

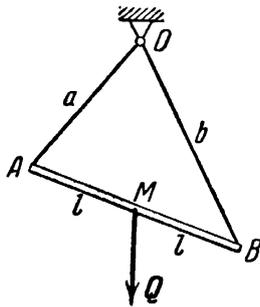
Задача 1

(Задача 32. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

Однородная палочка весом $Q=mg$ длиной $2l$ подвешена на двух нитях длиной a и b к неподвижной точке O . Определить натяжения нитей при равновесии.

Для решения задачи использовать следующие значения параметров:

$a = 3 \text{ м}$, $b = 5 \text{ м}$, $l = 2 \text{ м}$, $m = 10 \text{ кг}$.



Точный теоретический ответ

Натяжения нитей определяется по следующим формулам:

$$T1 = \frac{m \cdot g \cdot a}{\sqrt{2 \cdot a^2 + 2 \cdot b^2 - 4 \cdot l^2}};$$

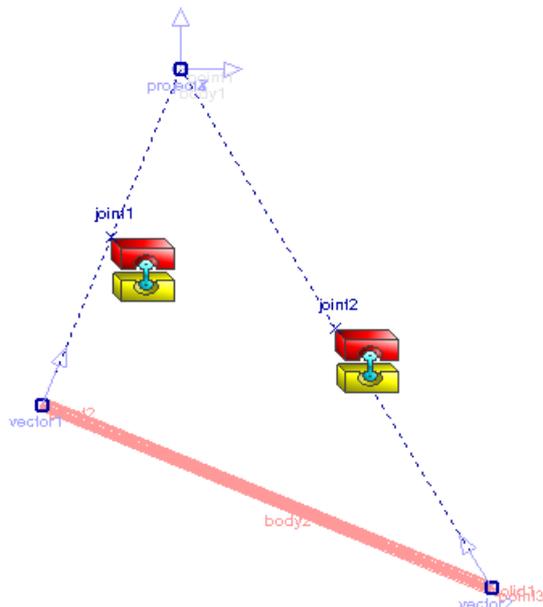
$$T2 = \frac{m \cdot g \cdot b}{\sqrt{2 \cdot a^2 + 2 \cdot b^2 - 4 \cdot l^2}}.$$

Для принятых значений параметров натяжения равны

$$T1 = 40.8121 \text{ Н};$$

$$T2 = 68.0201 \text{ Н}.$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте звено отображается точкой point1.
- Палочка (body2). Звено отображается цилиндром solid1 и имеет массу m .

Нити моделируются шарнирами joint1 и joint2 типа *тяга со сферическими наконечниками*. Для определения натяжения нитей созданы датчики сил T1 и T2. В проекте задана гравитация с ускорением свободного падения 9.81 [m/ s2].

Для определения равновесного состояния системы используется команда **Расчет равновесного состояния** в меню **Анализ**.

Результаты моделирования

При моделировании в EULER для принятых значений параметров получено

$$T1 = 40.8121 [N];$$

$$T2 = 68.0201 [N].$$

Относительное отличие от теоретического решения

$$T1_delta_rel = 0.000\ 000\ 012 [-];$$

$$T2_delta_rel = 0.000\ 000\ 006\ 5 [-].$$

Текст проекта в EULER

```
scalar a = 3 [ m ];
scalar b = 5 [ m ];
scalar l = 2 [ m ];
scalar m = 10 [ kg ];
scalar g = 9.81 [ m/ s2 ];
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], -a, 0 [ m ] );
point point3 = point( 2*l, -a, 0 [ m ] );
solid solid1 = cylinder( point2, point3, 0.05 [ m ], mass = m );
body body1 = body( color = RGB( 229, 229, 229 ) );
body body2 = body( color = RGB( 255, 153, 153 ) );
joint joint1 = sphericalSpherical( body1, point1, body2, point2 );
joint joint2 = sphericalSpherical( body1, point1, body2, point3 );
vector vector1 = vectorPP( point2, point1, point = point2 );
vector vector2 = vectorPP( point3, point1, point = point3 );
gravity gravity1 = parallel( reverse( projectY ), g = g );
sensor T1 = jointForce( force:, body2, point2, vector1, joint2, fixing = unlock: );
sensor T2 = jointForce( force:, body2, point3, vector2, joint3, fixing = unlock: );
sensor T1_theoretical = m*g*a/sqrt ( 2*a*a + 2*b*b - 4*l*l );
sensor T2_theoretical = m*g*b/sqrt ( 2*a*a + 2*b*b - 4*l*l );
sensor T1_delta_rel = abs( (T1 - T1_theoretical)/ T1_theoretical );
sensor T2_delta_rel = abs( (T2 - T2_theoretical)/ T2_theoretical );
```

```
////////////////////////////////////
```

```
Λ Прикрепление объектов
```

```
body body1 < (point1 );
```

```
body body2 < (solid1, point2, point3 );
```

```
////////////////////////////////////
```

```
Λ Инерциальное звено;
```

```
set ground = body1;
```

```
////////////////////////////////////
```

```
Λ Единицы измерения;
```

```
set units = SI;
```

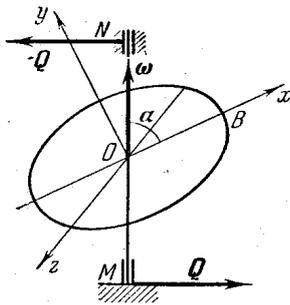
Задача 2

(Задача 845. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

Определить динамические реакции подшипников M и N вала, на который насажен однородный круглый диск радиуса R и массой m так, что его плоскость образует с осью вращения угол α , если угловая скорость постоянна и равна ω , $OM = a$, $ON = b$.

Для решения задачи использовать следующие значения параметров:

$R = 0.3 \text{ м}$, $m = 1 \text{ кг}$, $\alpha = 45 \text{ град}$, $\omega = 1 \text{ рад/с}$, $a = 0.5 \text{ м}$, $b = 0.3 \text{ м}$.



Точный теоретический ответ

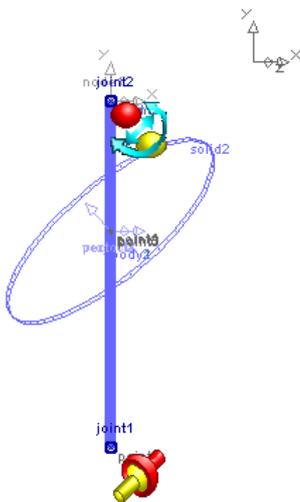
Динамические реакции в подшипниках M и N вала одинаковы и определяются по формуле

$$Q = \frac{m \cdot R^2 \cdot \omega^2 \cdot \sin 2\alpha}{8 \cdot (a + b)}$$

Для принятых значений параметров

$$Q = 0.0140625 \text{ Н.}$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Станина (body1). Станина является инерциальным звеном и в проекте не отображается.
- Вал (body2).

К валу прикреплены два тела: невесомая ось solid1 и наклонный диск solid2. Нижняя опора вала моделируется шарниром joint1 типа *сферический шарнир*. Верхняя опора вала моделируется шарниром joint2 типа *пользовательский шарнир*, который имеет все три вращательные степени свободы и одну поступательную вдоль оси вала. Поступательная

степень свободы в верхней опоре позволяет убрать статическую неопределимость восприятия нагрузки вдоль оси вала.

Для задания угловой скорости вращения вала используется объект *условие состояния механизма condition1*.

Динамические реакции опор определяются в горизонтальной плоскости XZ. Для определения реакции нижней опоры, созданы датчики сил F1x и F1z, действующих со стороны шарнира на опору в направлениях X и Z. Суммарная реакция нижней опоры определяется в датчике F1. Реакция верхней опоры определяется аналогично.

Для определения значений реакций опор достаточно произвести **Расчет сил и ускорений**.

Результаты моделирования

При получении точного теоретического решения предполагалось, что диск имеет нулевую толщину. При моделировании в EULER толщина диска solid2 должна быть больше нуля. Поэтому в зависимости от заданной в проекте относительной толщины диска (скаляр h_rel) решение в EULER будет отличаться от теоретического. Эта зависимость представлена в следующей таблице:

Относительная толщина диска h_rel [-]	Относительное отличие от теоретического решения Q_delta_rel [-]
0.1	0.013
0.01	0.000 13
0.001	0.000 001 3
0.0001	0.000 000 013

Текст проекта в EULER

```

scalar R = 0.3[m];
scalar m = 1[kg];
scalar alfa = 45[deg];
scalar w = 1[rad/s];
scalar a = 0.5[m];
scalar b = 0.3[m];
scalar h_rel = 0.01;
scalar h = h_rel*2*R;
point point1 = point( 0[m], -a, 0[m] );
point point2 = point( 0[m], b, 0[m] );
solid solid1 = cylinder( point1, point2, 0.01[m] );
point point3 = point( 0[m], 0[m], 0[m] );
vector vector1 = rotate( projectY, projectZ, alfa );
point point4 = move( point3, vector1, h/2 );
point point5 = move( point3, vector1, -h/2 );
solid solid2 = cylinder( point4, point5, R, mass = m );
color color1 = index( 54 );
body body1 = body( color = color1 );
color color2 = index( 64 );
body body2 = body( color = color2 );
joint joint1 = spherical( body1, body2, point1 );
node node1 = nodePoint( point2 );
joint joint2 = user( body1, body2, node1, noX:, yesY:, noZ:, yesFiX:, yesFiY:, yesFiZ: );
condition condition1 = rotVelocity( body1, projectY, body2, w );
sensor F1x = jointForce( force:, body1, point1, projectX, joint1 );
sensor F1z = jointForce( force:, body1, point1, projectZ, joint1 );
sensor F1 = sqrt(F1x*F1x+F1z*F1z);
sensor F2x = jointForce( force:, body1, point2, projectX, joint2 );
sensor F2z = jointForce( force:, body1, point2, projectZ, joint2 );
sensor F2 = sqrt(F2x*F2x+F2z*F2z);
sensor Q_theoretical = m*R*R*(w/1[rad])*(w/1[rad])*sin(2*alfa)/(8*(a+b));
sensor Q_delta_rel = abs((F1-Q_theoretical)/Q_theoretical);

```

```

//
Λ Прикрепление объектов
body body2 < (solid1, solid2 );

```

```

//
Λ Инерциальное звено;
set ground = body1;

```

```

//
Λ Единицы измерения;
set units = SI;

```

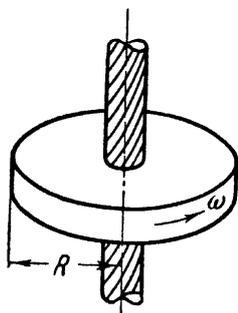

Задача 3

(Задача 959. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

На вертикально поставленный винт надета массивная гайка, имеющая форму полого цилиндра с радиусами r и R . Ей сообщена угловая скорость ω такого направления, что гайка начинает подниматься. На какую высоту z поднимется гайка, если шаг винта h , радиус r и трение отсутствует?

Для решения задачи использовать следующие значения параметров:

$$r = 0.02 \text{ м}, R = 0.2 \text{ м}, h = 0.01 \text{ м}, \omega = 1000 \text{ град/с.}$$



Точный теоретический ответ

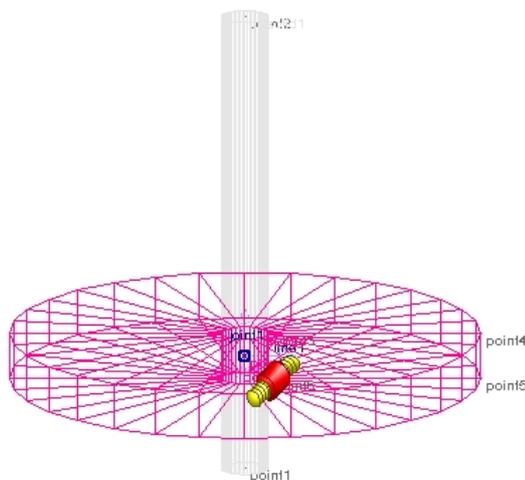
Высота, на которую поднимется гайка, определяется по формуле

$$z = \frac{\omega^2}{4 \cdot g} \cdot \left[R^2 + r^2 + \frac{h^2}{2 \cdot \pi^2} \right].$$

Для принятых значений параметров

$$z = 0.313662 \text{ м.}$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Винт (body1). Винт является инерциальным звеном и в проекте отображается цилиндром solid1.
- Гайка (body2). Гайка отображается телом вращения solid2 и имеет массу M.

