К РАСЧЕТУ БАЛЛИСТИЧЕСКОЙ КРИВОЙ С ИСПОЛЬЗОВАНИЕМ АВТОМАТИЧАСКОГО ДИФФЕРЕНЦИРОВАНИЯ НА ОСНОВЕ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Олифер В. И.

Если снаряд при полете подвергается действию двух сил: силы тяжести и силы сопротивления воздуха (эффект Мангуса не учитывается), то его траектория (см. рис. 1) по форме представляет собой неравномерно выпуклую плавную кривую [1 - 3].

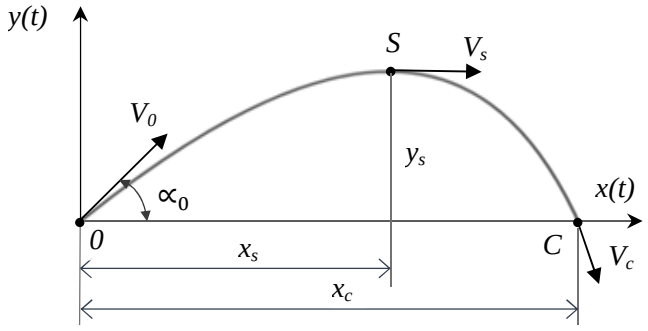

Рис. 1. Баллистическая кривая

На рис. 1 обозначено:

S – точка максимальной высоты подъема снаряда;

C – точка максимальной дальности полета снаряда;

$\propto_0$ – угол к горизонту в момент выстрела;

V_0 – начальная скорость снаряда;

V_s – скорость снаряда в точке S;

V_c – скорость снаряда в точке C;

y_s – максимальная высота подъема снаряда;

x_s –дальность полета снаряда до наивысшей точки траектории S;

x_c – максимальная дальность полета снаряда (до точки C).

Движение снаряда (при вышеуказанных условиях) в декартовых осях XOY описывается уравнениями [1 - 3]:

$$\begin{cases} x(t) = \dfrac{V_{x0}}{b}(1 - e^{-bt}), & (1) \\ y(t) = \dfrac{1}{b^2}\left(\mathrm{g} + bV_{y0}\right)(1 - e^{-bt}) - \dfrac{\mathrm{g}t}{b}, & (2) \end{cases}$$

где: $V_{x0} = V_0 \cos(\propto_0)$; $V_{y0} = V_0 \sin(\propto_0)$; $b = k\mathrm{g}$; k – коэффициент, характеризующий снаряд и его траекторию [1]; g = 9.81 – ускорение свободного падения.

Дальность полета снаряда определяется по уравнению (1) при известном времени полета снаряда t, которое можно найти из решения уравнения (2) при $y(t) = 0$. Однако, точное решение этого тресцендентного уравнения аналитически невозможно [3].

Некоторые характеристики баллистической кривой можно получить в аналитическом виде [2, 3]:

$$t = -\frac{\ln(1 - \frac{b \cdot x}{V_{x0}})}{b}, \tag{3}$$

$$t_s = \frac{\ln(V_{y0}\frac{b \cdot V_{y0}}{\mathrm{g}} + 1)}{b}, \tag{4}$$

$$y_s = (V_{y0} - t_s \cdot \mathrm{g})/b, \tag{5}$$

$$x_s = V_{x0} \cdot V_{y0}/(V_{y0} \cdot b + \mathrm{g}) \tag{6}$$

Подстановка (3) в (2) дает уравнение баллистической кривой в явном виде:

$$y(x, \propto_0) = q_1 \cdot \ln(1 - q_3(\propto_0) \cdot x) + q_2(\propto_0) \cdot x), \tag{7}$$

где: $q_1 = \frac{\mathrm{g}}{b^2}$, $q_2(\propto_0) = \frac{V_{y0} + \mathrm{g/b}}{V_{x0}}$, $q_3(\propto_0) = \frac{b}{V_{x0}}$

Используя уравнение (7) и формулы (3) - (6), рассмотрим итерационные решения следующих баллистических задач:

