

Буксировка космического мусора на тросе

Интегрированные математические пакеты.

Лекция №2. Работа со списками, элементы функционального программирования, составление уравнений движения систем тел.

Кафедра теоретической механики. Самарский университет.

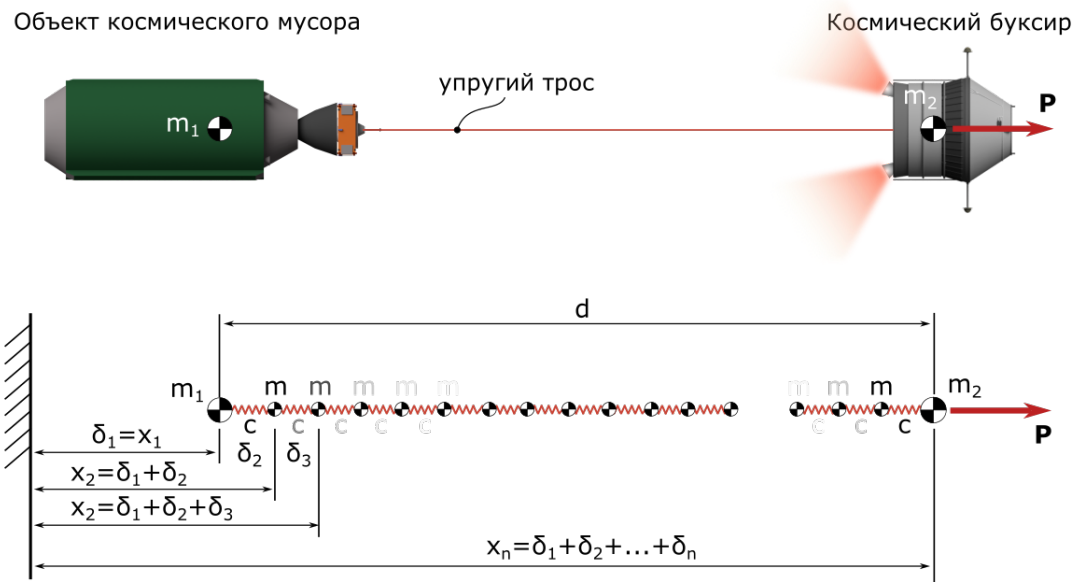
Юдинцев В. В.



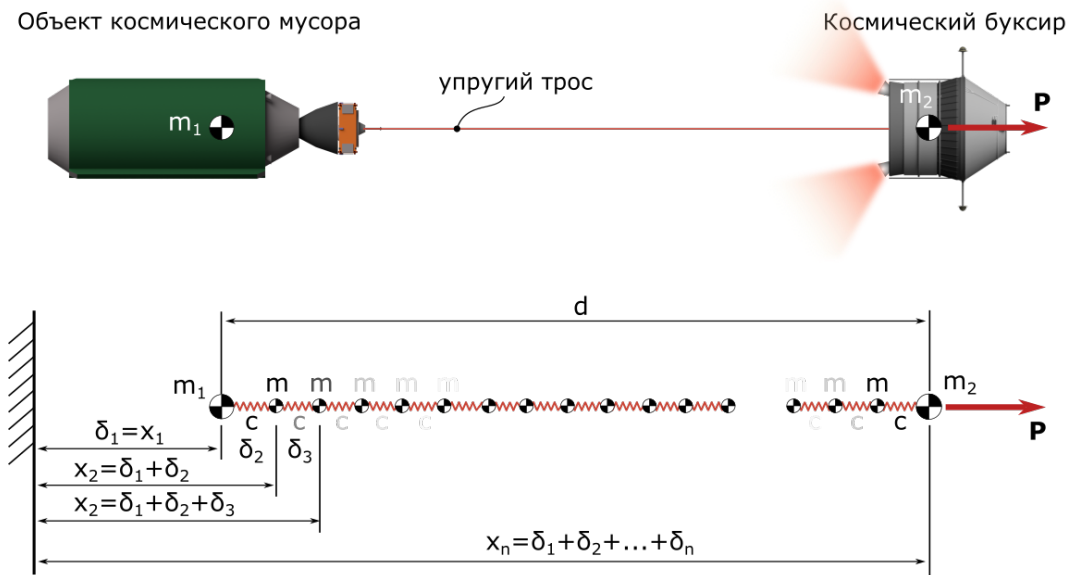
Постановка задачи

Рассматривается процесс тросовой буксировки объекта космического мусора на тросовой связи при помощи космического буксира.

