

ЧИСЛЕННОЕ РЕШЕНИЕ ПАРАМЕТРИЧЕСКИХ УРАВНЕНИЙ С ИСПОЛЬЗОВАНИЕМ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Олифер В. И.

Параметрическое уравнение определяет группу величин как функции одной или нескольких независимых переменных, называемых параметрами [1]. Эти уравнения обычно используются для выражения координат точек, составляющих геометрические объекты, такие как кривые или поверхности. Так, например, одно-параметрическое уравнение для плоских кривых (которое будет рассматриваться далее) имеет вид

$$(\varphi(t), f(t)) \equiv \begin{cases} x = \varphi(t) \\ y = f(t) \end{cases}, \quad (1)$$

где: $\varphi(t)$, $f(t)$ – некоторые аналитические функции, определенные на t ; x и y – декартовы координаты, а t – параметр с областью определения $t_- \leq t \leq t_+$.

Параметрическое представление даёт такое важное преимущество, что позволяет изучать неявные функции в тех случаях, когда их приведение к явному виду иначе как через параметры затруднительно или невозможно через элементарные функции.

Для численного определения корня уравнения (1) необходимо тем или иным итерационным методом найти корень t_* уравнения $f(t) = 0$, а затем вычислить $x_* = \varphi(t_*)$. Реализацию первого шага будем осуществлять по итерационной формуле Чебышева 4-го порядка, которая имеет вид [2]:

$$t_{i+1} = t_i - \frac{f(t_i)}{\dot{f}(t_i)} - \frac{1}{2} \left(\frac{f(t_i)}{\dot{f}(t_i)} \right)^2 \frac{\ddot{f}(t_i)}{\dot{f}(t_i)} - \frac{1}{2} \left(\frac{f(t_i)}{\dot{f}(t_i)} \right)^3 \left[\left(\frac{\ddot{f}(t_i)}{\dot{f}(t_i)} \right)^2 - \frac{1}{3} \frac{\dddot{f}(t_i)}{\dot{f}(t_i)} \right] =$$
$$= x_i - [1 + (L_i + L_i^2)/2 - K_i/6] D_i, \quad i = 0, 1, 2, \dots,$$

где: $L_i = D_i \frac{\ddot{f}(t_i)}{\dot{f}(t_i)}$, $K_i = D_i^2 \frac{\dddot{f}(t_i)}{\dot{f}(t_i)}$, $D_i = \frac{f(t_i)}{\dot{f}(t_i)}$, $(\dot{\cdot}) = \frac{d(\cdot)}{dt}$

Условие прекращения итерационного процесса (2) определяя выражением $|f(t_{i+1})| \leq \delta$, где δ – заданная малая величина, например, $\delta = 10^{-15}$. Результатом является численное значение корня t_* уравнения (1), обеспечивающее выполнение условия $|f(t_*)| \leq \delta$.

По найденному корню t_* нетрудно получить $x_* = \varphi(t_*)$ и производные y по x в точке t_* [1]:

$$y' = \dot{f}/\dot{\phi}, \quad y'' = (\ddot{f}\dot{\phi} - \dot{f}\ddot{\phi})/\dot{\phi}^3, \quad y''' = [\dot{\phi}(\ddot{f}\dot{\phi} - 3\dot{f}\ddot{\phi}) + \dot{f}(3\dot{\phi}^2 - \ddot{\phi}\dot{\phi})]/\dot{\phi}^5,$$

$$\dots, \quad y^{(n)} = (y^{(n-1)})', \quad \text{где } (\cdot)' = \frac{d(\cdot)}{dx}$$

Последнии позволяют определить геометрические характеристики кривой, заданной уравнением (1), в точке t_* [3]. По найденным значениям $f(t_*)$, $\dot{f}(t_*)$, $\ddot{f}(t_*)$ и $\ddot{f}(t_*)$ также не трудно определить кратность корня t_* [1].

Заметим, что вообще говоря, функции $\varphi(t)$, $f(t)$ могут быть весьма сложными композициями элементарных функций, представленные программным кодом. В этом случае точное вычисление производных может оказаться весьма трудоёмкой или даже невыполнимой задачей.

Так как для численной реализации формулы Чебышева 4-го порядка (2) необходимо вычисление $f(t)$, $\dot{f}(t)$, $\ddot{f}(t)$ и $\ddot{f}(t)$, то будем использовать супер-дуальные числа 3-го класса [4] с базисом $\{1, \varepsilon, \omega, \gamma\}$. Правила умножения базисов мнимых частей ε , ω и γ ($\varepsilon^2 = 2\omega$, $\varepsilon\omega = 3\gamma$, $\varepsilon\gamma = \omega\gamma = \gamma^2 = \omega^2 = 0$) можно представить в виде таблицы:

\times	ε	ω	γ
ε	2ω	3γ	0
ω	3γ	0	0
γ	0	0	0

Табл. 1. Умножение базисов мнимых частей для 3-го класса супер-дуальных чисел (ε , ω и γ).