1. Найти (x_s, y_s) при заданных $\propto_0$, V_0, b:

$$x_s = \frac{V_{x0} \cdot V_{y0}}{V_{y0} \cdot b + \mathrm{g}}, \qquad y_s = \left(V_{y0} - \frac{\mathrm{g}}{b}\ln\left(1 + \frac{b \cdot V_{y0}}{\mathrm{g}}\right)\right)/b$$

2. Найти x_c при заданных $\propto_0$, V_0, b:

$x_c = \forall(x > x_s)$: $y(x) = 0 \Rightarrow x_0 = \frac{V_{x0}}{b} - 10$, $x_c = CHB(y(x, \propto_0), x_0)$

3. Найти $\propto_x$ при заданных $\underline{x_c}, V_0$, b:

$\propto_x = \forall \underline{x_c} > 0$: $y(\underline{x_c}) = 0 \Rightarrow \propto_o = \arcsin(\underline{x_c} \cdot \mathrm{g}/V_0^2)/2$, $\propto_x = CHB(y(\propto, \underline{x_c}), \propto_o)$

4. Найти $\propto_c$ (угол $\propto_c$, соответствующий $max(x_c)$) при заданных $V_0,\ b$:

$\propto_c = \max_{\propto_c \in l}(x_c),\quad l = [0, \pi/2]$

5. Найти $\propto$ при заданных $\underline{x},\ \underline{y},\ V_0,\ b$:

$\propto = \forall \underline{x},\ \underline{y} > 0:\ \ \underline{y}(\propto, \underline{x}) = 0$

Заметим, что задача 3 является частным случаем задачи 5 ($\underline{x} > 0,\ \underline{y} = 0$).

Как видно, первая задача имеет замкнутое решение, которое используется в других задачах, в том числе для поиска точек кривой безопасности.

Для решения второй, третьей и пятой задач будем применять итерационный метод Чебышева 4-го порядка $CHB(y(x),\ x_0)$ [4]:

$$x_{i+1} = x_i - [1 + \ (L_i + L_i^2)/2 - K_i/6]D_i, \quad i = 0, 1, 2, \dots , \tag{8}$$

где: $D_i = \frac{y(x_i)}{y'(x_i)},\ L_i = \ D_i \cdot \frac{y''(x_i)}{y'(x_i)},\ K_i = D_i^2 \cdot \frac{y'''(x_i)}{y'(x)}$ и $y',\ y'',\ y'''$– производные y по x соответствующего порядка, а x_0 – начальное приближение, которое должно обеспечивать сходимость метода к нужному корню.

И, наконец, четвертая задача связана с поиском максимума функции двух переменных, который реализуется совместно методами Чебышева и модифицированным методом золотого сечения [5]. Схема модифицированного метода золотого сечения для поиска максимума на отрезке $[a, b]$ имеет вид:

$$\varphi = \frac{3 - \sqrt{5}}{2},\ \ \propto_1 = a + \varphi \cdot (b - a),\ \ \propto_2 = a + b - \propto_1,\ \ x_1 = f(\propto_1),\quad x_2 = f(\propto_2)$$

R: $(x_1 > x_2) \Rightarrow b = \ \propto_2,\ \propto_2 = \propto_1,\ x_2 = \ x_1,$

$$\propto_1 = a + \varphi \cdot (b - a),\ \ x_1 = f(\propto_1),\ \ go\ to\ R;$$

$(x_1 \le x_2) \Rightarrow a = \propto_1,\ \propto_1 = \propto_2,\ x_1 = \ x_2,$

$$\propto_2 = a + b - \propto_1,\ \ x_2 = f(\propto_2),\ \ go\ to\ R;$$

где: $f(\propto)$ – возвращает результаты работы второй задачи.

Условием окончания вычислительного процесса служит выражение $|b - a| \le \delta$, а выходные результаты определяются выражениями $\propto = (\propto_1 + \propto_2)/2,\ \ x = f(\propto)$.

Для реализации метода Чебышева 4-го порядка, используемого в баллистических задачах 2, 3, 4 и 5, необходимо наличие простого и точного способа определения значений частных производных (до третьего порядка включительно) исходных функций. Это может быть успешно решено, если воспользоваться методом автоматического дифференцирования (АД) [6], суть которого заключается в создании нового типа данных и переопределении операций и базовых функций над ними.

Для рассматриваемого случая в качестве такого нового типа данных могут быть исползованы гипер-дуальные числа 3-го класса [6].