- Движение буксира и объекта космического мусора, которые рассматриваются как материальные точки происходит вдоль прямой, совпадающей с линией действия тяги буксира.
- Движение происходит в безгравитационном поле (притяжение Земли не учитывается).
- Трос рассматривается как система материальных точек, соединенных невесомыми упругими элементами (пружинами).



Задача



- Построить и проинтегрировать уравнения движения системы “буксир - трос - объект” космического мусора под действием постоянной тяги буксира.
- Построить графики изменения скорости буксира и объекта космического мусора.

Уравнения движения

Обобщенные координаты

Количество тел вместе с буксиром и объектом космического мусора.

$In[*]:= n = 3;$

Трос разбивается на $n-2$ узлов. Генерируем список координат - расстояния между точками системы (узлами троса и телами).

Два способа использования функции Map

$In[*]:= q = \delta_{\#}[t] \& /@ \text{Range}[n]$

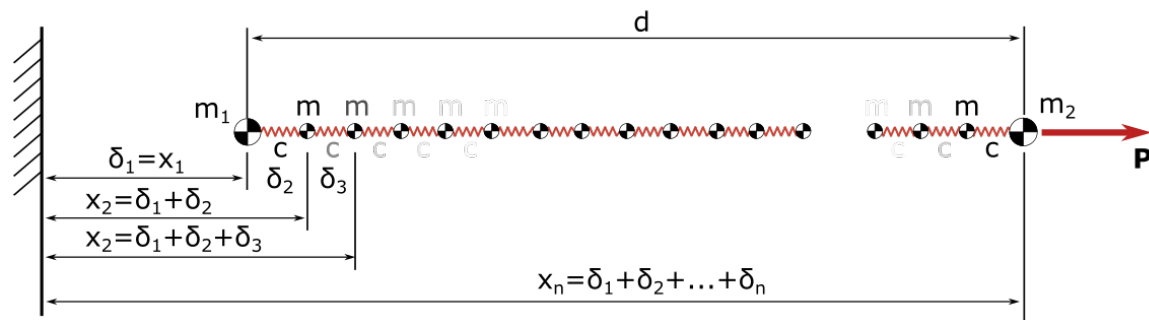
$Out[*]:= \{\delta_1[t], \delta_2[t], \delta_3[t]\}$

$In[*]:= q = \text{Map}[\delta_{\#}[t] \&, \text{Range}[n]]$

$Out[*]:= \{\delta_1[t], \delta_2[t], \delta_3[t]\}$

$In[*]:= dq = D[q, t]$

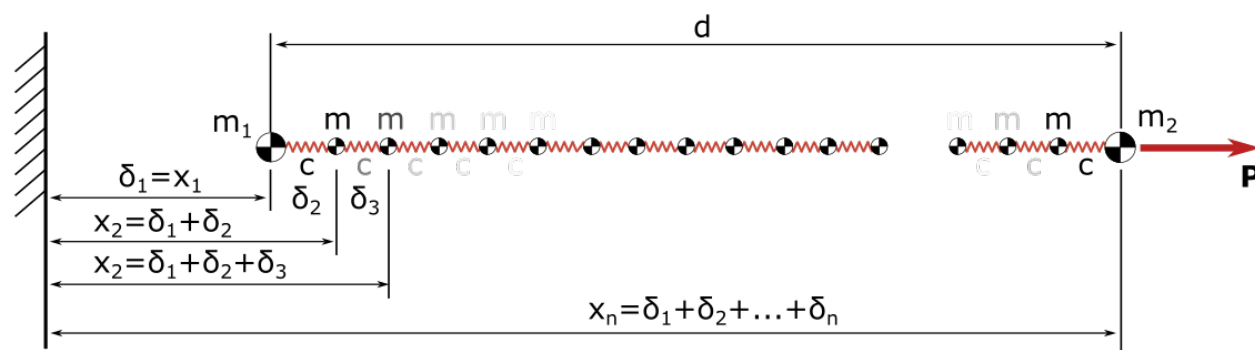
$Out[*]:= \{\delta_1'[t], \delta_2'[t], \delta_3'[t]\}$



Уравнения движения

Абсолютные координаты

Абсолютные координаты - расстояние до тела (узла троса или буксира, космического мусора) от некоторой неподвижной системы координат.