Винт и гайка связаны шарниром joint1 типа *винтовая пара*. В проекте задана гравитация с ускорением свободного падения $9.81 \text{ [м/с}^2]$. Начальное вращение гайки задано с помощью объекта *условие состояния механизма condition1*. События event1 останавливает расчет, когда вертикальная скорость гайки станет равна нулю.

Для решения задачи необходимо установить время интегрирования равное 25 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 z_delta_rel [-]
0.1	0.000 001 9
0.01	0.000 000 009 2
0.001	0.000 000 000 82

Текст проекта в EULER

```
scalar r = 0.02 [ m ];
scalar R = 0.2 [ m ];
scalar w = 1000 [ deg/ s ];
scalar h = 0.01 [ m ];
scalar M = 1 [ kg ];
point point1 = point( 0 [ m ], -0.1 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], 0.3 [ m ], 0 [ m ] );
solid solid1 = cylinder( point1, point2, r );
point point3 = point( r, 0.02 [ m ], 0 [ m ] );
point point4 = point( R, 0.02 [ m ], 0 [ m ] );
point point5 = point( R, -0.02 [ m ], 0 [ m ] );
point point6 = point( r, -0.02 [ m ], 0 [ m ] );
line line1 = polyLine( list( point3, point4, point5, point6, point3 ) );
solid solid2 = spin( line1, projectY, mass = M );
body body1 = body( color = RGB( 229, 229, 229 ) );
body body2 = body( color = RGB( 255, 0, 153 ) );
point point7 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
joint joint1 = screw( body1, body2, point7, projectY, h );
gravity gravity1 = parallel( reverse( projectY ) );
condition condition1 = rotVelocity( body1, projectY, body2, w );
sensor v = derivative( joint1.s );
event event1 = reformsBySensor( list( stop( ) ), v, 0 [ m/ s ] );
sensor z = max( joint1.s );
sensor z_theoretical = 0.25*w*w*( R*R + r*r + 0.5*h*h/ Pi/ Pi )/ 9.81 [ m/ s2 ]/ 1 [ rad2 ];
sensor z_delta_rel = abs( ( z - z_theoretical )/ z_theoretical );
```

////////////////////////////////////

```
Λ Прикрепление объектов
body body1 < (solid1 );
body body2 < (solid2 );
```

////////////////////////////////////

```
Λ Инерциальное звено;
set ground = body1;
```

////////////////////////////////////

```
Λ Единицы измерения;
set units = SI;
```

Задача 4

(Задача 930. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

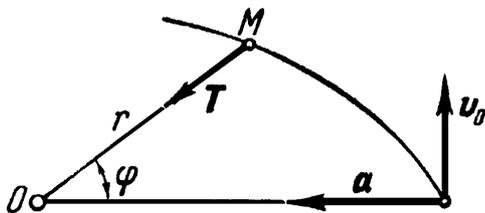
Тяжелый шарик M массы m находится на гладкой горизонтальной плоскости (плоскость чертежа) и прикреплен к гибкой нерастяжимой нити, которая втягивается с постоянной скоростью a в отверстие O .

Определить траекторию шарика и натяжение T нити, если в начальный момент он находился на расстоянии R от отверстия O и имел перпендикулярную к нити скорость v_0 .

При моделировании в EULER необходимо найти полярные координаты шарика M и натяжение нити в заданный момент времени t .

Для решения задачи использовать следующие значения параметров:

$$m = 1 \text{ кг}, R = 1 \text{ м}, a = 0.1 \text{ м/с}, v_0 = 0.2 \text{ м/с}, t = 5 \text{ с}.$$



Точный теоретический ответ

Полярные координаты шарика определяются по формулам

$$r = R - a \cdot t;$$

$$\varphi = \frac{v_0 \cdot t}{R - a \cdot t}.$$

Натяжение нити определяется по формуле

$$T = \frac{m \cdot v_0^2}{(R - a \cdot t)^3}.$$

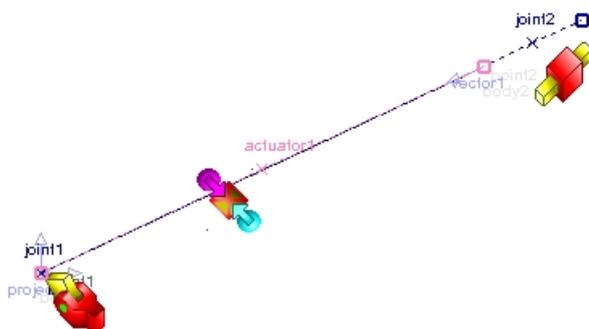
Для принятых значений параметров

$$R = 0.5 \text{ м};$$

$$\varphi = 114.592 \text{ град};$$

$$T = 0.32 \text{ Н}.$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Инерциальное звено (body1). В проекте звено не отображается.

- Шарик (body2). В проекте звено отображается точкой point2 и имеет массу m.
- Нить (body3). В проекте звено не отображается.

Вращение нити моделируется шарниром joint1 типа *пара вращения*. Перемещение шарика моделируется шарниром joint2 типа *поступательная пара*. Втягивание шарика с постоянной скоростью моделируется объектом *программное движение* motion1. Это программное движение имеет в своем составе *привод* actuator1. Начальная скорость шарика сообщается с помощью объекта *условие состояния механизма* condition1.

Для определения полярных координат используются датчики r и fi. Для определения натяжения нити используется датчик T типа *силовой фактор от привода*. Для остановки расчета в заданный момент времени используется событие event1.

Для определения значений требуемых параметров необходимо установить время интегрирования равное 5 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 r_delta_rel [-]	Относительное отличие от теоретического решения fi_delta_rel [-]	Относительное отличие от теоретического решения T_delta_rel [-]
0.1	0.000 000 000 000 000 44	0.000 099	0.000 29
0.01	0.000 25	0.000 50	0.000 75
0.001	0.000 000 000 000 077	0.000 000 010	0.000 000 030
0.000 1	0.000 000 000 000 72	0.000 000 000 073	0.000 000 000 22

Текст проекта в EULER

```

scalar m = 1 [ kg ];
scalar R = 1 [ m ];
scalar a = 0.1 [ m/ s ];
scalar v0 = 0.2 [ m/ s ];
scalar t = 5 [ s ];
body body1 = body( color = RGB( 229, 229, 229 ) );
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2 = point( R, 0 [ m ], 0 [ m ] );
line line1 = polyLine( list( point1, point2 ) );
body body2 = body( color = RGB( 229, 229, 229 ) );
body body3 = body( color = RGB( 229, 229, 229 ) );
MIP MIP1 = massPoint( point2, m );
joint joint1 = rotational( body1, body3, point1, projectZ );
vector vector1 = vectorPP( point2, point1, point = point2 );
joint joint2 = translational( body2, body3, point2, vector1 );
sensor r = bodyDisplacement( body2, point2, vector1, body1, point1, fixingVector = unlockVector: );
actuator actuator1 = forcePP( body1, point1, body2, point2 );
function function1(t [ s ])= -a*t;
sensor fi = angleVV( vectorPP( point1, point2 ), projectX, axisDirection = projectZ );
motion motion1 = ideal( actuator1, function1, joint2.s, time );
condition condition1 = rotVelocity( body1, projectZ, body2, v0/R*1[rad] );
sensor T = actuatorForce( force:, body2, point2, vector1, motion1, fixing = unlock: );
event event1 = reformsBySensor( list( stop( ) ), time, t );
sensor r_theoretical = R - a*t;
sensor fi_theoretical = v0*t/ ( R - a*t)*1 [ rad ];
sensor T_theoretical = m*v0*v0/ pow( ( R - a*t )/ 1 [ m ], 3 )/ 1 [ m ];
sensor r_delta_rel = abs( ( r - r_theoretical )/ r_theoretical );
sensor fi_delta_rel = abs( ( fi - fi_theoretical )/ fi_theoretical );
sensor T_delta_rel = abs( ( T - T_theoretical )/ T_theoretical );

```

```

////////////////////////////////////

```

```

Λ Прикрепление объектов
body body2 < (point2, MIP1 );

```

```

////////////////////////////////////

```

```

Λ Инерциальное звено;
set ground = body1;

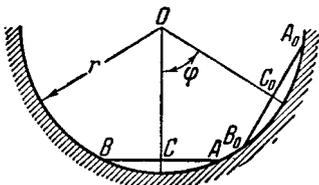
```


Задача 5

(Задача 949. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

Тяжелая однородная палочка веса $P=mg$ и длины $2a$ движется в шаровой чаше радиуса r , оставаясь в вертикальной плоскости, проходящей через центр O чаши. В начальном моменте палочка составляла с горизонтом угол φ и находилась в покое. Найти ее угловую скорость в момент прохождения через низшее положение.

Для решения задачи использовать следующие значения параметров:
 $r = 1 \text{ м}$, $a = 0.5 \text{ м}$, $\varphi = 45 \text{ град}$.



Точный теоретический ответ

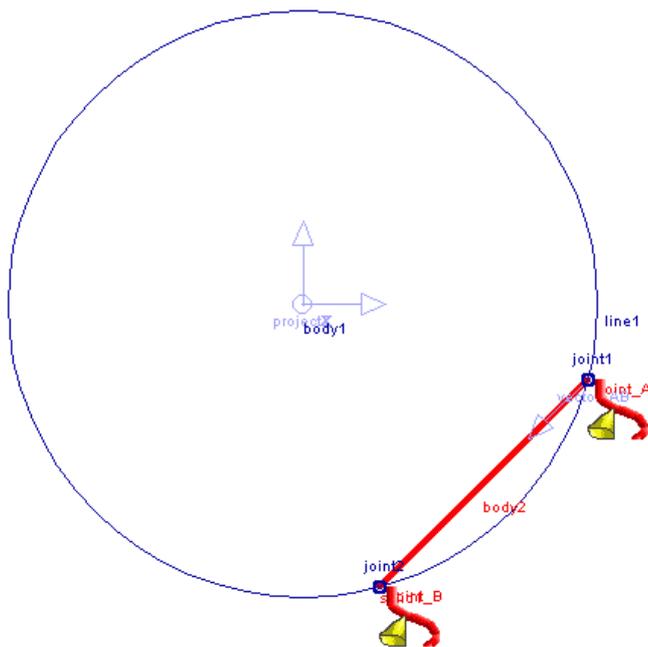
Угловая скорость палочки в низшем положении определяется по формуле

$$\omega = 2 \sin \frac{\varphi}{2} \sqrt{\frac{g \sqrt{r^2 - a^2}}{r^2 - \frac{2}{3} a^2}}.$$

Для принятых значений параметров

$$\omega = 2.44377 \text{ рад/с}.$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Чаша (body1). В проекте чаша является инерциальным звеном и отображается окружностью line1.
- Палочка (body2) Звено отображается цилиндром solid1 и имеет массу m.

Контакт палочки и чаши моделируется с помощью двух шарниров joint1 и joint2 типа контактная пара точка – линия. В проекте задана гравитация с ускорением свободного падения $9.81 \text{ [м/ с}^2\text{]}$.

Датчик w измеряет угловую скорость палочки. Датчик $angle$ измеряет угол палочки с горизонтом. Событие $event1$ останавливает расчет в момент, когда палочка проходит горизонтальное положение.