Гипер-дуальное число 3-го класса имеет представление в виде $Z = z + z_1\boldsymbol{\varepsilon} + z_2\boldsymbol{\omega} + z_3\boldsymbol{\gamma}$, $z,\ z_1,\ z_2,\ z_3 \in \mathbb{R}$. Параметр z называется главной (*Re* - действительной) частью гипер-дуального числа, а z_1, z_2, z_3 – его мнимыми (*Im1*, *Im2*, *Im3* - инфинитезимальными) частями. Абстрактные элементы $\boldsymbol{\varepsilon},\ \boldsymbol{\omega},\ \boldsymbol{\gamma}$ образуют базис мнимых частей гипер-дуального числа, отвечающий следующими правилам: $\boldsymbol{\varepsilon^2} = 2\boldsymbol{\omega},\ \boldsymbol{\varepsilon\omega} = 3\boldsymbol{\gamma},\ \boldsymbol{\varepsilon\gamma} = \boldsymbol{\omega\gamma} = \boldsymbol{\gamma^2} = \boldsymbol{\omega^2} = 0$.

Алгебраические операции над дуальными числами 3-го класса определяются формулами:

$$
\begin{aligned}
&X = x + x_1\boldsymbol{\varepsilon} + x_2\boldsymbol{\omega} + x_3\boldsymbol{\gamma}, \qquad Y = y + y_1\boldsymbol{\varepsilon} + y_2\boldsymbol{\omega} + y_3\boldsymbol{\gamma},\\
&X + Y = x + y + (x_1 + y_1)\boldsymbol{\varepsilon} + (x_2 + y_2)\boldsymbol{\omega} + (x_3 + y_3)\boldsymbol{\gamma},\\
&X{\cdot}Y = x{\cdot}y + (x{\cdot}y_1 + y{\cdot}x_1)\boldsymbol{\varepsilon} + (x{\cdot}y_2 + 2x_1{\cdot}y_1 + y{\cdot}x_2)\boldsymbol{\omega} + (x{\cdot}y_3 + y{\cdot}x_3 + 3(x_1{\cdot}y_2\\
&\qquad + y_1{\cdot}x_2))\boldsymbol{\gamma},\\
&Y^{-1} = y^{-1} - y^{-2}{\cdot}y_1\boldsymbol{\varepsilon} + y^{-2}(2{\cdot}y_1^2{\cdot}y^{-1} - y_2)\boldsymbol{\omega} + a^{-2}(6y_1{\cdot}y^{-1}(y_2 - y_1^2{\cdot}y^{-1})\\
&\qquad - y_3)\boldsymbol{\gamma},\\
&X/Y = X{\cdot}Y^{-1},
\end{aligned}
\tag{9}
$$

а функция гипер-дуального аргумента 3-го класса, после ее разложения в ряд Тейлора, имеет вид

$$
\begin{aligned}
F(X) = f(x) + x_1 f'(x)\boldsymbol{\varepsilon} + [x_2 f'(x) + x_1^2 f''(x)]\boldsymbol{\omega} +\\
+ [x_3 f'(x) + 3x_1 x_2 f''(x) + x_1^3 f'''(x)]\boldsymbol{\gamma}
\end{aligned}
\tag{10}
$$

При $x_1 = 1$ и $x_2 = x_3 = 0$ соотношение (10) принимает форму:

$$F(x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}) = f(x) + f'\boldsymbol{\varepsilon} + f''(x)\boldsymbol{\omega} + f'''(t)\boldsymbol{\gamma}, \quad f(x) = F(x + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}),$$

что позволяет представить (8) следующим образом

$$D_i = \frac{Y(X_i).Re}{Y(X_i).Im1},\ L_i = D_i \cdot \frac{Y(X_i).Im2}{Y(X_i).Im1},\ K_i = D_i^2 \cdot \frac{Y(X_i).Im3}{Y(X_i).Im1},$$

где: $X_i = x_i + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}$, $Y(X)$ – целевая функция $y(x)$, записанная в терминах гипер-дуальных чисел 3-го класса.