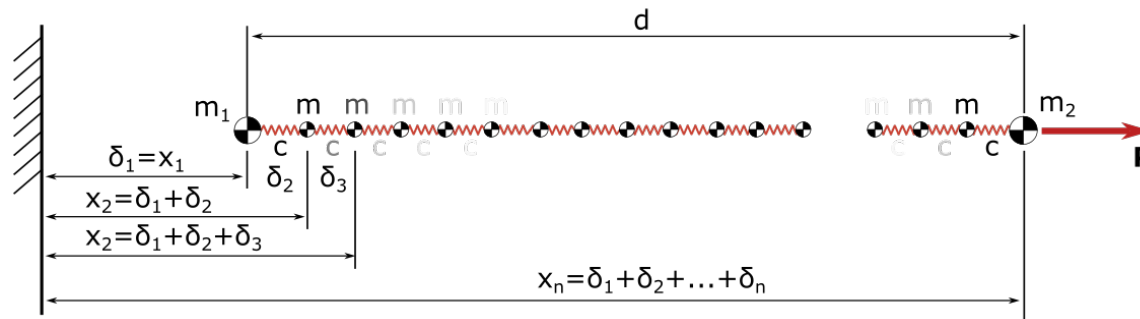
Функция **Accumulate**.

$In[*]:= \mathbf{x = Accumulate[q]}$

$Out[*]:= \{\delta_1[t], \delta_1[t] + \delta_2[t], \delta_1[t] + \delta_2[t] + \delta_3[t]\}$

Массы тел системы

Формирование списка масс



m1 - m - m - m - m - m - m - m2

- m_1 - масса объекта космического мусора,
- m_2 - масса буксира,
- m - масса узла троса .

Используются функция **Join** и **ConstantArray**

```
In[ ]:= masses = Join[ {m1}, ConstantArray[m, n - 2], {m2} ]
```

```
Out[ ]:= {m1, m, m2}
```

(трос разбивается на $n - 2$ узлов)

Левая часть уравнений движения $\mathbf{ma} = \mathbf{F}$

Левая часть уравнений движения $\mathbf{ma} = \mathbf{F}$

In[]:=* eqLeft = D[x, {t, 2}] * masses

Out[]:=* {m1 $\delta_1''[t]$, m ($\delta_1''[t] + \delta_2''[t]$), m2 ($\delta_1''[t] + \delta_2''[t] + \delta_3''[t]$)}

Силы растяжения пружин

Деформация пружин между телами (узлами троса).

Для построения списка этих расстояний используем функцию **Map** с анонимной функцией: $\# - l_0 \&$

■ l_0 – свободная длина пружины ("нерастянутая" длина пружины)

$In[*]:= \# - l_0 \& /@ q$

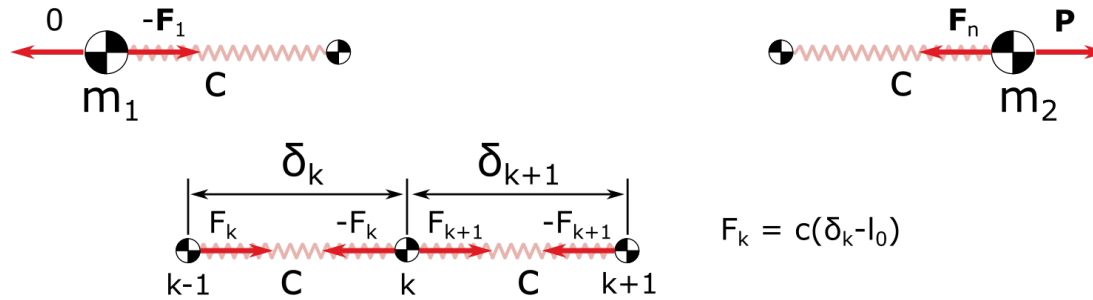
$Out[*]:= \{-l_0 + \delta_1[t], -l_0 + \delta_2[t], -l_0 + \delta_3[t]\}$

Второй вариант вызова функции **Map**

$In[*]:= \text{Map}[\# - l_0 \&, q]$

$Out[*]:= \{-l_0 + \delta_1[t], -l_0 + \delta_2[t], -l_0 + \delta_3[t]\}$

Силы, действующие на узлы



- На внутренние узлы троса действует всегда две силы со стороны соседних узлов.
- На объект космического мусора действует только одна сила натяжения со стороны ближайшего узла.
- На буксир действует только сила натяжения со стороны ближайшего узла и сила тяги двигателя.

