [ m ], 0.05 [ m ] );
point point9=point( 0.2 [ m ], 0.1 [ m ], 0.05 [ m ] );
point point10=point( 0.2 [ m ], 0.05 [ m ], 0 [ m ] );
line line2=polyLine( list( point5, point6, point7, point8, point5 ) );
solid solid1=extrude( line2, point5, point9, mass = m );
body body2=body( color = RGB( 229, 0, 229 ) );
body body2 < ( solid1, point10 );
force force1=spring( body1, point4, body2, point10, c );
sensor s=bodyDisplacement( body1, point10, projectX, body2, point10 );
sensor v=transVelocity( body1, projectX, body2, point10 );
function function1(v [ m/ s ])=mu*#v*#v;
force force2=damper( body1, point4, body2, point10, function1 );
condition condition1=transVelocity( body1, projectX, body2, point10, v0 );
event event1=reformsBySensor( list( stop( ) ), v, 0 [ m/ s ] );
sensor v0_theoretical=sqrt( c*m/2/mu/mu*( 1+( 2*mu*l/m-1 )*exp( 2*mu*l/m ) ) );
sensor v0_delta_rel=abs( ( v0 - v0_theoretical ) / v0_theoretical );
command command1=boundaryProblem( list( v0 ), list( s ), list( l ), 0.00001, 0.00001, 100, comDynamics );

```

Λ//

Λ Единицы измерения;  
set units = SI;