Компьютерная реализация гипер-дуальных чисел 3-го класса (Sdn3) была выполнена на языке программирования SWIFT 5 для macOS Sonoma в виде статической библиотеки (static library) SdnLibrary3 [6], которую можно скачать с интернет ресурса (см. Приложение 1), а затем добавить в свой Xcode проект.

С целью проверки адекватности работы описанного выше численного метода была проведена компьютерная верификация решений баллистических задач 1÷5 с использованием программного кода на языке SWIFT 5 (см. Приложение 1). Указанный SWIFT код оформлен в виде структуры Blst, которая включает 13 функций: α_x_y(), α_x( ), α_x_max(), xy_s(), x_c(x0), Y_x(x), Y_α(α), t_x(x), x_α(α), α_xy(), CHB (f:(), x0), new_α(α), RG (l:(a, b)). Первые три являются публичными (public), а остальные приватными (private). Функции Y_x(x), Y_α(α) описывают уравнение (7) в терминах Sdn3, причем для первой функции x – переменная, α – константа, а для второй наоборот. Коммуникации между функциями осуществляются с использованием картежа (tuple) gB.
Расчеты проводились в операционной среде macOS Sonoma 14.5 с CPU 2.3 GHz.
В табл. 1 представлены численные результаты для $\propto_0 = 45^o$, $V_0 = 800$, $b = 0.04$ и g = 9.81.

№ задачи	Результат	t_{mc}	i
1	x_s = 9,865.150484085429 y_s = 6,809.710422016575	0.005	-
2	x_c = 13,546.51128628792	0.039	4
3	$\propto_x = 9.5465^o$ *при* $\underline{x}$ = 12,000.0	0.007	2
4	$\propto_c = 25.86767^o$ x_c =15,873.066510851604	0.034	23
5	$\propto_1 = 21.64089^o$, $\propto_2 = 54.81854^o$ *при* $\underline{x}$ = 11,000.0, $\underline{y}$ = 2,500.0	0.058	6

Табл. 1. Результаты расчета при $\propto_0 = 45^o, V_0 = 800, b = 0.04, g = 9.81$;
$\boldsymbol{t_{mc}}$*– затраченное время в милисекундах ,* $\boldsymbol{i}$ *– количество итераций*

При других значениях угла $\propto_0$, полученные результаты даны в виде графиков (см. Рис. 2, 3).

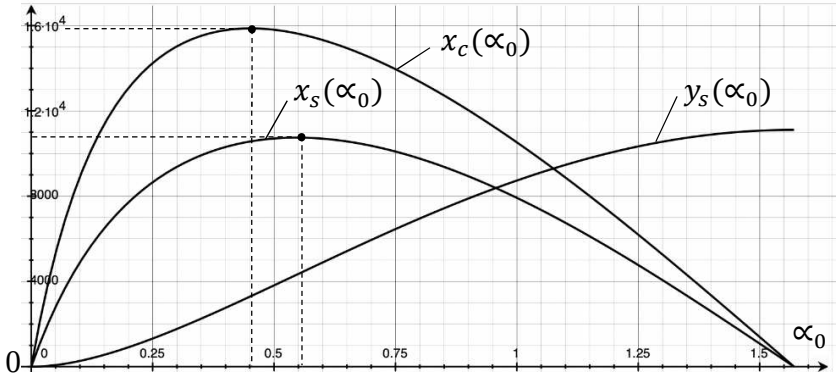

Рис. 2. Зависимости x_s, y_s *и* x_c *от* $\propto_0$; *(*$\propto_0$ *в радианах)*