$In[*]:=$ **Fsprings**

$Out[*]:=$ $\{0, c(-l_0 + \delta_2[t]), c(-l_0 + \delta_3[t]), P\}$

$In[*]:=$ **Partition** $[\{a, b, c, d\}, 2, 1]$

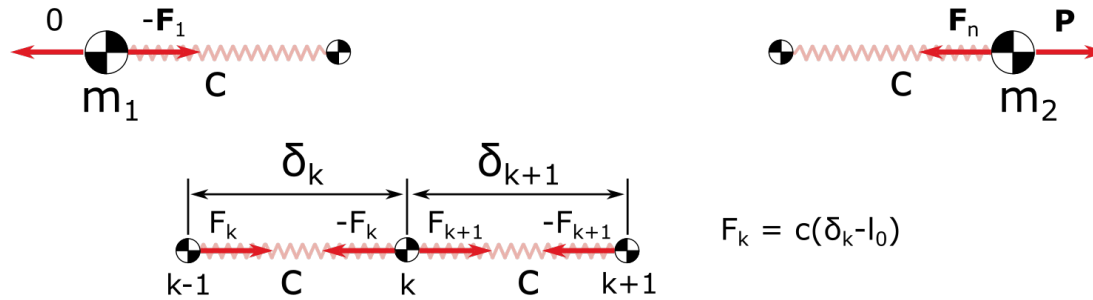
$Out[*]:=$ $\{\{a, b\}, \{b, c\}, \{c, d\}\}$

Список двух сил, действующих на узлы троса и на объект космического мусора с буксиром

$In[*]:=$ **Partition** $[Fsprings, 2, 1]$

$Out[*]:=$ $\{\{0, c(-l_0 + \delta_2[t])\}, \{c(-l_0 + \delta_2[t]), c(-l_0 + \delta_3[t])\}, \{c(-l_0 + \delta_3[t]), P\}\}$

Силы, действующие на узлы



```
In[*]:= Partition[Fsprings, 2, 1]
```

```
Out[*]:= {{0, c (-l0 + delta2[t])}, {c (-l0 + delta2[t]), c (-l0 + delta3[t])}, {c (-l0 + delta3[t]), P}}
```

```
In[*]:= F = -#[[1]] + #[[2]] & /@ Partition[Fsprings, 2, 1]
```

```
Out[*]:= {c (-l0 + delta2[t]), -c (-l0 + delta2[t]) + c (-l0 + delta3[t]), P - c (-l0 + delta3[t])}
```

Получен список сумм сил, действующих на каждое тело системы (узел троса или концевое тело)

Формирование уравнений $m_i a_i = F_i$

```
In[*]:= eqLeft
F
```

```
Out[*]:= {m1 δ1''[t], m (δ1''[t] + δ2''[t]), m2 (δ1''[t] + δ2''[t] + δ3''[t]) }
```

```
Out[*]:= {c (-l0 + δ2[t]), -c (-l0 + δ2[t]) + c (-l0 + δ3[t]), P - c (-l0 + δ3[t]) }
```

Поэлементное “приравнивание” пар из двух списков левой и правой части

```
In[*]:= eq = MapThread[#1 == #2 &, {eqLeft, F}]
```

```
Out[*]:= {m1 δ1''[t] == c (-l0 + δ2[t]), m (δ1''[t] + δ2''[t]) == -c (-l0 + δ2[t]) + c (-l0 + δ3[t]), m2 (δ1''[t] + δ2''[t] + δ3''[t]) == P - c (-l0 + δ3[t]) }
```

Уравнения движения

Весь код формирования уравнений

```
In[ ]:= n = 3;
q =  $\delta_{\#}$ [t] & /@ Range[n]; dq = D[q, t]; x = Accumulate[q];
eqLeft = D[x, {t, 2}] * Join[{m1}, ConstantArray[m, n - 2], {m2}];
F = -#[[1]] + #[[2]] & /@ Partition[Join[{0}, ( $-l_0$  & /@ q[[2 ;;]]) c, {P}], 2, 1];
eq = MapThread[#1 == #2 &, {eqLeft, F}]

Out[ ]:= {m1  $\delta_1''$ [t] == c (- $l_0$  +  $\delta_2$ [t]), m ( $\delta_1''$ [t] +  $\delta_2''$ [t]) == -c (- $l_0$  +  $\delta_2$ [t]) + c (- $l_0$  +  $\delta_3$ [t]), m2 ( $\delta_1''$ [t] +  $\delta_2''$ [t] +  $\delta_3''$ [t]) == P - c (- $l_0$  +  $\delta_3$ [t]) }
```