Алгебраические операции над супер-дуальными числами 3-го класса приводятся в [4]. Произвольная функция супер-дуального аргумента 3-го класса после ее разложения в ряд Тейлора принимает вид ($T = t + t_1\varepsilon + t_2\omega + t_3\gamma$):

$$F(T) = f(t) + t_1\dot{f}(t)\varepsilon + [t_2\dot{f}(t) + t_1^2\ddot{f}(t)]\omega + [t_3\dot{f}(t) + 3t_1t_2\ddot{f}(t) + t_1^3\ddot{f}(t)]\gamma,$$

$$F(t + \varepsilon + 0\omega + 0\gamma) = f(t) + \dot{f}(t)\varepsilon + \ddot{f}(t)\omega + \ddot{f}(t)\gamma, \tag{3}$$

где $f(t) = F(t + 0\varepsilon + 0\omega + 0\gamma)$

Для выделения главной и мнимых частей T вводятся соответствующие операции:

$$t = T.Re, \quad t_1 = T.Im1, \quad t_2 = T.Im2, \quad t_3 = T.Im3 \tag{4}$$

Тогда

$$f(t) = F(\tilde{T}).Re, \dot{f}(t) = F(\tilde{T}).Im1, \ddot{f}(t) = F(\tilde{T}).Im2, \dddot{f}(t) = F(\tilde{T}).Im3, \quad (5)$$

где: $\tilde{T} = t + \varepsilon + 0\omega + 0\gamma$ и компоненты формулы (2) вычисляются по формулам

$$L_i = D_i \frac{F(\tilde{T}_i).Im2}{F(\tilde{T}_i).Im1}, \quad K_i = D_i^2 \frac{F(\tilde{T}_i).Im3}{F(\tilde{T}_i).Im1}, \quad D_i = \frac{F(\tilde{T}_i).Re}{F(\tilde{T}_i).Im1} \quad (6)$$

Компьютерная реализация супер-дуальных чисел 3-го класса (Sdn3) была выполнена [4] на языке программирования SWIFT 5 для macOS 13.3 в виде статической библиотеки (static library) SdnLibrary3, которую можно скачать с интернет ресурса (см. Приложение 1), а затем добавить в свой Xcode проект. Для выполнения численных экспериментов по формулам (2) и (6) в Приложение 1 приводится итерационная процедура СНВ_P(...), реализующая метод Чебышева 4-го порядка, а также процедура определения кратности полученного корня RootK(...). Там же представлены процедуры, описывающие некоторые тестируемые функции.

$(x(t), y(t))$	t_0	t_*	x_*	Кол-во итераций
$\begin{cases} x = t^2 \\ y = t^3/3 - t \end{cases}$	1.4	1.732050807568877	3.0000000000000000	3
$\begin{cases} x = \sin(t) \\ y = \cos(t) \end{cases}$	1.0	1.570796326794896	1.0000000000000000	2
$\begin{cases} x = ch(t) \\ y = sh(t) \end{cases}$	3.0	0.0000000000000000	1.0000000000000000	3
$\begin{cases} x = t - \sin(t) \\ y = 1 - \cos(t) \end{cases}$	2.0	-2.46796988271·10 ⁻⁸	0.0000000000000000	12
$\begin{cases} x = \cos^3(t) \\ y = \sin^3(t) \end{cases}$	2.0	3.141586362255838	-0.999999999940629	11
$\begin{cases} x = \cos^4(t) \\ y = \sin^4(t) \end{cases}$	2.0	3.141472276192888	0.999999971018565	16
$\begin{cases} x = \sin(t) \\ y = \cos(2t) \end{cases}$	2.0	2.356194490192345	0.707106781186548	2
$\begin{cases} x = (1 - t^2)/(1 + t^2) \\ y = 2t/(1 + t^2) \end{cases}$	0.4	0.0000000000000000	1.0000000000000000	3

$\begin{cases} x = 2t/(1-t^2) \\ y = \frac{1+t^2}{1-t^2} - 2 \end{cases}$	0.9	0.577350269189625	1.732050807568877	3
$\begin{cases} x = 2\cos(t) + \cos(\frac{t}{2}) \\ y = 2\sin(t) - \sin(\frac{t}{2}) \end{cases}$	2.0	2.636232143305636	-1.5000000000000000	3
$\begin{cases} x = 10t - \sin(10t) \\ y = 1 + \cos(10t) \end{cases}$	1.1	0.942477799520410	9.4247777802963882	14
$\begin{cases} x = 1 - e^{-t} \\ y = 2(1 - e^{-t}) - t \end{cases}$	0.0	1.593624260040040	0.7968121300200200	3

Табл. 2. Результаты тестируемых задач

Аналогичный подход можно использовать и для нахождения экстремума кривой, заданной уравнением (1). Для этого воспользуемся итерационной формулой Чебышева-Хэлли третьего порядка [5]

$$t_{i+1} = t_i - \left[1 + \frac{1}{2} \cdot \frac{L(t_i)}{1 - \alpha L(t_i)} \right] D(t_i), \quad \alpha \in \mathbb{R}, \quad i = 0, 1, 2, \dots \quad (7)$$