Для решения задачи необходимо установить время интегрирования равное 1 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

При получении точного теоретического решения предполагалось, что радиус палочки равен нулю. При моделировании в EULER радиус должен быть больше нуля. Поэтому решение в EULER будет зависеть от заданного в проекте относительно радиуса палочки (скаляр r_rel). Для постоянного шага интегрирования 0.001 [s] эта зависимость приведена в следующей таблице.

Относительный радиус палочки r_rel [-]	Относительное отличие от теоретического решения w_delta_rel [-]
0.1	0.000 37
0.01	0.000 003 4
0.001	0.000 000 36

Значение погрешности 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 при $r_rel=0.001$ w_delta_rel [-]
0.1	0.000 12
0.05	0.000 29
0.01	0.000 008 1
0.005	0.000 004 1
0.001	0.000 000 36

Текст проекта в EULER

```

scalar r = 1 [m];
scalar a = 0.5 [m];
scalar phi = 45 [deg];
scalar r_rel = 0.001;
scalar m = 1 [kg];
line line1 = circle( point( 0 [m], 0 [m], 0 [m] ), projectZ, r );
point point_A = pointCyl( -90 [deg] + phi + arcsin(a/r)*1[rad], r, 0 [m] );
point point_B = pointCyl( -90 [deg] + phi - arcsin(a/r)*1[rad], r, 0 [m] );
solid solid1 = cylinder( point_A, point_B, r_rel*a, mass = m );
vector vector_AB = vectorPP( point_A, point_B );
body body1 = body( color = index( 78 ) );
body body2 = body( color = index( 15 ) );
joint joint1 = contactPointLine( point_A, line1 );
joint joint2 = contactPointLine( point_B, line1 );
gravity gravity1 = parallel( reverse( projectY ) );
sensor angle = angleVV( reverse( projectX ), vector_AB, axisDirection = projectZ );
event event1 = reformsBySensor( list( stop( ) ), angle, 0 [rad] );
sensor w = rotVelocity( body1, reverse( projectZ ), body2 );
sensor w_theoretical = 2*sin(phi/2)*sqrt(9.81[m/s2]*sqrt(r*r-a*a))/(r*(2/3)*a*a)*1[rad];
sensor w_delta_rel = abs ((w -w_theoretical)/w_theoretical);

```

Λ//

```

Λ Прикрепление объектов
body body1 < (line1 );
body body2 < (solid1, point_A, point_B );

```

Λ//

```

Λ Инерциальное звено;

```

```
set ground = body1;
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ  Единицы измерения;
```

```
set units = SI;
```

Задача 6

(Задача 979. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

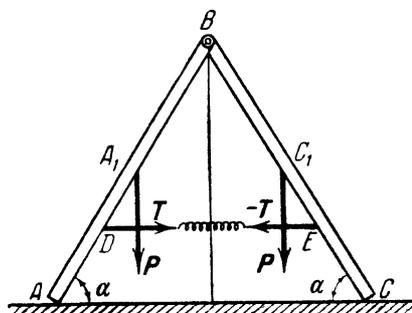
Две одинаковые невесомые доски AB и BC длиной a , соединенные в B шарниром, опираются на гладкий горизонтальный пол. В точках D и E на расстояниях $BD = BE = b$ доски связаны упругой нитью, натяжение T которой пропорционально удлинению:

$$T = k \cdot (l - l_0),$$

где l_0 длина нити в ненапряженном состоянии. Определить вес каждой из досок, если в положении равновесия они образуют с горизонтом угол α .

Для решения задачи использовать следующие значения параметров:

$$a = 0.5 \text{ м}, b = 0.3 \text{ м}, k = 100 \text{ Н/м}, l_0 = 0.1 \text{ м}, \alpha = 60 \text{ град.}$$



Точный теоретический ответ

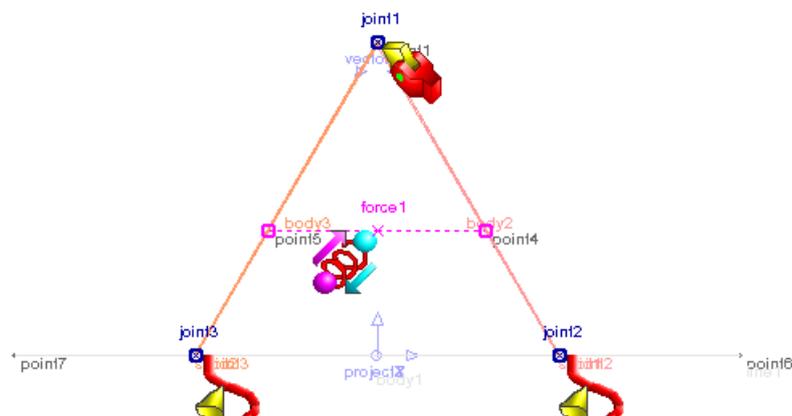
Вес каждой из досок определяется по формуле

$$P = \frac{2 \cdot b}{a} \cdot k \cdot (2 \cdot b \cdot \cos \alpha - l_0) \cdot \operatorname{tg} \alpha.$$

Для принятых значений параметров

$$P = 41.5692 \text{ Н}.$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Горизонтальный пол (body1). Пол является инерциальным звеном и в проекте отображается линией line1.
- Две доски (body2, body3). Доски отображаются цилиндрами solid1 и solid2. Каждая из досок имеет массу M .

Две доски соединены между собой шарниром joint1 типа пара вращения.

Взаимодействие между досками и полом моделируется двумя шарнирами joint2 и joint3 типа контактная пара точка-линия. Упругая нить моделируется силовым элементом

force1 типа *поступательная пружина с коэффициентом жесткости*. В проекте задана гравитация с ускорением свободного падения 9.81 [m/ s2].

Одним из признаков равновесного состояния системы является нулевое ускорение любой из точек досок. Для определения ускорения точки point2 используется датчик ускорения accel.

Для определения веса досок создана команда **Краевая задача** command1. Эта команда позволяет определить вес досок при помощи команды **Расчет сил и ускорений** для условия равенства нулю ускорения точки point2 (датчик accel).

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 P_delta_rel [-]
0.000 1	0.000 001 9
0.000 001	0.000 000 000 021

Текст проекта в EULER

```

scalar a = 0.5 [ m ];
scalar b = 0.3 [ m ];
scalar k = 100 [ N/ m ];
scalar l0 = 0.1 [ m ];
scalar alfa = 60 [ deg ];
scalar P = 40 [ N ];
scalar M = P/ 9.81 [ m/ s2 ];
point point1 = point( 0 [ m ], a*sin(alfa), 0 [ m ] );
vector vector1 = rotate( reverse( projectY ), projectZ, 90 [ deg ] - alfa, point = point1 );
vector vector2 = rotate( reverse( projectY ), projectZ, -(90 [ deg ] - alfa), point = point1 );
point point2 = move( point1, vector1, a );
point point3 = move( point1, vector2, a );
point point4 = move( point1, vector1, b );
point point5 = move( point1, vector2, b );
solid solid1 = cylinder( point1, point2, 0.001 [ m ], mass = M );
solid solid2 = cylinder( point1, point3, 0.001 [ m ], mass = M );
point point6 = point( 0.5 [ m ], 0 [ m ], 0 [ m ] );
point point7 = point( -0.5 [ m ], 0 [ m ], 0 [ m ] );
line line1 = polyLine( list( point6, point7 ) );
body body1 = body( color = RGB( 229, 229, 229 ) );
body body2 = body( color = RGB( 255, 153, 153 ) );
body body3 = body( color = RGB( 255, 153, 102 ) );
joint joint1 = rotational( body2, body3, point1, projectZ );
joint joint2 = contactPointLine( point2, line1 );
joint joint3 = contactPointLine( point3, line1 );
force force1 = spring( body2, point4, body3, point5, k, s0 = l0 );
gravity gravity1 = parallel( reverse( projectY ) );
sensor accel = acceleration( point2 );
sensor P_theoretical = 2*b*k*(2*b*cos(alfa)-l0)*tg(alfa)/ a;
sensor P_delta_rel = abs( ( P - P_theoretical )/ P_theoretical );

////////////////////////////////////
Λ Прикрепление объектов
body body1 < (line1 );
body body2 < (point2, solid1 );
body body3 < (point3, solid2 );

////////////////////////////////////
Λ Команды исследования
command command1 = boundaryProblem( list( P ), list( accel ), list( 0 [ m/ s2 ] ), 0.0001, 0.0001, 100, comForceAndAccelerate );

////////////////////////////////////
Λ Инерциальное звено;
set ground = body1;

////////////////////////////////////
Λ Единицы измерения;
set units = SI;

```

Задача 7

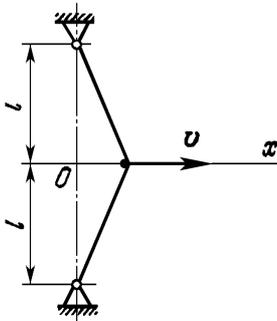
(Задача 8.9. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Материальная точка массы m , двигавшаяся прямолинейно по гладкой горизонтальной плоскости (совпадающей с плоскостью рисунка) с постоянной скоростью v_0 , в некоторый момент времени ($t = 0$) касается упругой нити в ее середине и при дальнейшем движении растягивает эту нить. В момент касания скорость v_0 точки перпендикулярна нити. Предварительное натяжение нити пренебрежимо мало и при ее растяжении возникает сила упругости $T = c \cdot \lambda$, где λ – деформация нити, $c = \text{const} > 0$.

Найти максимальную величину силы натяжения нити.

Для решения задачи использовать следующие значения параметров:

$$m = 1 \text{ кг}, v_0 = 2 \text{ м/с}, c = 8 \text{ Н/м}.$$



Точный теоретический ответ

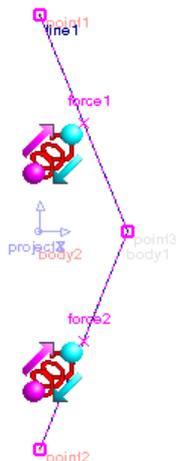
Максимальная величина силы натяжения нити определяется по формуле

$$T_{\max} = v_0 \cdot \sqrt{c \cdot m}.$$

Для принятых значений параметров

$$T_{\max} = 5.65685 \text{ м}.$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body2). В проекте инерциальное звено отображается двумя точками point1 и point2.
- Материальная точка (body1). Она отображается точкой point3 и имеет массу m .

Начальная скорость материальной точки задана с помощью объекта *условие состояния механизма* condition1. Упругая нить моделируется двумя силовыми элементами

force1 и force2 типа *поступательная пружина с коэффициентом жесткости*. Поскольку нить моделируется двумя силовыми элементами, то коэффициенты жесткости заданы равными $c1 = 2 \cdot c$. Для определения силы натяжения нити используется датчик силы T. Для фиксации максимального значения натяжения нити используется датчик Tmax типа *максимальное значение датчика*.

Для решения задачи необходимо установить время интегрирования равное 1 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 Tmax_delta_rel [-]
0.01	0.000 029
0.001	0.000 0026
0.000 1	0.000 000 015

Текст проекта в EULER

```

scalar m = 1 [ kg ];
scalar c = 8 [ N/ m ];
scalar c1 = 2*c;
scalar v0 = 2 [ m/ s ];
point point1 = point( 0 [ m ], 0.2 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], -0.2 [ m ], 0 [ m ] );
point point3 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
MIP MIP1 = massPoint( point3, m );
body body1 = body( color = RGB( 229, 229, 229 ) );
body body2 = body( color = RGB( 255, 153, 153 ) );
line line1 = polyLine( list( point1, point3, point2 ) );
condition condition1 = transVelocity( body2, projectX, body1, point3, v0 );
force force1 = spring( body2, point1, body1, point3, c1 );
force force2 = spring( body2, point2, body1, point3, c1 );
sensor T = force( force:, body1, point3, vectorPP( point3, point1 ), force1, fixing = unlock: );
sensor Tmax = max( T );
sensor Tmax_theoretical = v0*sqrt( c*m );
sensor Tmax_delta_rel = abs( ( Tmax - Tmax_theoretical ) / Tmax_theoretical );

```

```

////////////////////////////////////
Λ Прикрепление объектов
body body1 < (point3, MIP1);
body body2 < (point1, point2);

```

```

////////////////////////////////////
Λ Инерциальное звено;
set ground = body2;

```

```

////////////////////////////////////
Λ Единицы измерения;
set units = SI;

```

Задача 8

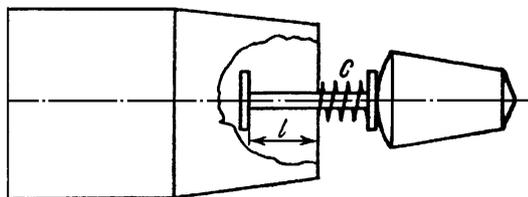
(Задача 9.66. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Для отделения искусственного спутника Земли массы m от последней ступени ракеты-носителя массы M используется пружинный толкатель, рабочим элементом которого является пружина, коэффициент жесткости которой равен c .