Менее громоздкий **вид** уравнений

```
In[ ]:= eq /. {f_[t] -> f} // TableForm

Out[ ]//TableForm=
m1  $\delta_1''$  == c (- $l_0$  +  $\delta_2$ )
m ( $\delta_1''$  +  $\delta_2''$ ) == -c (- $l_0$  +  $\delta_2$ ) + c (- $l_0$  +  $\delta_3$ )
m2 ( $\delta_1''$  +  $\delta_2''$  +  $\delta_3''$ ) == P - c (- $l_0$  +  $\delta_3$ )
```

Параметры системы

Используются замены

$ln[*]:=$ eq

$$Out[*]:= \{m1 \delta_1''[t] == c (-l_0 + \delta_2[t]), m (\delta_1''[t] + \delta_2''[t]) == -c (-l_0 + \delta_2[t]) + c (-l_0 + \delta_3[t]), m2 (\delta_1''[t] + \delta_2''[t] + \delta_3''[t]) == P - c (-l_0 + \delta_3[t])\}$$

$ln[*]:=$ parameters = {m1 → 4000.0, m2 → 1500, YM → 100 × 10⁹, density → 1100.0, tetherDiameter → 0.003,

$$tetherLength \rightarrow 1000.0, tetherArea \rightarrow \frac{\pi tetherDiameter^2}{4}, tetherMass \rightarrow tetherArea * tetherLength * density,$$

$$m \rightarrow tetherMass / (n - 2), l_0 \rightarrow \frac{tetherLength}{n - 1}, c \rightarrow \frac{tetherArea * YM}{l_0}, P \rightarrow 1000.0\};$$

Обычная замена производится один раз

$ln[*]:=$ eq /. parameters

$$Out[*]:= \left\{ \begin{aligned} 4000. \delta_1''[t] &= \frac{tetherArea YM \left(-\frac{tetherLength}{2} + \delta_2[t] \right)}{l_0}, \\ \\ tetherMass (\delta_1''[t] + \delta_2''[t]) &= -\frac{tetherArea YM \left(-\frac{tetherLength}{2} + \delta_2[t] \right)}{l_0} + \frac{tetherArea YM \left(-\frac{tetherLength}{2} + \delta_3[t] \right)}{l_0}, \\ \\ 1500 (\delta_1''[t] + \delta_2''[t] + \delta_3''[t]) &= 1000. - \frac{tetherArea YM \left(-\frac{tetherLength}{2} + \delta_3[t] \right)}{l_0} \end{aligned} \right\}$$

Повторяющаяся замена //. производится до тех пор, пока итоговое выражение не перестает меняться в результате замен.

$ln[*]:=$ eq //. parameters

$$Out[*]:= \{4000. \delta_1''[t] == 1413.72 (-500. + \delta_2[t]), 7.77544 (\delta_1''[t] + \delta_2''[t]) == -1413.72 (-500. + \delta_2[t]) + 1413.72 (-500. + \delta_3[t]), \\ 1500 (\delta_1''[t] + \delta_2''[t] + \delta_3''[t]) == 1000. - 1413.72 (-500. + \delta_3[t])\}$$

Начальные условия

Пусть в начальный момент длины всех участков троса равны его свободной длине $\delta_i(0) = l_0$

$In[*]:=$ **q**

$Out[*]:=$ $\{\delta_1[t], \delta_2[t], \delta_3[t]\}$

$In[*]:=$ **q /. t → 0**

$Out[*]:=$ $\{\delta_1[0], \delta_2[0], \delta_3[0]\}$

$In[*]:=$ **q0 = MapThread[Equal, {q /. t → 0, ConstantArray[l0, n]}]**

$Out[*]:=$ $\{\delta_1[0] == l_0, \delta_2[0] == l_0, \delta_3[0] == l_0\}$

$In[*]:=$ **D[q, t]**

$Out[*]:=$ $\{\delta_1'[t], \delta_2'[t], \delta_3'[t]\}$

$In[*]:=$ **dq0 = MapThread[Equal, {D[q, t] /. t → 0, ConstantArray[0, n]}]**

$Out[*]:=$ $\{\delta_1'[0] == 0, \delta_2'[0] == 0, \delta_3'[0] == 0\}$

Собираем

Уравнения

```
In[ ]:= n = 5;
q =  $\delta_{\#}$ [t] & /@ Range[n]; dq = D[q, t]; x = Accumulate[q];
eqLeft = D[x, {t, 2}] * Join[{m1}, ConstantArray[m, n - 2], {m2}];
F = -#[[1]] + #[[2]] & /@ Partition[Join[{0}, ( $\# - l_0$  & /@ q[[2 ;;]]) c, {P}], 2, 1];
eq = MapThread[#1 == #2 &, {eqLeft, F}];
```

