

О СЛОЖНОМ ДВИЖЕНИИ ТОЧКИ С УЧЕТОМ РЫВКА

В механике под сложным движением материальной точки [1] понимается изменение положения этой точки от независимой переменной t , обычно называемой временем. При этом вводятся две системы отсчета: неподвижная (НС), снабженная часами, и подвижная (ПС). Каждой точке неподвижной системы можно сопоставить некий объект, реагирующий на трансляционное и спинарное движения. Этот объект имеет свой подвижный репер и называется подвижной системой отсчета. Математическое описание положения некоторой точки M (изначально заданной в координатах подвижной системы в момент t) в координатах неподвижной системы опреляется соотношениями (см. Рис.1):

$$\vec{R}(t) = \vec{r}(t) + \vec{\rho}(t), \quad \vec{r}(t) = x_n(t)\vec{e}_n, \quad \vec{\rho}(t) = x'_n(t)\vec{e}'_n(t), \quad (1)$$

где: $\vec{e}_n, \vec{e}'_n(t)$ – ортогональные базисы в (НС) и (ПС) системах отсчета соответственно; используется суммирование по повторяющемуся индексу и подразумевается геометрическое сложение векторов.

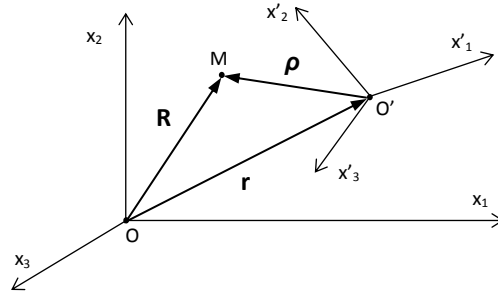


Рис 1. Сложное движение точки M

Для наблюдателя, связанного с (ПС), орты $\vec{e}'_n(t)$ не меняют свое направление, а для наблюдателя, связанного с (НС), орты $\vec{e}'_n(t)$, определяющие (ПС), уже меняют свое направление, поэтому они являются переменными векторами постоянного модуля.

Последовательное дифференцирование вектора $\vec{R}(t)$ в (НС) даёт абсолютные векторы скорости $\dot{\vec{R}}$, ускорения $\ddot{\vec{R}}$ и рывка $\dddot{\vec{R}}$ точки M :

$$\dot{\vec{R}} = \dot{\vec{r}} + \dot{x}'_n \vec{e}'_n + x'_n \dot{\vec{e}}'_n,$$

$$\ddot{\vec{R}} = \ddot{\vec{r}} + (\ddot{x}'_n \vec{e}'_n + x'_n \ddot{\vec{e}}'_n) + 2\dot{x}'_n \dot{\vec{e}}'_n,$$

$$\dddot{\vec{R}} = \dddot{\vec{r}} + (\dddot{x}'_n \vec{e}'_n + x'_n \dddot{\vec{e}}'_n) + 3(\ddot{x}'_n \dot{\vec{e}}'_n + \dot{x}'_n \ddot{\vec{e}}'_n)$$

В полученных соотношениях преобразуем производные базисных векторов $\dot{\vec{e}}'_n$, воспользовавшись формулами Эйлера ($\dot{\vec{e}} = \vec{\omega}$, $\ddot{\vec{e}} = \dot{\vec{\omega}}$):

$$\begin{aligned}\dot{\vec{e}}'_n &= \vec{\omega} \times \vec{e}'_n, \\ \ddot{\vec{e}}'_n &= \vec{\varepsilon} \times \vec{e}'_n + \vec{\omega} \times (\vec{\omega} \times \vec{e}'_n), \\ \ddot{\vec{e}}'_n &= \vec{\gamma} \times \vec{e}'_n + 2\vec{\varepsilon} \times (\vec{\omega} \times \vec{e}'_n) + \vec{\omega} \times (\vec{\varepsilon} \times \vec{e}'_n + \vec{\omega} \times (\vec{\omega} \times \vec{e}'_n))\end{aligned}$$