Определить относительную скорость V_r искусственного спутника Земли после разделения, если ход штока толкателя равен l и по окончании движения толкателя пружина не напряжена. Разделение происходит в пустоте после окончания работы двигательной установки последней ступени.

Для решения задачи использовать следующие значения параметров:

$$m = 150 \text{ кг}, M = 1500 \text{ кг}, c = 10000 \text{ Н/м}, l = 0.2 \text{ м}, V_0 = 8000 \text{ м/с}.$$



Точный теоретический ответ

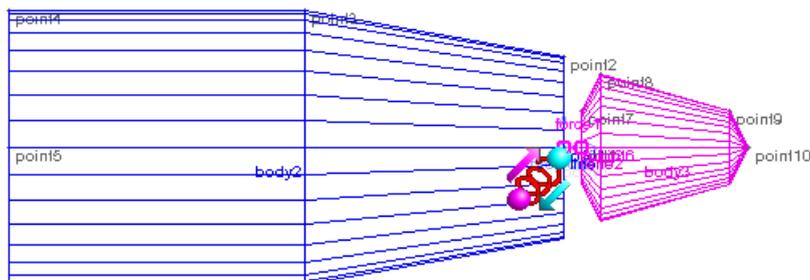
Относительная скорость искусственного спутника Земли определяется по формуле

$$V_r = l \cdot \sqrt{\frac{c \cdot (M + m)}{M \cdot m}}.$$

Для принятых значений параметров

$$V_r = 1.71270 \text{ м/с}.$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Инерциальное звено (body1). Звено в проекте не отображается.
- Ракета-носитель (body2). В проекте звено отображается телом вращения solid1 и имеет массу M
- Искусственный спутник Земли (body3). Звено отображается телом вращения solid2 и имеет массу m .

Пружинный толкатель моделируется силовым элементом force1 типа *поступательная пружина с коэффициентом жесткости*. Скорость ракеты-носителя вместе со спутником задается с помощью начальных условий condition1 и condition2. Для определения относительной скорости искусственного спутника Земли используется датчик скорости Vr. Остановка расчета происходит с помощью события event1.

Для решения задачи необходимо установить время интегрирования равное 1 [с] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 Vr_delta_rel [-]
0.01	0.000 12
0.001	0.000 001 7
0.000 1	0.000 000 012

Текст проекта в EULER

```
scalar m = 150 [ kg ];
scalar M = 1500 [ kg ];
scalar c = 10000 [ N/ m ];
scalar l = 0.2 [ m ];
scalar v0 = 8000 [ m/ s ];
body body1 = body( color = RGB( 229, 229, 229 ) );
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], 0.5 [ m ], 0 [ m ] );
point point3 = point( -1.4 [ m ], 0.75 [ m ], 0 [ m ] );
point point4 = point( -3 [ m ], 0.75 [ m ], 0 [ m ] );
point point5 = point( -3 [ m ], 0 [ m ], 0 [ m ] );
line line1 = polyLine( list( point1, point2, point3, point4, point5 ) );
solid solid1 = spin( line1, projectX, mass = M );
body body2 = body( color = RGB( 0, 0, 229 ) );
point point6 = point( 0.1 [ m ], 0 [ m ], 0 [ m ] );
point point7 = point( 0.1 [ m ], 0.2 [ m ], 0 [ m ] );
point point8 = point( 0.2 [ m ], 0.4 [ m ], 0 [ m ] );
point point9 = point( 0.9 [ m ], 0.2 [ m ], 0 [ m ] );
point point10 = point( 1 [ m ], 0 [ m ], 0 [ m ] );
line line2 = polyLine( list( point6, point7, point8, point9, point10 ) );
solid solid2 = spin( line2, projectX, mass = m );
body body3 = body( color = RGB( 229, 0, 229 ) );
force force1 = spring( body2, point1, body3, point6, c, s0 = l+0.1 [ m ] );
condition condition1 = bodiesFixation( list( body2, body3 ) );
condition condition2 = transVelocity( body1, projectX, body2, point1, v0 );
sensor s = bodyDisplacement( body2, point6, projectX, body3, point6 );
event event1 = reformsBySensor( list( complex( list( forcesOff( list( force1 ) ) ), s, l ) );
sensor Vr = transVelocity( body2, projectX, body3, point6 );
sensor Vr_theoretical = l*sqrt( c*( M + m ) / M / m );
sensor deltaVr_rel = abs( ( Vr - Vr_theoretical ) / Vr_theoretical );
```

```
////////////////////////////////////
```

```
Λ Прикрепление объектов
```

```
body body2 < (solid1 );
```

```
body body3 < (solid2 );
```

```
////////////////////////////////////
```

```
Λ Инерциальное звено;
```

```
set ground = body1;
```

```
////////////////////////////////////
```

```
Λ Единицы измерения;
```

```
set units = SI;
```

Задача 9

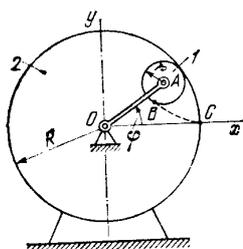
(Задача 8.34. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Кривошип OA вращается вокруг неподвижной оси, проходящей через точку O перпендикулярно плоскости рисунка. Он приводит в движение подвижную шестерню 1 радиуса r , которая обкатывается по внутренней поверхности неподвижной шестерни 2 радиуса R .

Найти уравнения движения точки B подвижной шестерни 1 в декартовых координатах, если угол поворота кривошипа $\varphi = \omega t$ ($\omega = \text{const} > 0$). В начальный момент времени ($t=0$) точка B совпадала с точкой C неподвижной шестерни 2, а кривошип OA был расположен на оси Ox .

При моделировании в EULER необходимо найти координаты точки B в заданный момент времени t . Для решения использовать следующие значения параметров:

$$r = 1 \text{ м}, R = 4 \text{ м}, \omega = 1 \text{ рад/с}, t = 10 \text{ с.}$$



Точный теоретический ответ

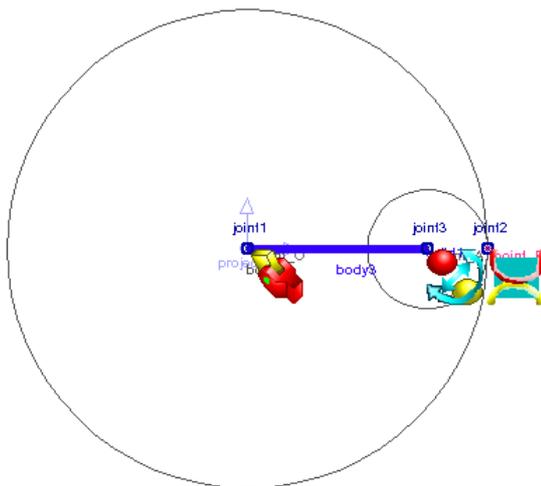
Траектория точки B описывается формулами

$$x = (R-r) \cos \omega t + r \cos \left[\frac{(R-r)}{r} \omega t \right],$$
$$y = (R-r) \sin \omega t - r \sin \left[\frac{(R-r)}{r} \omega t \right].$$

Для принятых значений параметров координаты точки B в момент времени 10 с равны

$$x = -2.36296 \text{ м};$$
$$y = -0.644032 \text{ м}.$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Подвижная шестерня (body1). В проекте звено отображается окружностью line1 с центром в точке point_A и радиусом r.
- неподвижная шестерня (body2). неподвижная шестерня является инерциальным звеном и отображается окружностью line2 с центром в точке point_O и радиусом R.
- Кривошип (body3). Звено отображается цилиндром solid1 с точками point_O и point_A в основаниях.

Кривошип и неподвижная шестерня соединены шарниром joint1 типа *пара вращения*. Взаимодействие подвижной шестерни и неподвижной шестерни описывается шарниром joint2 типа *контактная пара линия-линия в плоскости без проскальзывания*. Кривошип и подвижная шестерня соединены шарниром joint3 типа *пользовательский шарнир*. Он имеет две степени свободы - одну вращательную и одну поступательную вдоль оси кривошипа. Поступательная степень свободы необходима для того, чтобы избежать заклинивания системы в процессе движения.

Начальное вращение кривошипа задается с помощью объекта *условие состояния механизма condition1*. В дальнейшем процессе движения угловая скорость кривошипа не изменяется. Для определения координат точки point_B используются датчики X и Y. Остановка расчета в заданный момент времени осуществляется с помощью события event1.

Для решения задачи необходимо установить время интегрирования равное 10 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Интегрирование системы проводилось с переменным шагом со следующими параметрами:

- начальный шаг = 5.0e-5 [s],
- минимальный шаг = 1.0e-8 [s],
- максимальный шаг = 5.0e-1 [s],
- относительная и абсолютная погрешности интегрирования задавались равными и варьировались.

Результаты моделирования представлены в следующей таблице.

Задаваемые значения абс. и отн. погрешностей интегрирования [s]	Относительное отличие от теоретического решения X_delta_rel [-]	Относительное отличие от теоретического решения Y_delta_rel [-]
0.000 1	0.002 3	0.005 5
0.000 01	0.000 44	0.002 1
0.000 001	0.000 17	0.000 55
0.000 000 1	0.000 012	0.000 000 58
0.000 000 01	0.000 013	0.000 008 3
0.000 000 001	0.000 000 36	0.000 000 64

Текст проекта в EULER

```

scalar r = 1 [m];
scalar R = 4 [m];
scalar w = 1 [rad/s];
scalar t = 10 [s];
point point_O = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point_A = move( point_O, projectX, R-r );
line line1 = circle( point_A, projectZ, r );
line line2 = circle( point_O, projectZ, R );
solid solid1 = cylinder( point_O, point_A, 0.05 [m], mass = 1 [ kg ] );
point point_B = move( point_O, projectX, R, pointStyle = thickdot:, color = index( 84 ) );
color color1 = index( 7 );
body body1 = body( color = color1 );
body body2 = body( color = color1 );
color color2 = index( 76 );
body body3 = body( color = color2 );
joint joint1 = rotational( body2, body3, point_O, projectZ );
joint joint2 = flatContactLineLine( line1, line2, lines_slip = linesNoSlip: );

```

```
joint joint3 = user( body3, body1, nodePoint( point_A ), yesX:, noY:, noZ:, noFiX:, noFiY:, yesFiZ: );
condition condition1 = sensorValue( derivative( joint1.gamma ), w);
sensor X = bodyDisplacement( body2, point_O, projectX, body1, point_B );
sensor Y = bodyDisplacement( body2, point_O, projectY, body1, point_B );
event event1 = reformsBySensor( list( stop( ) ), time, t );
sensor X_theoretical = (R-r)*cos(w*t) + r*cos((R-r)/r*w*t);
sensor Y_theoretical = (R-r)*sin(w*t) - r*sin((R-r)/r*w*t);
sensor X_delta_rel = abs ((X -X_theoretical)/X_theoretical);
sensor Y_delta_rel = abs ((Y -Y_theoretical)/Y_theoretical);
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Прикрепление объектов
body body1 < (line1, point_B );
body body2 < (line2 );
body body3 < (solid1 );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Инерциальное звено;
set ground = body2;
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Единицы измерения;
set units = SI;
```

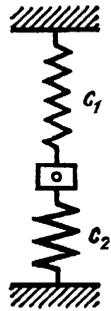
Задача 10

(Задача 32.26. Задачи по теоретической механике. И.В. Мещерский.
СПб.: Издательство «Лань», 2001. – 448 с.)