Параметры системы

```
In[ ]:= parameters = {m1 → 4000.0, m2 → 1500, YM →  $100 \times 10^9$ , density → 1100.0, tetherDiameter → 0.003,
  tetherLength → 1000.0, tetherArea →  $\frac{\pi \text{tetherDiameter}^2}{4}$ , tetherMass → tetherArea * tetherLength * density,
  m → tetherMass / (n - 2),  $l_0 \rightarrow \frac{\text{tetherLength}}{n - 1}$ ,  $c \rightarrow \frac{\text{tetherArea} * \text{YM}}{l_0}$ , P → 1000.0};
```

Начальные условия

```
In[ ]:= q0 = MapThread[Equal, {q /. t → 0, ConstantArray[l0, n]}];
dq0 = MapThread[Equal, {D[q, t] /. t → 0, ConstantArray[0, n]}];
```

Интегрирование

```
In[ ]:= tk = 20;
sol = NDSolve[{eq, q0, dq0} /. parameters, {q, dq} // Flatten, {t, 0, tk}, Method → {"EquationSimplification" → "Residual"}];
```


Интегрирование

Численное интегрирование уравнений движения

```
In[ ]:= tk = 20;
sol = NDSolve[{eq, q0, dq0} /. parameters, {q, dq} // Flatten, {t, 0, tk}, Method -> {"EquationSimplification" -> "Residual"}]
```

Out[]:= $\left\{ \left\{ \delta_1[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \delta_2[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \right.$

$\delta_3[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \delta_4[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t],$

$\delta_5[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \delta_1'[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t],$

$\delta_2'[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \delta_3'[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t],$

$\delta_4'[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t], \delta_5'[t] \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{0., 20.\} \\ \text{Output: scalar} \end{array} \right] [t] \left. \right\} \}$

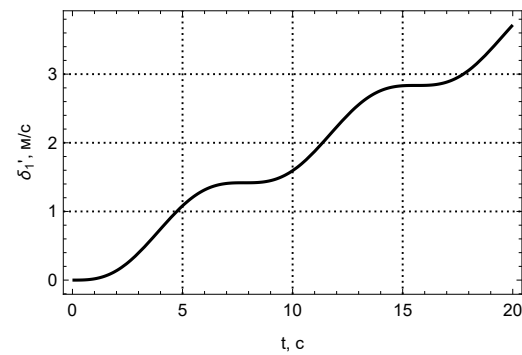
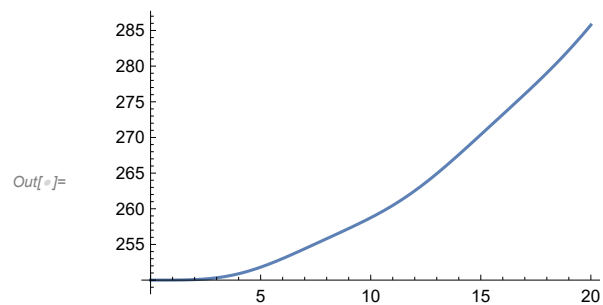
Графики

Положение и скорость объекта космического мусора

```

In[ ]:= GraphicsGrid[
  {{
    Plot[ $\delta_1[t]$  /. sol, {t, 0, tk}],
    Plot[ $\delta_1'[t]$  /. sol, {t, 0, tk}, PlotTheme -> {"Monochrome", "FrameGrid"}, FrameLabel -> {"t, c", " $\delta_1'$ , м/с"}]
  }}
]

```



Графики

```
In[ ]:= SetOptions[Plot, PlotTheme -> {"Monochrome", "FrameGrid"}];
```

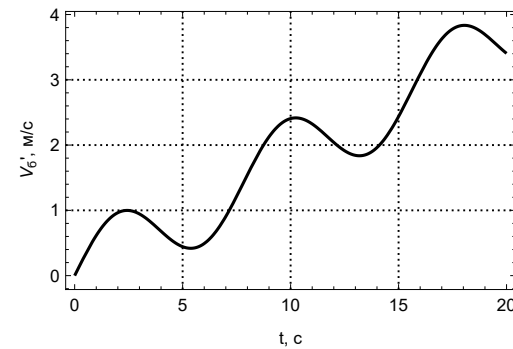
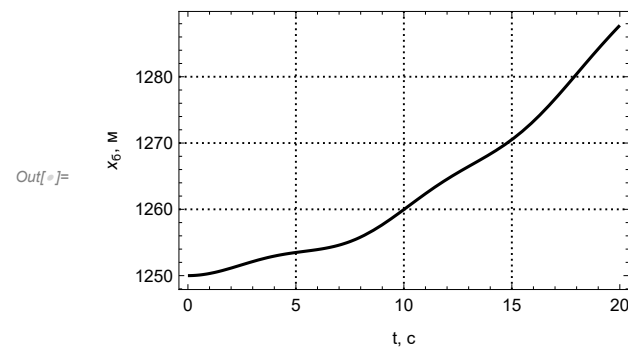
Положение и скорость (абсолютные) буксира

```
In[ ]:= x[[-1]]
```

```
Out[ ]:=  $\delta_1[t] + \delta_2[t] + \delta_3[t] + \delta_4[t] + \delta_5[t]$ 
```

```
In[ ]:= vx = D[x, t];
```