В результате получим

$$\begin{aligned}\dot{\vec{R}} &= \dot{\vec{r}} + \dot{\vec{\rho}} + \vec{\omega} \times \vec{\rho}, \\ \ddot{\vec{R}} &= \ddot{\vec{r}} + \ddot{\vec{\rho}} + 2\vec{\omega} \times \dot{\vec{\rho}} + \vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}), \\ \ddot{\vec{R}} &= \ddot{\vec{r}} + \ddot{\vec{\rho}} + 3(\vec{\omega} \times \dot{\vec{\rho}} + \vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho})) + \\ &\quad + \vec{\gamma} \times \vec{\rho} + 2\vec{\varepsilon} \times (\vec{\omega} \times \vec{\rho}) + \vec{\omega} \times (\vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}))\end{aligned}$$

Комбинируя члены с учётом правил дифференцирования в системах (НС) и (ПС) можно записать

$$\begin{aligned}\vec{v}_o &= \dot{\vec{\rho}}, \quad \vec{v}_p = \dot{\vec{r}}(t) + \vec{\omega}(t) \times \vec{\rho}(t); \\ \vec{w}_o &= \ddot{\vec{\rho}}, \quad \vec{w}_p = \ddot{\vec{r}} + \vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}), \quad \vec{w}_k = 2\vec{\omega} \times \dot{\vec{\rho}}; \\ \vec{u}_o &= \ddot{\vec{\rho}}, \quad \vec{u}_p = \ddot{\vec{r}} + \vec{\gamma} \times \vec{\rho} + 2\vec{\varepsilon} \times (\vec{\omega} \times \vec{\rho}) + \vec{\omega} \times (\vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho})), \\ \vec{u}_k &= 3(\vec{\omega} \times \dot{\vec{\rho}} + \vec{\varepsilon} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}));\end{aligned}\tag{2}$$

где: $\vec{v}_o, \vec{v}_p$ – относительная и переносная скорости точки M ;

$\vec{w}_o, \vec{w}_p, \vec{w}_k$ – относительное, переносное и Кориолисово ускорения точки M ;

$\vec{u}_o, \vec{u}_p, \vec{u}_k$ – относительный, переносный и Кориолисов¹⁾ рывки точки M .

Тогда абсолютная скорость, ускорение и рывок запишутся в виде векторных сумм:

$$\vec{V} = \vec{v}_o + \vec{v}_p, \quad \vec{W} = \vec{w}_o + \vec{w}_p + \vec{w}_k, \quad \vec{U} = \vec{u}_o + \vec{u}_p + \vec{u}_k\tag{3}$$

В качестве примера, рассмотрим вращающийся плоский диск единичного радиуса, лежащий на плоскости, и точка M движется по ободу диска (см. рис.2). В этом случае $\vec{\rho} = \cos(t)\vec{i} + \sin(t)\vec{j}$, $\vec{\omega} = t^4\vec{k}$ и $\vec{i} = \vec{e}_1$, $\vec{j} = \vec{e}_2$, $\vec{k} = \vec{e}_3$. Плоскость на которой лежит диск является неподвижной системой отсчета, а вращающей диск – подвижной.

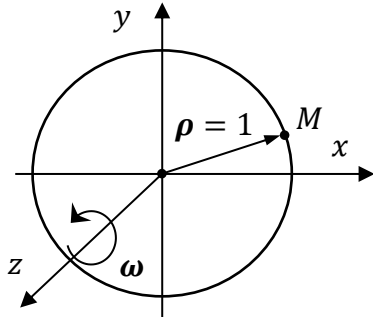


Рис 2. Вращение вокруг оси Z

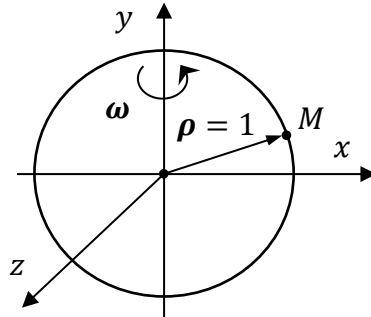


Рис 3. Вращение вокруг оси Y

¹⁾ Кориолисов рывок – это условное название дополнительного рывка.