Определить период свободных колебаний груза массой m , зажатого между двумя пружинами с коэффициентами жесткости c_1 и c_2 .

Для решения задачи использовать следующие значения параметров:

$$m = 10 \text{ кг}, \quad c_1 = 300 \text{ Н/м}, \quad c_2 = 500 \text{ Н/м}$$



Точный теоретический ответ

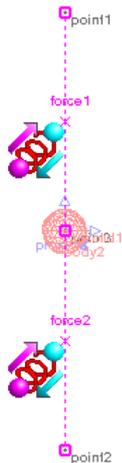
Период определяется по следующей формуле:

$$T = 2\pi \sqrt{\frac{m}{c_1 + c_2}}.$$

Для принятых значений параметров

$$T = 0.702481 \text{ с.}$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте звено не отображается.
- Груз (body2). Звено отображается сферой solid1 и имеет массу m .

Пружины моделируются силовыми элементами force1 и force2 типа *поступательная пружина с коэффициентом жесткости*. Для того чтобы груз начал совершать колебания, ему сообщается начальная скорость с помощью объекта *условие состояния механизма condition1*. Для определения смещения и скорости груза используются соответственно датчики s и v .

Период колебаний определяется как промежуток времени, за которое груз возвращается в первоначальное положение и при этом имеет положительную скорость (как в начальный момент). Для определения периода создано событие event1. Оно останавливает расчет, когда груз проходит первоначальное положение и при этом выполняются условия: время больше и груз имеет положительную скорость. В момент

остановки расчета датчик time будет иметь значение равное периоду свободных колебаний груза.

Для решения задачи необходимо установить время интегрирования равное 1 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 T_delta_rel
0.1	0.003 9
0.01	0.000 000 61
0.001	0.000 000 000 26

Текст проекта в EULER

```

scalar m = 10 [kg];
scalar c1 = 300 [kg/ s2 ];
scalar c2 = 500 [kg/ s2 ];
point point1 = point( 0 [ m ], 1 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], -1 [ m ], 0 [ m ] );
body body1 = body( color = RGB( 229, 229, 229 ) );
point point3 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
solid solid1 = sphere( point3, 0.1 [ m ], mass = m );
body body2 = body( color = RGB( 255, 153, 153 ) );
force force1 = spring( body1, point1, body2, point3, c1 );
force force2 = spring( body1, point2, body2, point3, c2 );
sensor s = bodyDisplacement( body1, point3, projectY, body2, point3 );
scalar v0 = 2 [ m/ s ];
condition condition1 = transVelocity( body1, projectY, body2, point3, v0 );
sensor v = transVelocity( body1, projectY, body2, point3 );
event event1 = reformsBySensor( list( stop( ) ), s, 0 [ m ], logCondition = list( ( " ( time > 0 [ s ] ) & ( v > 0 [ m/ s ] )" ) ) );
sensor T_theoretical = 2*PI*sqrt(m/(c1+c2));
sensor T_delta_rel = abs( ( time - T_theoretical ) / T_theoretical );

```

```

////////////////////////////////////
Λ Прикрепление объектов
body body2 < (solid1 );

```

```

////////////////////////////////////
Λ Инерциальное звено;
set ground = body1;

```

```

////////////////////////////////////
Λ Единицы измерения;
set units = SI;

```

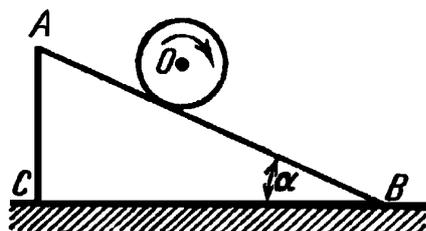
Задача 11

(Задача 48.29. Задачи по теоретической механике. И.В. Мещерский. СПб.: Издательство «Лань», 2001. – 448 с.)

На гладкой горизонтальной плоскости помещена треугольная призма ABC массой m_1 и углом $ABC = \alpha$, которая может скользить без трения по этой плоскости. По грани призмы AB катится без скольжения однородный круглый цилиндр массы m_2 . Определить величину ускорения призмы.

Для решения задачи использовать следующие значения параметров:

$$m_1 = 8 \text{ кг}, m_2 = 2 \text{ кг}, \alpha = 30 \text{ град.}$$



Точный теоретический ответ

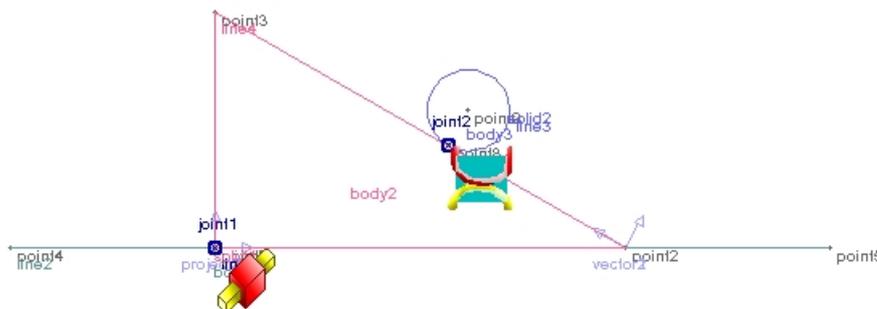
Ускорение определяется по формуле

$$a = \frac{m_2 \sin 2\alpha}{3(m_1 + m_2) - 2m_2 \cos^2 \alpha} \cdot g.$$

Для принятых значений параметров

$$a = 0.629312 \text{ м/с}^2.$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Плоскость (body1). Плоскость является инерциальным звеном и в проекте отображается прямой линией line2.
- Призма (body2). Звено отображается телом выдавливания и имеет массу m_1 .
- Цилиндр (body3). Звено отображается цилиндром и имеет массу m_2 .

Взаимодействие призмы с плоскостью моделируется шарниром *поступательная пара* joint1. Качение цилиндра по призме моделируется шарниром joint2 типа *контактная пара линия-линия в плоскости* без проскальзывания. В качестве контактируемых линий этого шарнира заданы контуры цилиндра line3 и призмы line4, которые прикреплены к соответствующим звеньям.

Для определения ускорения призмы в проекте создан датчик ускорения accel. Для определения его значения достаточно произвести **Расчет сил и ускорений**.

Результаты моделирования

При моделировании в EULER для принятых значений параметров получено $accel = 0.629318 \text{ [м/с}^2\text{]}$.

Относительное отличие от теоретического решения

$accel_delta_rel = 0.000\ 009\ 3 [-]$.

Текст проекта в EULER

```
scalar m1 = 8[kg];
scalar m2 = 2[kg];
scalar a = 30 [deg];
scalar r = 0.2[m];
scalar g = 9.81[m/s2];
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2 = point( 2 [ m ], 0 [ m ], 0 [ m ] );
point point3 = point( 0 [ m ], 2 [ m ]*tg(a), 0 [ m ] );
line line1 = polyLine( list( point1, point2, point3, point1 ) );
point point4 = point( -1 [ m ], 0 [ m ], 0 [ m ] );
point point5 = point( 3 [ m ], 0 [ m ], 0 [ m ] );
line line2 = polyLine( list( point4, point5 ) );
color color1 = index( 58 );
body body1 = body( color = color1 );
point point6 = point( 0 [ m ], 0 [ m ], 0.1 [ m ] );
point point7 = point( 0 [ m ], 0 [ m ], -0.1 [ m ] );
solid solid1 = extrude( line1, point6, point7, mass = m1 );
color color2 = index( 82 );
body body2 = body( color = color2 );
vector vector1 = vectorPP( point2, point3 );
point point8 = move( point2, vector1, 1[m] );
vector vector2 = rotate( vector1, projectZ, -90[deg] );
point point9 = move( point8, vector2, r );
line line3 = circle( point9, projectZ, r );
solid solid2 = extrude( line3, point6, point7, mass = m2 );
color color3 = index( 64 );
body body3 = body( color = color3 );
joint joint1 = translational( body1, body2, point1, projectX );
line line4 = polyLine( list( point3, point2 ) );
joint joint2 = flatContactLineLine( line4, line3, lines_slip = linesNoSlip );
gravity gravity = parallel( reverse( projectY ), g = g );
sensor v = derivative( joint1.s );
sensor accel = derivative( v );
sensor accel_theoretical = -m2*sin(2*a)/(3*(m1+m2)-2*m2*cos(a)*cos(a))*g;
sensor accel_delta_rel = abs((accel-accel_theoretical)/accel_theoretical);
```

Λ//

Λ Прикрепление объектов

```
body body1 < (line2 );
body body2 < (solid1, line4 );
body body3 < (solid2, line3 );
```

Λ//

Λ Инерциальное звено;

```
set ground = body1;
```

Λ//

Λ Единицы измерения;

```
set units = SI;
```

Задача 12

(Задача 8.54. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

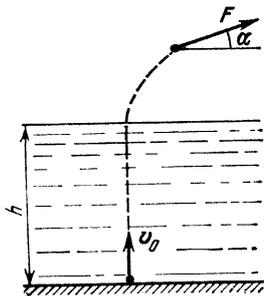
Тело массы m начинает движение в воде на глубине h с вертикальной скоростью v_0 и движется вверх, преодолевая силу \mathbf{R}_1 сопротивления воды. Модуль этой силы $R_1 = \mu_1 v^2$, где v – скорость тела, $\mu_1 = \text{const} > 0$. После выхода тела из воды на него в течении T секунд действует сила $\mathbf{F} = \text{const}$, линия действия которой в течении всего отрезка времени наклонена под углом $\alpha = 30^\circ$ к горизонту. Сила сопротивления воздуха силы $\mathbf{R}_2 = -\mu_2 \mathbf{v}$, где $\mu_2 = \text{const} > 0$.

Пренебрегая выталкивающей силой, действующей на тело в воде, найти высоту H его подъема над поверхностью воды в момент окончания действия силы \mathbf{F} , если $F = 4mg$,

$\mu_2 = 0.5\mu_1 v_0$, $h = \frac{m}{2\mu_1} \ln \frac{\mu_1 v_0^2}{mg}$ и поле сил тяжести однородно.

Для решения задачи использовать следующие значения параметров:

$m = 10$ кг, $v_0 = 100$ м/с, $c = 10$ Н/м, $\alpha = 30$ град, $\mu_1 = 1$ Нс²/м², $T = 0.5$ с.



Точный теоретический ответ

Высота определяется по формуле

$$H = \frac{m^2 \cdot g}{2 \cdot \mu_2^2} \cdot \left(e^{\frac{-\mu_2 \cdot T}{m}} - 1 \right) + \frac{m \cdot g}{\mu_2} \cdot T.$$

Для принятых значений параметров

$$H = 0.800905 \text{ м.}$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Инерциальное звено (body1). В проекте инерциальное звено не отображается.
- Тело (body2). Тело изображается точкой point1 и имеет массу m.

В проекте задана гравитация с ускорением свободного падения 9.81 [m/ s2].

Начальная скорость телу сообщается с помощью объекта *условие состояния механизма condition1*. Сила сопротивления воды моделируется силовым элементом force1 типа *сила по датчикам на одну точку* с функцией силы function1. Сила, толкающая тело, моделируется силовым элементом force2 типа *сила по датчикам на одну точку*. Сила сопротивления воздуха моделируется силовыми элементами force3 и force4 типа *сила по датчикам на одну точку* с функцией силы function2. Момент выхода тела из воды моделируется событием event1. Оно выключает силовой элемент force1 и включает силовые элементы force2, force3, force4 и датчик относительного времени event1.timer. Для определения высоты тела используется датчик H. Для остановки расчета используется событие event2.