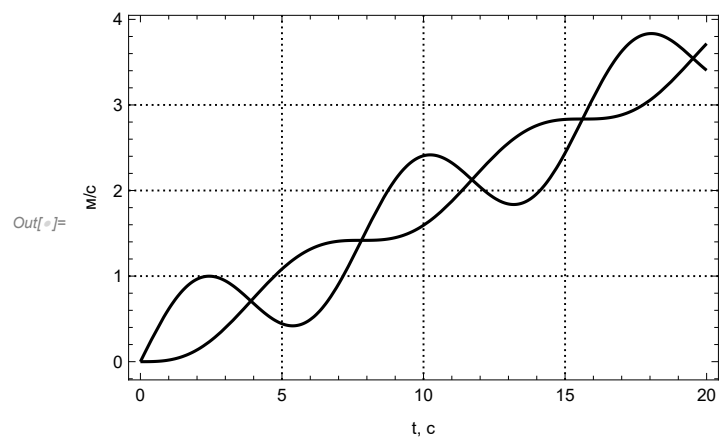
```
In[ ]:= GraphicsGrid[
  {{
    Plot[x[[-1]] /. sol, {t, 0, tk}, FrameLabel -> {"t, c", "x6, m"}],
    Plot[vx[[-1]] /. sol, {t, 0, tk}, FrameLabel -> {"t, c", "V6', m/c"}]
  }}
]
```



Несколько графиков на одном рисунке

Если на одном рисунке выводится сразу несколько графиков, то в некоторых случаях Mathematica автоматически не меняет стили кривых.

```
In[ ]:= Plot[vx[1, -1]] /. sol, {t, 0, tk}, FrameLabel -> {"t, c", "m/c"}
```