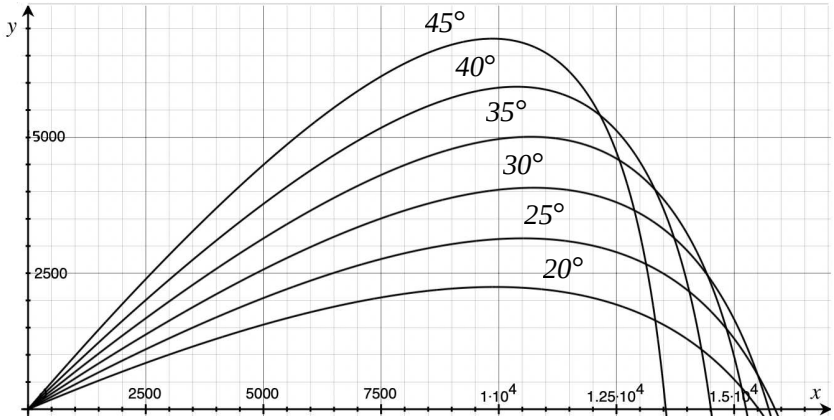

Рис. 3. Зависимость y(x) от $\propto_0$

Расчет всех пяти баллистических задач потребовал не более 0,14 мс. машинного времени, а точность решения значительно привышает реальные нужды. В практических вопросах наиболее востребованными являются третья и пятая задачи. Последняя дает два значения угла $\propto_0$, обеспечивающих попадание в цель. Причем первое значение $\propto_0$ соответствует более пологой траектории, а второе – более выпуклой. Следует отметить, что за одно обращение к функции Y_x(x) автоматически вычисляются основные геометрические характеристики кривой $(y,\ y',\ y'',\ y''')$ в исследуемой точке x. Полученные численные результаты соответствуют всем особенностям исследованной баллистической кривой, например, $\max_{\propto}(x_s) \nLeftrightarrow \max_{\propto}(x_c)$ – при заданном V_0, из максимума $x_s(\propto)$ не следует, что $x_c(\propto)$ будет максимально, и наоборот (см. рис. 2). Задача 2 также решалась путем численного интегрирования системы дифференциальнах уравнений (1) и (2), методом Рунге-Кутты 4-го порядка. Для достижения результата (с точностью 4-х знаков после запятой), представленного в табл. 1, потребовалось около 30 mc против 0.039 mc. SWIFT код реализующий указанную выше процедуру Рунге-Кутты дан в Приложении 2.