Для решения задачи необходимо установить время интегрирования равное 2 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 H_delta_rel [-]
0.01	0.000 059
0.001	0.000 29
0.000 1	0.000 029
0.000 01	0.000 001 2
0.000 001	0.000 000 18

Текст проекта в EULER

```

scalar m = 10 [ kg ];
scalar v0 = 100 [ m/ s ];
scalar T = 0.5 [ s ];
scalar a = 30 [ deg ];
scalar mu1 = 1 [ N s2/ m2 ];
scalar mu2 = 0.5*mu1*v0;
scalar g = 9.81 [ m/ s2 ];
scalar h = 0.5*m*ln( mu1*v0*v0/ m/ g )/ mu1;
scalar F = 4*m*g;
body body1 = body( color = RGB( 229, 229, 229 ) );
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
MIP MIP1 = massPoint( point1, m );
body body2 = body( color = RGB( 255, 153, 153 ) );
point point2 = point( 0 [ m ], h, 0 [ m ] );
gravity gravity1 = parallel( reverse( projectY ), g = g );
condition condition1 = transVelocity( body1, projectY, body2, point1, v0 );
function function1(v [ m/ s ])= -mu1*#v*#v;
sensor vx = transVelocity( body1, projectX, body2, point1 );
sensor vy = transVelocity( body1, projectY, body2, point1 );
force force1 = force( body2, point1, projectY, function1, list( vy ) );
vector vector1 = rotate( projectX, projectZ, a );
function function2(v [ m/ s ])= -mu2*#v;
force force2 = force( body2, point1, vector1, F, list( ), work = off: );
force force3 = force( body2, point1, projectX, function2, list( vx ), work = off: );
force force4 = force( body2, point1, projectY, function2, list( vy ), work = off: );
sensor H = bodyDisplacement( body1, point2, projectY, body2, point1 );
event event1 = reformsBySensor( list( complex( list( forcesOff( list( force1 ) ), forcesOn( list( force2, force3, force4 ) ) ) ) ), H,
0 [ m ], timer = timerOn: );
event event2 = reformsBySensor( list( stop( ) ), event1.timer, T );
sensor H_theoretical = 0.5*m*m*g/ mu2/ mu2*( exp( -mu2*T/ m ) - 1 )+m*g*T/ mu2;
sensor H_delta_rel = abs( ( H -H_theoretical )/ H_theoretical );

```

////////////////////////////////////

Λ Прикрепление объектов
body body2 < (point1, MIP1);

Задача 13

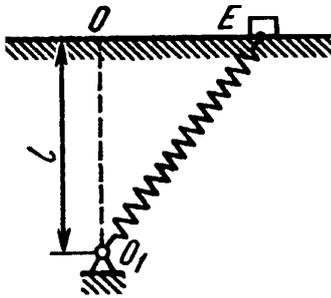
(Задача 30.2. Задачи по теоретической механике. И.В. Мещерский.
СПб.: Издательство «Лань», 2001. – 448 с.)

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

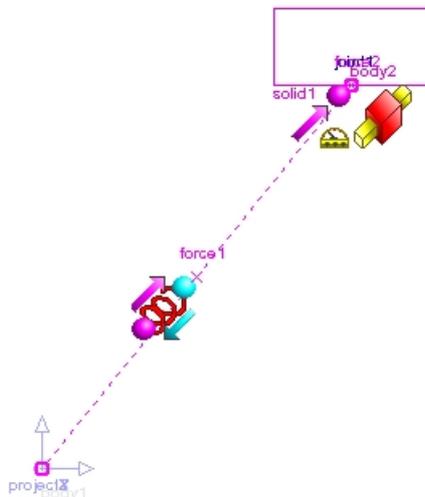
Скорость тела определяется по формуле

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

Для принятых значений параметров

$$v = 1.12336 \text{ м/с}.$$

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 $9.81 \text{ [м/с}^2]$. 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 необходимо установить время интегрирования равное 1 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

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

Шаг интегрирования [s]	Точность решения теоретического уравнения V_delta_rel [-]
0.001	0.000 28
0.000 1	0.000 028
0.000 01	0.000 004 2
0.000 001	0.000 000 57

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 ) );
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 ) );
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+10*( l-sqrt( l*l+a*a ) ) )-f*( a*( m*g+c*l )+c*10*l*ln( l/( a+sqrt( l*l+a*a ) ) ) ) ) );
sensor v_delta_rel = abs( ( v - v_theoretical ) / v_theoretical );

```

```

////////////////////////////////////
Λ Прикрепление объектов
body body2 < (solid1 );

```

```

////////////////////////////////////
Λ Инерциальное звено;
set ground = body1;

```

```

////////////////////////////////////
Λ Единицы измерения;
set units = SI;

```

Задача 14

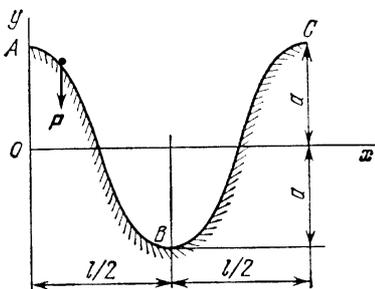
(Задача 8.12. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Материальная точка массы m начинает двигаться без начальной скорости из точки A по гладкой направляющей, уравнение которой $y = \cos \frac{2\pi x}{l}$.

Определить силу давления точки на направляющую в тот момент, когда она приходит через точку B .

Для решения задачи использовать следующие значения параметров:

$m = 1 \text{ кг}$, $a = 0.2 \text{ м}$, $l = 0.4 \text{ м}$.



Точный теоретический ответ

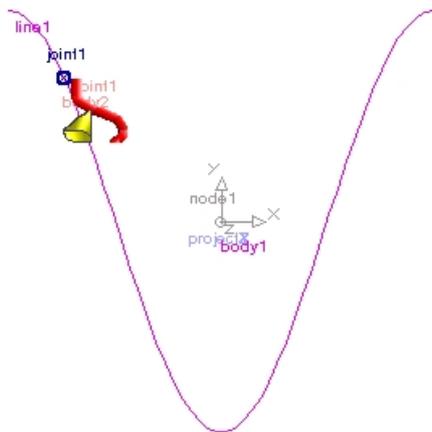
Сила давления точки на направляющую в нижней точке определяется по формуле

$$N = m \cdot g \cdot \left(1 + 16 \cdot \pi^2 \cdot \frac{a^2}{l^2} \right).$$

Для принятых значений параметров

$$N = 397.093 \text{ Н}.$$

Решение задачи в EULER



Система состоит из двух звеньев.

- Направляющая (body1). В проекте звено отображается линией line1. Уравнение линии задано с помощью функций function1 и function2.
- Материальная точка (body2). Она изображается точкой point1 и имеет массу m .

Движение материальной точки по направляющей моделируется шарниром joint1 типа *контактная пара точка-линия*. Для того чтобы точка начала движение в проекте задано начальное условие condition1 с малой скоростью $v0$. В проекте задана гравитация с

set units = SI;

Задача 15

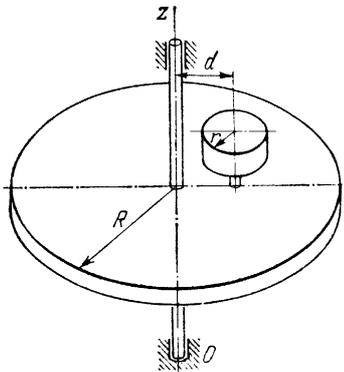
(Задача 9.20. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Горизонтальная платформа, представляющая собой однородный диск радиуса R и массы M , вращается вокруг вертикальной оси с угловой скоростью ω_0 . На платформе на расстоянии $d=0.4R$ от оси вращения находится маховик, ось которого вертикальна. Он представляет собой однородный диск радиуса $r=0.2R$ и массы $m=0.1M$. В начальный момент маховик относительно платформы не вращался.

До какой угловой скорости ω_r относительно платформы необходимо раскрутить маховик, чтобы платформа остановилась? Сопротивлением вращению пренебречь.

Для решения задачи использовать следующие значения параметров:

$M = 50$ кг, $R = 0.5$ м, $\omega_0 = 10$ град/с.



Точный теоретический ответ

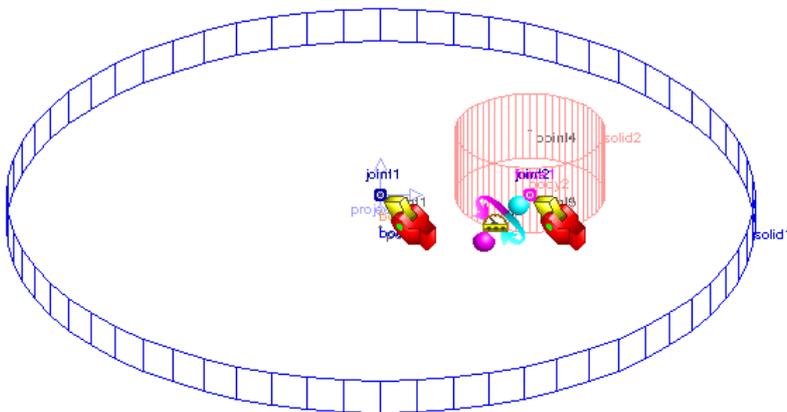
Относительная скорость маховика определяется по формуле

$$\omega_r = \left(\frac{M}{m} \cdot \frac{R^2}{r^2} + \frac{2 \cdot d^2}{r^2} + 1 \right) \cdot \omega_0.$$

Для принятых значений параметров

$$\omega_r = 2590.00 \text{ град/с.}$$

Решение задачи в EULER



Система состоит из трех звеньев.

- Инерциальное звено (body3). Звено в проекте не отображается.
- Платформа (body1). В проекте звено отображается цилиндром solid1 с радиусом R и имеет массу M

- Маховик (body2). В проекте звено отображается цилиндром solid2 с радиусом r и имеет массу m .

Звенья соединены между собой шарнирами joint1 и joint2 типа *пара вращения*.

Начальная угловая скорость платформе сообщается с помощью объекта *условие состояния механизма* condition1. Для раскрутки маховика используется силовой элемент force1 типа *момент по датчикам на две точки*. Для определения угловых скоростей платформы и маховика используются датчики W и Wr. События event1 останавливает расчет, когда угловая скорость платформы станет равна нулю.

Для решения задачи необходимо установить время интегрирования равное 2 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

Относительное отличие 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 Wr_delta_rel [-]
0.1	0.000 000 000 000 020
0.01	0.000 000 000 000 018

Текст проекта в EULER

```

scalar R = 0.5 [ m ];
scalar M = 50 [ kg ];
scalar w0 = 10 [ deg/ s ];
scalar d = 0.4*R;
scalar r = 0.2*R;
scalar m = 0.1*M;
point point1 = point( 0 [ m ], 0 [ m ], 00 [ m ] );
point point2 = point( 0 [ m ], -0.05 [ m ], 0 [ m ] );
solid solid1 = cylinder( point1, point2, R, mass = M );
body body1 = body( color = RGB( 0, 0, 229 ) );
point point3 = point( d, 0 [ m ], 0 [ m ] );
point point4 = point( d, 0.1 [ m ], 0 [ m ] );
point point5 = point3;
solid solid2 = cylinder( point3, point4, r, mass = m );
body body2 = body( color = RGB( 255, 153, 153 ) );
body body3 = body( color = RGB( 255, 153, 102 ) );
joint joint1 = rotational( body3, body1, point1, projectY );
joint joint2 = rotational( body1, body2, point3, projectY );
condition condition1 = rotVelocity( body3, projectY, body1, w0 );
force force1 = moment2( body1, point3, projectY, body2, point3, -1 [ kg m2/ s2 ], list( ) );
sensor w = rotVelocity( body3, projectY, body1 );
sensor Wr = rotVelocity( body1, projectY, body2 );
event event1 = reformsBySensor( list( stop( ) ), w, 0 [ deg/ s ] );
sensor Wr_theoretical = ( M/ m*R*R/ r/ r + 2*d*d/ r/ r + 1 )*w0;
sensor Wr_delta_rel = abs( ( Wr - Wr_theoretical ) / Wr_theoretical );

```