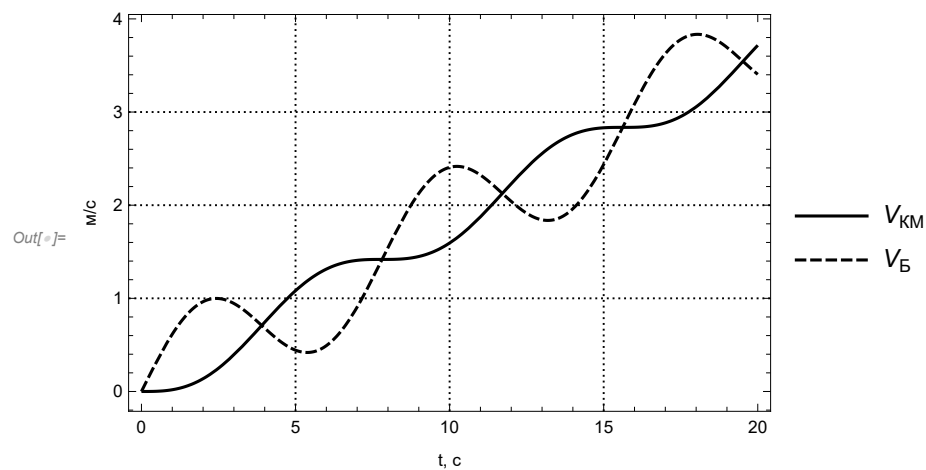
Несколько графиков на одном рисунке

Применяем функцию **Evaluate**

```

In[ ]:= Plot[vx[[{1, -1}]] /. sol // Evaluate, {t, 0, tk}, FrameLabel -> {"t, c", "M/c"},
PlotLegends -> {"VKM", "VБ"}]

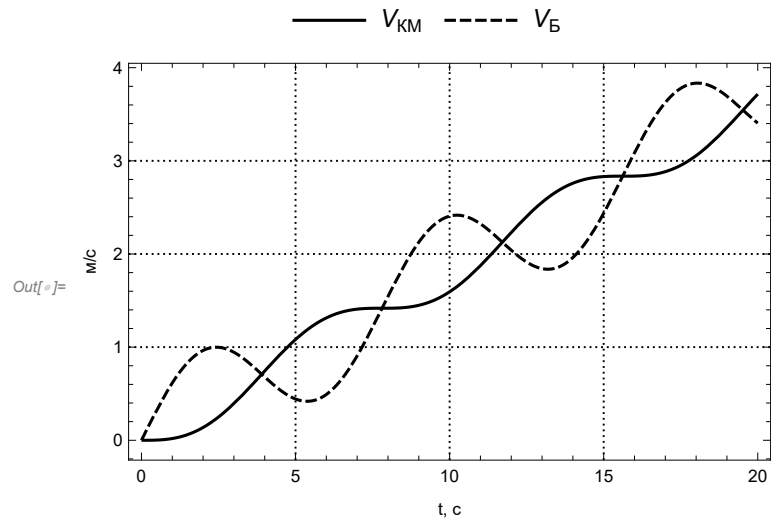
```



Legend

Положение "легенды"

```
In[ ]:= Plot[vx[[{1, -1}]] /. sol // Evaluate, {t, 0, tk}, FrameLabel -> {"t, c", "M/c"},
  PlotLegends -> Placed[{"VKM", "VB"}, Above]]
```



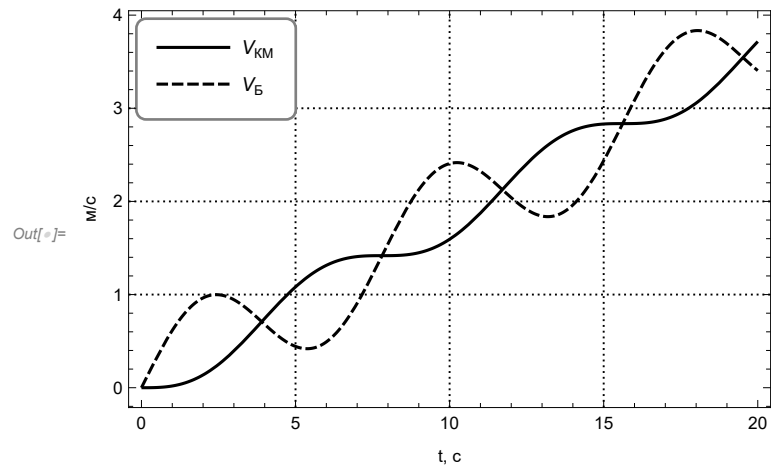
Legend

Стиль “легенды”

```

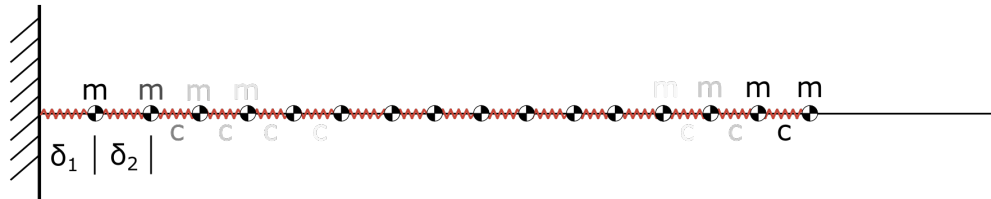
In[ ]:= Plot[vx[[1, -1]] /. sol // Evaluate, {t, 0, tk}, FrameLabel -> {"t, c", "M/c"},
  PlotLegends -> Placed[
    LineLegend[
      {"Vкм", "Vб"}, LegendFunction -> (Framed[#, Background -> White, RoundingRadius -> 5, FrameStyle -> Gray] &)
    ], {Left, Top}
  ]
]

```



Лабораторная работа №2

1. Построить графики изменения положения и скорости центра масс системы
2. На основе примера разработать программу моделирования движения системы материальных точек ($n = 5$), связанных пружинами, по горизонтальной шероховатой плоскости.



- Первое тело соединено пружиной с неподвижной стенкой, жесткости всех пружин равны 200 Н/м. Массы всех тел равны 5 кг, количество тел равно 5. В начальный момент времени расстояние между смежными материальными точками равно 1,5 м. Свободная длина пружин, соединяющих точки равна 1 м.
 - На тела системы действует сила трения
3. Построить графики изменения абсолютных скоростей всех точек системы (на одном графике).
 4. Построить графики изменения кинетической энергии системы.
 5. Построить графики изменения потенциальной энергии системы .
 6. Построить графики изменения полной механической энергии системы.