Результаты ручного расчёта такой системы (рис.2) по уравнениям (2) и (3) при вращении вокруг оси z и $t = 0, \pi/4, \pi/2$ приведены в следующей таблице.

	t		
	0	l	s
$\vec{\rho}$	$\vec{l}$	$g(\vec{l} + \vec{j})$	$\vec{j}$
$\dot{\vec{\rho}}$	$\vec{j}$	$g(\vec{j} + \vec{l})$	$-\vec{l}$
$\ddot{\vec{\rho}}$	$-\vec{l}$	$-g(\vec{l} + \vec{j})$	$-\vec{j}$
$\ddot{\vec{\rho}}$	$-\vec{j}$	$g(\vec{l} - \vec{j})$	$\vec{l}$
$\vec{\omega}$	0	$l^4 \vec{k}$	$s^4 \vec{k}$
$\dot{\vec{\omega}}$	0	$4l^3 \vec{k}$	$4s^3 \vec{k}$
$\ddot{\vec{\omega}}$	0	$12l^2 \vec{k}$	$12s^2 \vec{k}$
$\ddot{\vec{\omega}}$	0	$24l \vec{k}$	$24s \vec{k}$
$\vec{v}_o$	$\vec{j}$	$g(\vec{j} - \vec{l})$	$-\vec{l}$
$\vec{v}_p$	0	$gl^4(\vec{j} - \vec{l})$	$-s^4 \vec{l}$
$\vec{V}$	$\vec{v}_o + \vec{v}_p$	$\vec{v}_o + \vec{v}_p$	$\vec{v}_o + \vec{v}_p$
$\vec{w}_o$	$-\vec{l}$	$-g(\vec{l} + \vec{j})$	$-\vec{j}$
$\vec{w}_p$	0	$(\alpha - \beta)\vec{j} - (\alpha + \beta)\vec{l}$	$-s^3(4\vec{l} + s^5\vec{j})$
$\vec{w}_k$	0	$-2gl^4(\vec{l} + \vec{j})$	$-2s^4\vec{j}$
$\vec{W}$	$\vec{w}_o + \vec{w}_p + \vec{w}_k$	$\vec{w}_o + \vec{w}_p + \vec{w}_k$	$\vec{w}_o + \vec{w}_p + \vec{w}_k$
$\vec{u}_o$	$-\vec{j}$	$g(\vec{l} - \vec{j})$	$\vec{l}$
$\vec{u}_p$	0	$g(\lambda\vec{l} - \mu\vec{j})$	$(s^{12} - s^2)\vec{l} - 12s^7\vec{j}$
$\vec{u}_k$	0	$6gl^3((l-2)\vec{l} - (l+2)\vec{j})$	$3((s^4 + s^8)\vec{l} - 4s^3\vec{j})$
$\vec{U}$	$\vec{u}_o + \vec{u}_p + \vec{u}_k$	$\vec{u}_o + \vec{u}_p + \vec{u}_k$	$\vec{u}_o + \vec{u}_p + \vec{u}_k$

Таблица 1. Результаты ручного расчета

$$(l = \pi/4, s = \pi/2 = 2l, g = 1/\sqrt{2}, \alpha = 4gl^3, \beta = gl^8, \lambda = l^{12} - 12(l^2 + l^7), \mu = l^{12} - 12(l^2 - l^7))$$

Ручной расчёт, даже такого сравнительно простого примера сложного движения с учетом рывка, требует заметных усилий и повышенного внимания. А что говорит, когда функции, описывающие движение в (НС) и (ПС) системах трудно дифференцируемы. Однако, если воспользоваться обобщенными векторами и автоматическим дифференцированием [2], то процедура расчёта становится тривиальной задачей. Для этих целей можно использовать статическую библиотеку [Vec3Lib.dll](#), которая содержит основные операции векторной алгебры: сложение $+$, вычитание $-$, скалярное $*$ и векторное $\wedge$ умножение векторов. Дифференцирование же компонент вектора осуществляется статической библиотекой [Sdn3Lib.dll](#), которая включает переопределение операций $+$, $-$, $*$, $\div$ и основных базовых функций $\text{pow}(\dots)$, $\text{powX}(\dots)$, $\text{exp}(\dots)$, $\text{ln}(\dots)$, $\text{log}(\dots)$, $\text{sqrt}(\dots)$, $\text{sin}(\dots)$, $\text{cos}(\dots)$, $\text{sh}(\dots)$, $\text{ch}(\dots)$, возвращающих гипер-дуальные числа 3-го порядка.