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS 14.5). Тип данных Sdn3 (супер-дуальные числа 3-го класса) приведен в [6], или можно скачать статическую библиотеку SdnLibrary3.zip и после распаковки добавить SdnLibrary3 в свой проект (как это сделать см. ReadMe.txt). Download SdnLibrary3.zip

```
import Foundation

      //-- Creating instance of structure Blst:
       var B = Blst (Vo: 800, b: 0.04, α0: 45.0);   //-- Task 1,2
       let rg = B.α_x_max();                        //-- Task 4
       let α  = B.α_x(x: xz).α;                     //-- Task 3
      let α12 = B.α_x_y(x: 11000.0, y: 2500.0);   //-- Task 5

struct Blst {
   var gB = (αg:0.0, α0:0.0, b:0.0, g:9.81,
             Vo:0.0, Vox:0.0, Voy:0.0, x:0.0, y:0.0,  xs:0.0, ys:0.0, xc:0.0, xmax:0.0, t:0.0);

   init(Vo:Double,b:Double,α0:Double) {
      gB.Vo = Vo;  gB.b = b; new_α(α: α0);
//--- task 1
          let xys = xy_s();   gB.xs = xys.xs;   gB.ys = xys.ys;
          gB.xmax = gB.Vox/gB.b;

//--- task 2
       gB.xc = x_c(x0: gB.xmax - 10).x;  gB.x = gB.xc;
      gB.y = Y_x(x: gB.x).re;      gB.t = t_x(x: gB.x);
   }

//--- task 3
    public mutating func α_x(x:Double)->(α:Double, t:Double){
      let x1 = gB.x,  gB.x = x,
          α0 = asin(x*gB.g/(gB.Vo*gB.Vo))/2.0,
           α = CHB (f: Y_α, x0: α0).x,
           a = α*180/Double.pi;
      gB.x = x1;
      let t = t_x(x: x);
      return (α:a,   t:t);
   }

//--- task 4
   public mutating func α_x_max()->(α:Double,x:Double, i:Int){
      return RG (l: (a: 0.0, b: 45.0));
   }

//--- task 5
   public mutating func α_x_y(x:Double, y:Double)->(α1:Double,α2:Double){
      var α1 = 10.0*Double.pi/180.0, α2 = Double.nan;
      gB.x = x;   gB.y = y;
      let q1 = gB.g/(gB.b*gB.b);
```

```
    for α:Double in stride(from: 10, through: 80, by: 1){
        var αr = α*Double.pi/180.0;
        var Vx = gB.Vo*cos(αr), Vy = gB.Vo*sin(αr);
        let q2 =  (Vy + gB.g/gB.b)/Vx,  q3 = gB.b/Vx;
        let z = q2*gB.x + q1*log(1.0 - q3*gB.x) - gB.y;
        if z.isNaN {break} else {α2 = α}
    }
    let ar2 =   α2*Double.pi/180.0,
      α_1 = CHB (f: α_xy, x0: ar1).x,
      α_2 = CHB (f: α_xy, x0: ar2).x;
    return (α1: α_1*180/Double.pi,   α2: α_2*180/Double.pi);
  }

//--- private functions:

  private func xy_s()->(xs:Double, ys:Double){
      let xs = gB.Vox*gB.Voy/(gB.Voy*gB.b + gB.g),
          ys = (gB.Voy - (gB.g/gB.b)*log(1.0 + gB.b*gB.Voy/gB.g))/gB.b;
    return (xs:xs,   ys:ys);
  }

  private func x_c(x0:Double)->(x:Double, i:Int){return CHB (f: Y_x, x0: x0c}

  private func Y_x(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0),
        T = -Sdn3.ln(X: Sdn3(re: 1) - gB.b*X/gB.Vox)/gB.b,
       p1 = (gB.b*gB.Voy + gB.g)/(gB.b*gB.b),  p2 = gB.g/gB.b,
        Y = p1*(Sdn3(re: 1.0) - Sdn3.exp(X: -gB.b*T)) - p2*T;
    return Y;
  }

  private func Y_α(α:Double)->Sdn3{
    let x = gB.x,  β = Sdn3(re: α, im1: 1.0, im2: 0.0, im3: 0.0),
      Vx = gB.Vo*Sdn3.cos(X: β),  Vy = gB.Vo*Sdn3.sin(X: β),
       T = -Sdn3.ln(X: Sdn3(re: 1) - Sdn3(re: gB.b*x)/Vx)/gB.b,
      p1 = (gB.b*Vy + Sdn3(re: gB.g))/(gB.b*gB.b),  p2 = gB.g/gB.b,
       Y = p1*(Sdn3(re: 1.0) - Sdn3.exp(X: -gB.b*T)) - p2*T;
    return Y;
  }

 private func α_xy(α:Double)->Sdn3{return Y_α(α: α) - Sdn3(re: gB.y)}

  private func t_x(x:Double)->Double{
    let Vx = gB.Vo*cos(gB.α0);
```

```
    return -log(1 - gB.b*x/Vx)/gB.b;
  }

  private mutating func x_α(α:Double)->Double {
    new_α(α: α);
    return x_c(x0: gB.xmax - 10).x;
  }

  private  mutating func new_α(α:Double){
    gB.α0  = α*Double.pi/180.0;    gB.αg = α;
    gB.Vox = gB.Vo*cos(gB.α0);  gB.Voy = gB.Vo*sin(gB.α0);
    gB.xmax = gB.Vox/gB.b;
  }

  private  func CHB (f:(Double) -> Sdn3,  x0:Double)->(x:Double,i:Int){
    var i = 0, F = f(x0),  x = x0, D = 0.0,  L = 0.0,  K = 0.0;
    repeat{
      D = F.re/F.im1; L = D*F.im2/F.im1; K = D*D*F.im3/F.im1;
      x -= (1.0 + 0.5*(L + L*L) - K/6.0)*D;
      if x >= gB.xmax  {x = gB.xmax;  break}
      i += 1;  F = f(x); if i > 1000 {break}
    } while (abs(F.re) >= 1E-10);
    return (x,  i);
  }

  private  mutating func RG (l:(a:Double, b:Double))->(α:Double,x:Double, i:Int){
    let φ = (3.0 - sqrt(5.0))/2.0,   δ:Double = 1E-3,     gB1 = gB;
    var i = 0, l = l, α1 = l.a + φ*(l.b - l.a), α2 = l.a + l.b - α1,
      x1 = x_α(α: α1),    x2 = x_α(α: α2);
    while (l.b - l.a) > δ{
      if x1>x2 {l.b = α2;    α2 = α1;    x2 = x1;
        α1 = l.a + φ*(l.b - l.a);      x1 = x_α(α: α1)}
      else {l.a = α1;   α1 = α2;       x1 = x2;
        α2 = l.a + l.b - α1;   x2 = x_α(α: α2)}
      i += 1;
    }
    let α = (α1 + α2)/2.0,    x = x_α(α: α);
    gB = gB1;
    return (α:α,  x:x,  i:i);
  }
}
```

Приложение 2.