```

////////////////////////////////////
Λ Прикрепление объектов
body body1 < (solid1 );
body body2 < (solid2 );

```

```

////////////////////////////////////
Λ Инерциальное звено;
set ground = body3;

```

```

////////////////////////////////////
Λ Единицы измерения;
set units = SI;

```

Задача 16

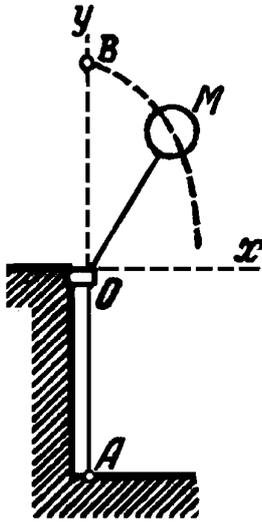
(Задача 27.55. Задачи по теоретической механике. И.В. Мещерский. СПб.: Издательство «Лань», 2001. – 448 с.)

Упругая нить, закрепленная в точке A , проходит через неподвижное гладкое кольцо O ; к свободному концу ее прикреплен шарик M , масса которого равна m . Длина невытянутой нити $l=AO$; для удлинения нити на l м нужно приложить силу $k^2 \cdot m$. Вытянув нить по прямой AB так, что ее длина увеличилась вдвое, сообщили шарiku скорость v_0 , перпендикулярную прямой AB . Определить траекторию шарика, пренебрегая действием силы тяжести и считая натяжение нити пропорциональным ее удлинению.

При моделировании в EULER необходимо найти координату x центра шарика в тот момент, когда координата $y = y_1$.

Для решения использовать следующие значения параметров:

$$m = 2 \text{ кг}, l = 1 \text{ м}, k = 2 \text{ 1/с}, v_0 = 1 \text{ м/с}, y_1 = 0.5 \text{ м}.$$



Точный теоретический ответ

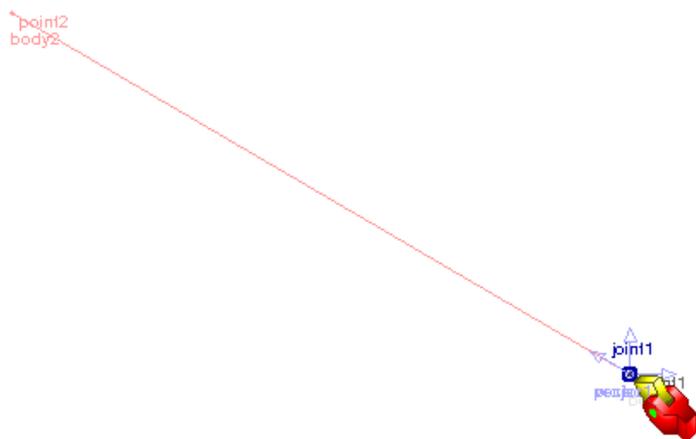
Координата x центра масс шарика определяется по формуле

$$x = \frac{v_0}{k} \cdot \sqrt{\left(1 - \frac{y_1^2}{l^2}\right)}.$$

Для принятых значений параметров

$$x = 0.433013 \text{ м}.$$

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 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 определения значений требуемых параметров необходимо установить время интегрирования равное 5 [s] и выполнить команду **Расчет динамики движения**.

Результаты моделирования

При получении точного теоретического решения предполагалось, что шарик начинает движение из положения неустойчивого равновесия с ничтожно малой скоростью. При моделировании в EULER шарик должен иметь ненулевую начальную скорость. Поэтому решение в EULER будет зависеть от заданной в проекте начальной скорости. Для постоянного шага интегрирования 0.0001 [s] эта зависимость 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 6	0.000 020	0.000 004 1	0.000 011
0.001	0.000 000 035	0.000 000 20	0.000 000 056	0.000 000 11

Относительное отличие решения задачи в EULER в зависимости от шага численного интегрирования (использовался постоянный шаг интегрирования) при начальной скорости $v_0=0.001 \text{ [m/ s]}$ представлено 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	0.000 098	0.000 000 20	0.000 16	0.000 012
0.001	0.000 000 83	0.000 000 20	0.000 001 3	0.000 000 24
0.000 1	0.000 000 035	0.000 000 20	0.000 000 056	0.000 000 11

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;
point point1 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
point point2 = point( 0 [ m ], l, 0 [ m ] );
line line1 = polyLine( list( point1, point2 ) );
MIP MIP1 = massPoint( point2, m );
body body1 = body( color = RGB( 229, 229, 229 ) );
body body2 = body( color = RGB( 255, 153, 153 ) );
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 );

```

```

////////////////////////////////////

```

```

Λ Прикрепление объектов
body body2 < (point2, MIP1);

```

```

////////////////////////////////////

```

```

Λ Инерциальное звено;
set ground = body1;

```

```

////////////////////////////////////

```

```

Λ Единицы измерения;
set units = SI;

```

Задача 18

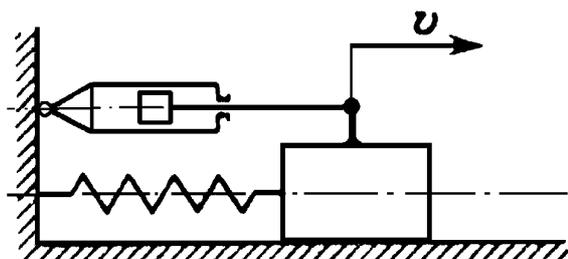
(Задача 8.7. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

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 λ , т. е. $F = c \cdot \lambda$, где $c = \text{const} > 0$. 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, $\mu = \text{const} > 0$.

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

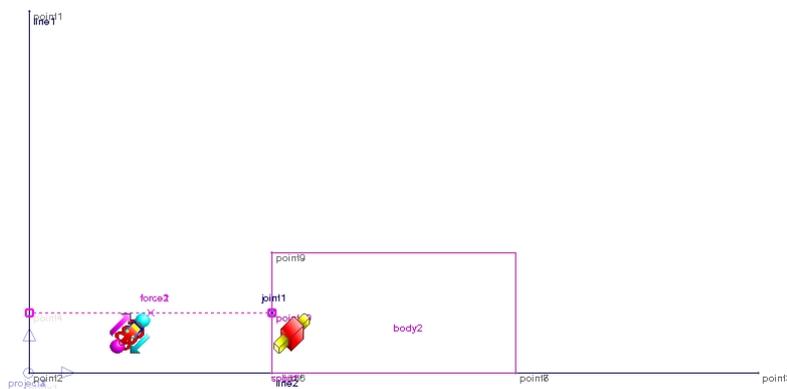
Начальная скорость определяется по формуле

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

Для принятых значений параметров

$$v_0 = 0.163496 \text{ м}.$$

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.

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 (для постоянного шага интегрирования 0.001 [s]).

Задаваемые значения абс. и отн. погрешностей решения краевой задачи [-]	Относительное отличие от теоретического решения v0_delta_rel [-]
0.001	0.000 20
0.000 1	0.000 22
0.000 01	0.000 000 15

Относительное отличие решения задачи в EULER в зависимости от шага численного интегрирования (использовался постоянный шаг интегрирования) представлено в следующей таблице (для значений абсолютной и относительной погрешностей решения краевой задачи 0.00001 [-]).

Шаг интегрирования [s]	Относительное отличие от теоретического решения v0_delta_rel [-]
0.1	0.001 2
0.01	0.000 004 1
0.001	0.000 000 15

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 ) );
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, point9 ) );
solid solid1 = extrude( line2, point5, point9, mass = m );
body body2 = body( color = RGB( 229, 0, 229 ) );
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 );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Прикрепление объектов
body body1 < (point4 );
body body2 < (solid1, point10 );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Команды исследования
command command1 = boundaryProblem( list( v0 ), list( s ), list( l ), 0.00001, 0.00001, 100, comDynamics );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Инерциальное звено;
set ground = body1;
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Единицы измерения;
set units = SI;
```

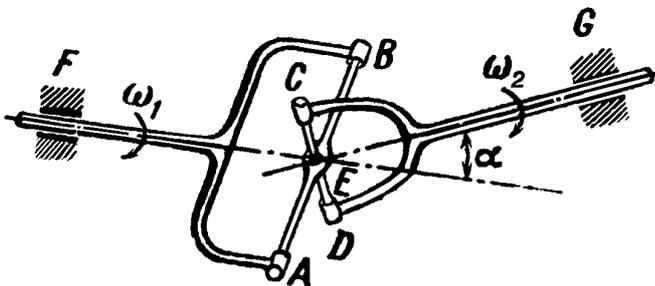
Задача 19

(Задача 24.41. Задачи по теоретической механике. И.В. Мещерский. СПб.: Издательство «Лань», 2001. – 448 с.)

Крестовина $ABCD$ универсального шарнира Кардана-Гука ($AB \perp CD$), употребляемого при передаче вращения между пересекающимися осями, вращается вокруг неподвижной точки E . Найти отношение ω_1/ω_2 для валов, связанных крестовиной в двух случаях:

- 1) когда плоскость вилки ABF горизонтальна, а плоскость вилки CDG вертикальна;
- 2) когда плоскость вилки ABF вертикальна, а плоскость вилки CDG ей перпендикулярна.

Угол между осями валов постоянный: $\alpha = 60^\circ$.



Точный теоретический ответ

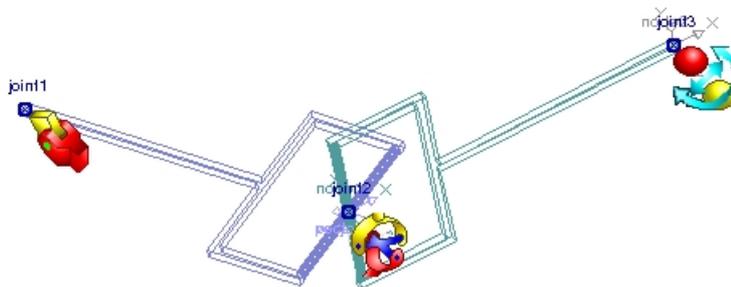
Отношение угловых скоростей валов определяется по формулам:

- 1) $\omega_1/\omega_2 = \cos \alpha$;
- 2) $\omega_1/\omega_2 = 1 / \cos \alpha$.

Для принятых значений параметров

- 1) $\omega_1/\omega_2 = 0.5$;
- 2) $\omega_1/\omega_2 = 2$.

Решение задачи в EULER



Система состоит из трех звеньев:

- Вал 1 (body1). Звено отображается телами solid1 и solid2.
- Вал 2 (body2). Звено отображается телами solid3 и solid4.
- Инерциальное звено (body3). Звено в проекте не отображается.

Вращение первого вала моделируется шарниром joint1 типа *пара вращения*.

Вращение второго вала моделируется шарниром joint3 типа *пользовательский шарнир*, который имеет все три вращательные степени свободы одну поступательную, вдоль оси вала. Поступательная и вращательные степени свободы второго вала позволяют избежать заклинивания механизма. Передача вращения от первого вала ко второму моделируется шарниром joint2 типа *кардановый шарнир*.

Начальная угловая скорость первому валу сообщается с помощью объекта *условие состояния механизма condition1*. Для определения угловых скоростей валов используются

датчики w_1 и w_2 . Отношение угловых скоростей валов вычисляется с помощью датчика w_1_w2 .

В проекте создан объект *условие состояния механизма* condition2. Он изменяет положение механизма, которое соответствует второму условию задачи.

Для решения задачи в первом случае достаточно перейти в режим исследования проекта. При этом condition2 должен находиться в режиме *выключен*.

Для решения задачи во втором случае необходимо переключить состояние объекта condition2 в режим *включен* и перейти в режим исследования проекта.

Результаты моделирования

При моделировании в EULER для принятых значений параметров получено.

В первом случае.

$$w1_w2 = 0.500000 [-].$$

Относительное отличие от теоретического решения

$$wa_delta_rel = 0.000\ 000 [-].$$

Во втором случае.

$$w1_w2 = 2.00000 [-].$$

Относительное отличие от теоретического решения

$$wb_delta_rel = 0.000\ 000\ 000\ 000\ 10 [-].$$

Текст проекта в EULER