В Приложении 1 приведен код на языке C#, реализующий указанный подход. Функция $\Omega(t)$ описывает вращение (ПС), функция же $R(t)$ – движение точки по окружности единичного радиуса. Обе эти функции возвращают компоненты вектора $(x(t), y(t), z(t))$ и его производных до третьего порядка включительно (тип данных `Vec3Lib.sdn3V3`). Функция $VWU(t)$ реализует формулы (2) и (3) с использованием функций $\Omega(t)$ и $R(t)$. Эта функция возвращает список vwu , структура которого дана на следующей схеме.

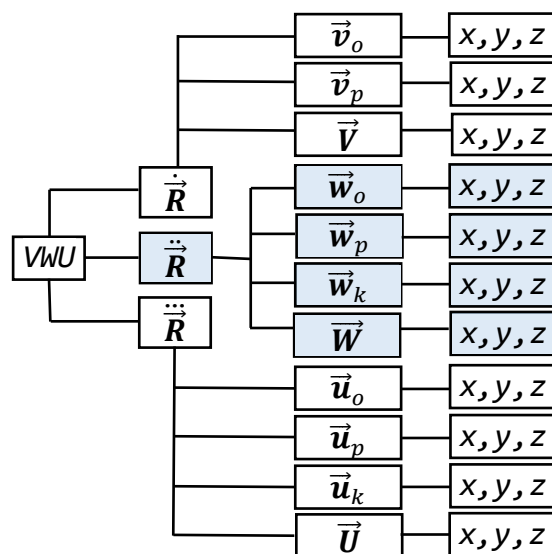


Схема 1. Структура списка, возвращаемого функцией $VWU(t)$

Результаты машинного расчета сложного движения по схеме, приведенной на рис. 3, при вращении вокруг оси y и $t = \pi/4$ представлены в следующей таблице.

	x	y	z
$\vec{v}_o$	-0.70710678118654746	0.70710678118654757	0
$\vec{v}_p$	0	0	-0.26905714382562829
$\vec{V}$	-0.70710678118654746	0.70710678118654757	-0.26905714382562829
$\vec{w}_o$	-0.70710678118654746	-0.70710678118654746	0
$\vec{w}_p$	-0.10237738990726292	0	-1.3702967812491447
$\vec{w}_k$	0	0	0.53811428765125646
$\vec{W}$	-0.80948417109381054	-0.70710678118654746	-0.83218249359788821
$\vec{u}_o$	0.70710678118654757	-0.70710678118654746	0
$\vec{u}_p$	-1.5642112958003724	0	-5.1951931165522067

$\vec{u}_k$	0.30713216972178869	0	4.9180617752243183
$\vec{U}$	-0.54997234489203628	-0.70710678118654757	-0.27713134132788841

Таблица 2. Результаты машинного расчета

Численный эксперимент проводился на компьютере с операционной системой Microsoft Windows 10 Pro, CPU 3.40 GHz, RAM 16 MB в среде MS VS 2021. Время затраченное на выполнение функции $VWU(t)$ составило порядка 0.01 ms. Частные случаи уравнений (2) приведены в Приложении 2.

Рывок играет важную роль во многих прикладных задачах (инженерия, физика и астрофизика, физиология и биомеханика, ...). Ненулевой рывок вызывает внезапное изменение ускорения, которое может иметь разрушительные последствия для физических систем.

Если, траектория движения объекта определяется достаточно сложными композициями математических функций (например, $y = \sin(\cos(\operatorname{tg}(\operatorname{sh}(\operatorname{ch}(\operatorname{th}(x)))))$), то точное вычисление их производных является сложной, а иногда и невыполнимой задачей. Описанный выше подход позволяет решить эту проблему.

Интересно отметить, что инженеры телескопа Хаббл учитывали даже производную рывка. В нашем случае, для учёта производной рывка необходимо заменить статическую библиотеку [Sdn3Lib.dll](#) на [Sdn4Lib.dll](#), которая позволяет точно вычислять производные до 4-го порядка включительно. Более того, используя подход, описанный в [3] можно точно вычислять производные до 8-го порядка включительно.

Приложение 1.