Код, реализующий процедуру Рунге-Кутты, для системы (1–2) представленной в виде [2]:

$$\begin{cases} \dot{V}_x(t) = -bV_x(t), \\ \dot{V}_y(t) = -\left(\mathrm{g} + bV_y(t)\right) \end{cases}$$

```
import Foundation
import simd
struct Blst_RK{
   // Task parameters
   let v0: Double = 800;       // starting speed
   let θ: Double = .pi/4;      // shooting angle, radians (45 degrees)
   let g: Double = 9.81;       // acceleration of gravity
   let b:Double = 0.04;
   var x0, y0, vx0, vy0, dt:Double;
   var state:simd_double4;
   var trajectory: [simd_double2] = [];

   init(){
      // Initial conditions
      x0 = 0.0;  y0 = 0.0;   vx0 = v0 * cos(θ);    vy0 = v0 * sin(θ);
      // Initial state
      state = simd_double4(x0, y0, vx0, vy0);
      // Numerical solution parameters
      dt = 0.01;             // time step
      // Numerical solution
      while state[1] >= 0 {
         trajectory.append(simd_double2(state[0], state[1]));
         state = rungeKutta(state: state, dt: dt);
      }
      // Output of motion trajectory
      for point in trajectory {
        print("x: \(point[0]), y: \(point[1])")
      }
   }

   // Functions for equations of motion
   func derivatives(state: simd_double4) -> simd_double4 {
      let vx = state[2], vy = state[3],
          ax = -b*vx,  ay = -(g + b*vy);
      return simd_double4(vx, vy, ax, ay);
   }
```

```
    // Runge-Kutta method 4th order
    func rungeKutta(state: simd_double4, dt: Double) -> simd_double4 {
        let k1 = derivatives(state: state),
            k2 = derivatives(state: state + 0.5 * dt * k1),
            k3 = derivatives(state: state + 0.5 * dt * k2),
            k4 = derivatives(state: state + dt * k3);
        return state + (dt / 6) * (k1 + 2 * (k2 + k3) + k4);
    }
}
```

ЛИТЕРАТУРА

1. *Балаганский И. А*. Основы баллистики и аэродинамики.– Новосибирск : Изд-во НГТУ, 2017. – 200 с.

2. *Амельянчик А.И., Горбач Н.И.* К вопросу о движении артиллерийского снаряда.– Теоретическая и прикладная механика : международный научно-технический сборник. – Вып. 24. – 2009. – с. 247-260.

3. *Корзун А.С., Крайник Д.А., Лысанович П.В., Горбач Н.И., Гурвич Ю.А*. Расчёт дальности полета артиллерийского снаряда в декартовых осях координа. – Белорусский национальный технический университет, Минск, 2010.

4. *Антонюк П.* О первой работе П. Л. Чебышева. – Научно-Исслед. Семинар по истории и методологии мат. и мех.: МГУ, 2014. –14 с.

5.*Васильев В.П.* Численные методы решения экстремальных задач. - М.: Наука, 1980. – 518с.

6. *Олифер В. И.* Автоматическое дифференцирование на основе супер-дуальных чисел. – URL:http://viosolutions.amerihomesrealty.com/pdf/Автоматическое_дифференцирование_на_основе_супер-дуальных_чисел.pdf, 2020.

Абстракт

В данной публикации рассматривается расчет баллистической кривой, основанный на итерационных формулах Чебышева, золотого сечения и автоматического дифференцирования со специальными дуальными чиселами (супер-дуальными чиселами 3-го класса). Представлена компьютерная реализация этого подхода для языка SWIFT операционной системы macOS. Проведены численные эксперименты.

Ключевые слова: *баллистическая кривая, итерационные методы, автоматическое дифференцирование, супер-дуальные числа, ballistic curve, iterative methods, automatic differentiation, super-dual numbers.*

20 июля 2024 г