```
scalar a = 60 [ deg ];
scalar w0 = 10 [ deg / s ];
point point1 = point( 0.01 [ m ], 0 [ m ], 0.2 [ m ] );
point point2 = point( 0.01 [ m ], 0 [ m ], -0.2 [ m ] );
point point3 = point( 0.01 [ m ], 0 [ m ], 0.22 [ m ] );
point point4 = point( 0.01 [ m ], 0 [ m ], -0.22 [ m ] );
point point5 = point( -0.22 [ m ], 0 [ m ], 0.22 [ m ] );
point point6 = point( -0.22 [ m ], 0 [ m ], -0.22 [ m ] );
point point7 = point( -0.22 [ m ], 0 [ m ], 0.01 [ m ] );
point point8 = point( -0.22 [ m ], 0 [ m ], -0.01 [ m ] );
point point9 = point( -0.8 [ m ], 0 [ m ], 0.01 [ m ] );
point point10 = point( -0.8 [ m ], 0 [ m ], -0.01 [ m ] );
point point11 = point( -0.2 [ m ], 0 [ m ], 0.2 [ m ] );
point point12 = point( -0.2 [ m ], 0 [ m ], -0.2 [ m ] );
line line1 = polyLine( list( point1, point3, point5, point7, point9, point10, point8, point6, point4, point2, point12, point11, point1 ) );
point point13 = point( 0.01 [ m ], 0.01 [ m ], 0.2 [ m ] );
point point14 = point( 0.01 [ m ], -0.01 [ m ], 0.2 [ m ] );
solid solid1 = extrude( line1, point13, point14 );
point point15 = point( 0 [ m ], 0 [ m ], 0.2 [ m ] );
point point16 = point( 0 [ m ], 0 [ m ], -0.2 [ m ] );
solid solid2 = cylinder( point15, point16, 0.01 [ m ] );
color color1 = RGB( 123, 128, 210 );
body body1 = body( color = color1 );
point point17 = point( 0 [ m ], 0 [ m ], 0 [ m ] );
node node1 = node( point17, 0 [ deg ], a, 0 [ deg ] );
point point18 = pointN( node1, -0.01 [ m ], 0.2 [ m ], 0 [ m ] );
point point19 = pointN( node1, -0.01 [ m ], -0.2 [ m ], 0 [ m ] );
point point20 = pointN( node1, -0.01 [ m ], 0.22 [ m ], 0 [ m ] );
point point21 = pointN( node1, -0.01 [ m ], -0.22 [ m ], 0 [ m ] );
point point22 = pointN( node1, 0.22 [ m ], 0.22 [ m ], 0 [ m ] );
point point23 = pointN( node1, 0.22 [ m ], -0.22 [ m ], 0 [ m ] );
point point24 = pointN( node1, 0.22 [ m ], 0.01 [ m ], 0 [ m ] );
point point25 = pointN( node1, 0.22 [ m ], -0.01 [ m ], 0 [ m ] );
point point26 = pointN( node1, 0.8 [ m ], 0.01 [ m ], 0 [ m ] );
point point27 = pointN( node1, 0.8 [ m ], -0.01 [ m ], 0 [ m ] );
point point28 = pointN( node1, 0.2 [ m ], 0.2 [ m ], 0 [ m ] );
point point29 = pointN( node1, 0.2 [ m ], -0.2 [ m ], 0 [ m ] );
line line2 = polyLine( list( point18, point20, point22, point24, point26, point26, point27, point25, point23, point21, point19, point29,
point28, point18 ) );
point point30 = pointN( node1, 0 [ m ], 0.2 [ m ], 0.01 [ m ] );
point point31 = pointN( node1, 0 [ m ], 0.2 [ m ], -0.01 [ m ] );
solid solid3 = extrude( line2, point30, point31 );
point point32 = pointN( node1, 0 [ m ], 0.2 [ m ], 0 [ m ] );
point point33 = pointN( node1, 0 [ m ], -0.2 [ m ], 0 [ m ] );
solid solid4 = cylinder( point32, point33, 0.01 [ m ] );
color color2 = RGB( 78, 156, 156 );
body body2 = body( color = color2 );
body body3 = body( color = RGB( 229, 229, 229 ) );
point point34 = point( -0.8 [ m ], 0 [ m ], 0 [ m ] );
joint joint1 = rotational( body3, body1, point34, projectX );
point point35 = pointN( node1, 0.8 [ m ], 0 [ m ], 0 [ m ] );
vector vector1 = vectorX( node1 );
```

```
vector vector2 = rotate( projectY, projectX, 45 [ deg ] );
joint joint2 = cardan( body1, body2, point17, projectX, vector1, projectZ );
node node2 = nodePN( point35, node1 );
joint joint3 = user( body3, body2, node2, yesX:, noY:, noZ:, yesFiX:, yesFiY:, yesFiZ: );
condition condition1 = rotVelocity( body3, projectX, body1, w0 );
condition condition2 = sensorValue( joint1.gamma, 90 [ deg ], work = off: );
sensor w1 = rotVelocity( body3, projectX, body1 );
sensor w2 = rotVelocity( body3, vector1, body2, fixingVector = unlockVector: );
sensor w1_w2 = w1/w2;
sensor wa_theoretical = cos( a );
sensor wa_delta_rel = abs( (w1_w2-wa_theoretical)/ wa_theoretical );
sensor wb_theoretical = 1/cos( a );
sensor wb_delta_rel = abs( (w1_w2-wb_theoretical)/ wb_theoretical );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Прикрепление объектов
body body1 < (solid1, solid2 );
body body2 < (solid3, solid4 );
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Инерциальное звено;
set ground = body3;
```

```
Λ////////////////////////////////////////////////////////////////
```

```
Λ Единицы измерения;
set units = SI;
```

Задача 20

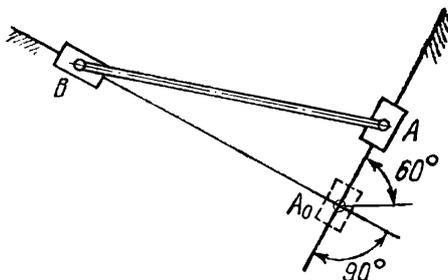
(Задача 9.48. Сборник задач по теоретической механике/ Под ред. К.С. Колесникова. М.: Наука, Главная редакция физико-математической литературы, 1983. – 320 с.)

Ползуны A и B одинаковой массы m , шарнирно соединенные однородным стержнем AB длины l , имеющим также массу m , могут скользить без трения по взаимно перпендикулярным направляющим, расположенным в вертикальной плоскости. В положении A_0 ползуну A сообщается начальная скорость v_0 .

Определить, при каком значении начальной скорости стержень достигнет горизонтального положения.

Для решения задачи использовать следующие значения параметров:

$$m = 1 \text{ кг}, l = 0.4 \text{ м}, v_0 = 1 \text{ м/с}.$$



Точный теоретический ответ

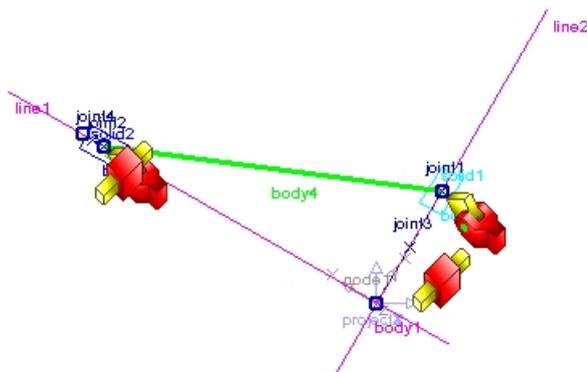
Начальная скорость ползуна определяется по формуле

$$v_0 = \frac{3}{4} \cdot \sqrt{2 \cdot g \cdot l \cdot (\sqrt{3} - 1)}.$$

Для принятых значений параметров

$$v_0 = 1.79768 \text{ м/с}.$$

Решение задачи в EULER



Система состоит из четырех звеньев.

- Инерциальное звено ($body1$). В проекте звено отображается линиями $line1$ и $line2$.
- Два ползуна ($body2$, $body3$). В проекте звенья отображаются параллелепипедами $solid1$ и $solid2$. Каждый из них имеет массу m .
- Стержень ($body4$). Звено изображается цилиндром $solid3$ и имеет массу m .

Ползуны и стержень связаны шарнирами $joint1$ и $joint2$ типа *пара вращения*.

Движение ползун по направляющим моделируется шарнирами $joint3$ и $joint4$ типа *поступательная пара*. В проекте задана гравитация с ускорением свободного падения

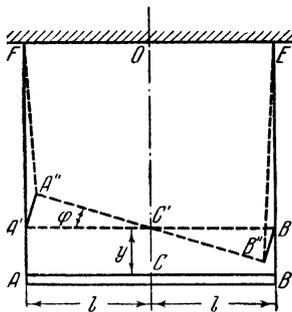
set units = SI;

Задача 21

(Задача 998. Сборник задач по теоретической механике. И.Н. Веселовский. М.: Государственное издательство технико-теоретической литературы, 1955. – 500 с.)

Бифилярный подвес. Однородный стержень длины $2l$ и веса $P=mg$ висит на двух нитях той же длины l так, что может поворачиваться, оставаясь всегда в горизонтальной плоскости. Горизонтальная пара сил с моментом M выводит стержень из начального положения. Определить угол φ отклонения стержня при равновесии.

Для решения задачи использовать следующие значения параметров:
 $l = 0.5 \text{ м}$, $m = 1 \text{ кг}$, $M = 1 \text{ Н м}$.



Точный теоретический ответ

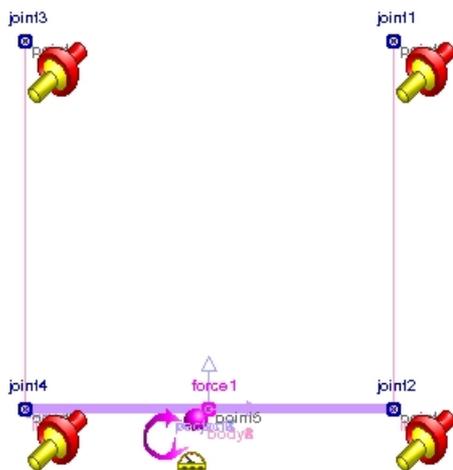
Угол отклонения стержня при равновесии определяется по формуле

$$\varphi = 2 \cdot \arcsin\left(\frac{M}{m \cdot g \cdot l}\right).$$

Для принятых значений параметров

$$\varphi = 0.410626 \text{ рад.}$$

Решение задачи в EULER



Система состоит из четырех звеньев.

- Инерциальное звено (body1). В проекте звено не отображается.
- Стержень (body2). Звено в проекте отображается горизонтальным цилиндром solid1 и имеет массу m .
- Две нити (body3, body4). Звенья отображаются вертикальными линиями line1 и line2. Массы нитей равны нулю.

Для соединения звеньев использованы шарниры joint1, joint2, joint3 и joint4 типа *сферический шарнир*. В проекте задана гравитация с ускорением свободного падения $9.81 \text{ [m/s}^2\text{]}$.