Вариант C# программного кода, реализующий обобщенные дуальные векторы 3-го порядка (статическая библиотека [Vec3Lib.dll](#)), с использованием статической библиотеки [Sdn3Lib.dll](#).

```
using System;
using System.Collections;
using D = Sdn3Lib.sdn3;
using V = Vec3Lib.sdn3V3;
using V3 = Vec3Lib.sdn3V3.vec3;

/-- function that defines absolute, translational, relative and jerky components
public static ArrayList VWU(double t)
{
    var p = R(t); var w = Ω(t);
    var vo = p.im1;
    var vp = w.re ^ p.re; /-- vector product AxB = A^B
    var v = vo + vp;
```

```

var wo = p.im2;
var wp = (w.im1 ^ p.re) + (w.re ^ (w.re ^ p.re));
var wk = 2.0 * (w.re ^ p.im1);
var w = wo + wp + wk;
var uo = p.im3;
var up = (w.im2 ^ p.re) + 2.0 * (w.im1 ^ (w.re ^ p.re))
        + (w.re ^ ((w.im1 ^ p.re) + (w.re ^ (w.re ^ p.re))));
var uk = 3.0 * ((w.re ^ p.im2) + (w.im1 ^ p.im1) + (w.re ^ (w.re ^ p.im1)));
var u = uo + up + uk;
V3[] V1 = { vo, vp, v }, W1 = { wo, wp, wk, w }, U1 = { uo, up, uk, u };
ArrayList r = new ArrayList(); r.Add(V1); r.Add(W1); r.Add(U1);
return r;
}

/-- circle of unit radius rotating about the y-axis
public static V Ω(double t) //Ω
{
    D T = new D(t, 1.0, 0.0, 0.0), z = T * T * T * T;
    var x = new V3(0.0, z.re, 0.0);
    var x1 = new V3(0.0, z.im1, 0.0);
    var x2 = new V3(0.0, z.im2, 0.0);
    var x3 = new V3(0.0, z.im3, 0.0);
    return new V(x, x1, x2, x3);
}

public static V R(double t)
{
    D T = new D(t, 1.0, 0.0, 0.0), x = D.cos(T), y = D.sin(T);
    return new V(new V3(x.re, y.re, 0.0),
                new V3(x.im1, y.im1, 0.0),
                new V3(x.im2, y.im2, 0.0),
                new V3(x.im3, y.im3, 0.0));
}

/-- call function VWU(t)
var vwu = VWU(Math.PI / 4.0);
/-- vwu[0]=[vo, vp, v]; vwu[1]=[wo, wp, wk, w]; vwu[2]=[uo, up, uk, u];

```

Приложение 2

- Система уравнений (2) при $\vec{\omega} = const$, $\vec{\rho} \neq const$

$$\begin{aligned}
 \vec{v}_o &= \dot{\vec{\rho}}, & \vec{v}_p &= \dot{\vec{r}} + \vec{\omega} \times \vec{\rho}; \\
 \vec{w}_o &= \ddot{\vec{\rho}}, & \vec{w}_p &= \ddot{\vec{r}} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}), & \vec{w}_k &= 2\vec{\omega} \times \dot{\vec{\rho}}; \\
 \vec{u}_o &= \ddot{\vec{\rho}}, & \vec{u}_p &= \ddot{\vec{r}} + \vec{\omega} \times (\vec{\omega} \times (\vec{\omega} \times \vec{\rho})), & \vec{u}_k &= 3(\vec{\omega} \times \ddot{\vec{\rho}} + \vec{\omega} \times (\vec{\omega} \times \dot{\vec{\rho}}));
 \end{aligned} \tag{2.1}$$

- Система уравнений (2) при $\vec{\omega} = const$, $\vec{\rho} = const$

$$\begin{aligned}
 \vec{v}_o &= 0, & \vec{v}_p &= \dot{\vec{r}} + \vec{\omega} \times \vec{\rho}; \\
 \vec{w}_o &= 0, & \vec{w}_p &= \ddot{\vec{r}} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}), & \vec{w}_k &= 0; \\
 \vec{u}_o &= 0, & \vec{u}_p &= \ddot{\vec{r}} + \vec{\omega} \times (\vec{\omega} \times (\vec{\omega} \times \vec{\rho})), & \vec{u}_k &= 0;
 \end{aligned} \tag{2.2}$$

- Система уравнений (2) при $\vec{\omega} = const, \vec{\rho} = const, \vec{r} = const$

$$\begin{aligned}\vec{v}_o &= 0, \quad \vec{v}_p = \vec{\omega} \times \vec{\rho}; \\ \vec{w}_o &= 0, \quad \vec{w}_p = \vec{\omega} \times (\vec{\omega} \times \vec{\rho}), \quad \vec{w}_k = 0; \\ \vec{u}_o &= 0, \quad \vec{u}_p = \vec{\omega} \times (\vec{\omega} \times (\vec{\omega} \times \vec{\rho})), \quad \vec{u}_k = 0;\end{aligned}\tag{2.3}$$

Равенство $\vec{d} = const$ означает, что вектор $\vec{d}$ имеет постоянную длину и направление.

- Система уравнений (2) при $\vec{\omega} = 0, \dot{\vec{r}} = const$

$$\begin{aligned}\vec{v}_o &= \dot{\vec{\rho}}, \quad \vec{v}_p = \dot{\vec{r}}; \\ \vec{w}_o &= \ddot{\vec{\rho}}, \quad \vec{w}_p = 0, \quad \vec{w}_k = 0; \\ \vec{u}_o &= \ddot{\vec{\rho}}, \quad \vec{u}_p = 0, \quad \vec{u}_k = 0;\end{aligned}\tag{2.4}$$

Эти соотношения фактически описывают преобразование Галилея в механике Ньютона.

- Система уравнений (2) с учетом $\vec{\omega} \times (\vec{\omega} \times \vec{d}) = -\alpha \vec{d}, \alpha = sec^{-2}$

$$\begin{aligned}\vec{v}_o &= \dot{\vec{\rho}}, \quad \vec{v}_p = \dot{\vec{r}}(t) + \vec{\omega}(t) \times \vec{\rho}(t); \\ \vec{w}_o &= \ddot{\vec{\rho}}, \quad \vec{w}_p = \ddot{\vec{r}} + \vec{\varepsilon} \times \vec{\rho} - \alpha \vec{\rho}, \quad \vec{w}_k = 2\vec{\omega} \times \dot{\vec{\rho}}; \\ \vec{u}_o &= \ddot{\vec{\rho}}, \quad \vec{u}_p = \ddot{\vec{r}} + \vec{\gamma} \times \vec{\rho} + 2\vec{\varepsilon} \times (\vec{\omega} \times \vec{\rho}) + \vec{\omega} \times (\vec{\varepsilon} \times \vec{\rho} - \alpha \vec{\rho}), \\ &\quad \vec{u}_k = 3(\vec{\omega} \times \ddot{\vec{\rho}} + \vec{\varepsilon} \times \dot{\vec{\rho}} - \alpha \dot{\vec{\rho}});\end{aligned}\tag{2.5}$$

Последнее позволяет сократить количество векторных произведений, что важно в вычислительных процессах.

ЛИТЕРАТУРА

1. Березкин Е. Н. Курс теоретической механики – М.: МГУ, 1974. – 647 с.
2. Олифер В. И. К рациональной теории обобщенных дуальных чисел – URL: https://viosolutions.amerihomesrealty.com/pdf/K_рациональной_теории_обобщенных_дуальных_чисел.pdf, – 2025.
3. Олифер В. И. К вычислению производных высокого порядка на основе гипер-дуальных чисел – URL: https://viosolutions.amerihomesrealty.com/pdf/Вычислению_производных_высокого_порядка.pdf, – 2025.

Абстракт

В данной публикации рассматривается вариант описания сложного движения материальной точки с учётом рывка. Описана процедура автоматического дифференцирования для такого движения с использованием обобщенных дуальных чисел 3-го порядка и векторов 3-го порядка. Представлена компьютерная реализация этого подхода для языка C# операционной системы Windows 10 Pro. Проведены численные эксперименты.

This publication considers a variant of description of complex motion of a material point taking into account a jerk. The procedure of automatic differentiation for such motion using generalized duals of 3rd order numbers and 3rd order vectors is described. A computer implementation of this approach for the C# language of the Windows 10 Pro operating system is presented. Numerical experiments are conducted.

Ключевые слова: *сложное движение материальной точки с учётом рывка, автоматическое дифференцирование, обобщенные дуальные числа, complex motion of a material point taking into account a jerk, automatic differentiation, generalized dual numbers.*

20 сентября 2025 г