где:

$$L(t_i) = D(t_i) \cdot \dot{\psi}(t_i) / \dot{\psi}(t_i), \quad D(t_i) = \psi(t_i) / \dot{\psi}(t_i), \quad \psi(t) = \dot{f}(t), \quad \dot{\psi}(t) = \ddot{f}(t), \quad \ddot{\psi}(t) = \dddot{f}(t)$$

или в терминах супер-дуальных чисел 3-го класса:

$$D(t) = \frac{F(\tilde{T}).Im1}{F(\tilde{T}).Im2}, \quad L(t) = \frac{D(t) \cdot F(\tilde{T}).Im3}{F(\tilde{T}).Im2},$$

Необходимые и достаточные условия оптимальности функции $f(t)$ служат следующие соотношения:

$$\dot{f}(t) = 0, \quad \ddot{f}(t) > 0 \quad - \text{min},$$

$$\dot{f}(t) = 0, \quad \ddot{f}(t) < 0 \quad - \text{max},$$

$$\dot{f}(t) = 0, \quad \ddot{f}(t) = 0 \quad - \text{inflection point}$$

Прекращение итерационного процесса (7) определятся выражением $|\dot{f}(t_{i+1})| \leq \delta$.

В Приложение 1 приводится итерационная процедура MinMax_P (...), реализующая метод вычисления экстремума по формуле (7). В табл. 3 представлены результаты некоторых тестируемых функций при $\alpha = 3/4$.

$(x(t), y(t))$	t_0	t_*/x_*	$f(t_*)$	Кол-во итераций
$\begin{cases} x = t^2 \\ y = t^3/3 - t \end{cases}$	0.1	$\frac{1.0000000000000000}{1.0000000000000000}$	-0.6666666666666666 <i>min</i>	3
	-0.1	$\frac{-1.0000000000000000}{1.0000000000000000}$	0.6666666666666666 <i>max</i>	3
$\begin{cases} x = \cos(t) \\ y = \sin(t) \end{cases}$	1.0	$\frac{1.570796326794896}{0.0000000000000000}$	1.0000000000000000 <i>max</i>	2
	-1.0	$\frac{-1.570796326794896}{0.0000000000000000}$	-1.0000000000000000 <i>min</i>	2
$\begin{cases} x = 1 - e^{-t} \\ y = 2(1 - e^{-t}) - t \end{cases}$	0.0	$\frac{0.693147180559945}{0.5000000000000000}$	0.306852819440054 <i>max</i>	2

Табл. 3. Результаты поиска экстремума тестируемых функций

Последние уравнения табл. 2 и 3 соответствуют упрощенному варианту баллистической кривой [6] при $v_{0x} = v_{0y} = 1$, $g/\beta = 1$, $\beta = b/m$, где: g, b и m – ускорение свободного падения, коэффициент сопротивления среды и масса снаряда соответственно, а v_{0x} и v_{0y} – начальные горизонтальная и вертикальная скорости. На рис.1 представлены составляющие указанного варианта баллистической кривой: 1 – $x(t)$; 2 – $y(t)$; 3 – $(x(t), y(t))$.

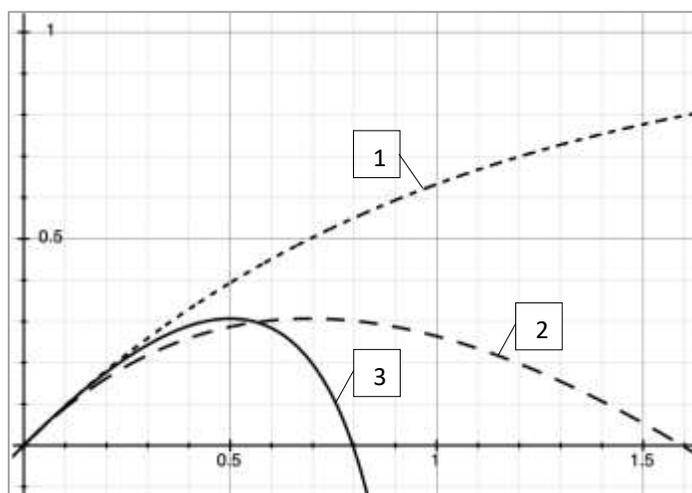


Рис. 1. Упрощенный вариант баллистической кривой

Эти два метода СНВ_P(...) и MinMax_P (...) требуют тщательного определения начального приближения, впрочем, как и другие итерационные методы. В любом случае,

целесообразно предварительное исследование исходной функции на интересующей области её определения.

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS 13.3). Тип данных Sdn3 (супер-дуальные числа 3-го класса) приведен в [4], или можно скачать статическую библиотеку SdnLibrary3 и после распаковки добавить SdnLibrary3 в свой проект (как это сделать см. ReadMe.txt). [Download SdnLibrary3.zip](#)

```
import Foundation;
import SdnLibrary3; // add this one if you use static library SdnLibrary3

let  $\delta$ :Double = 1E-15;
let Nmax = 1000; // max number of iterations

enum minmax:Int{
    case Min = 0
    case Max
    case Inf
}

// INPUT DATA:
// f and  $\varphi$  – functions pointers, to – start point
// OUTPUT DATA:
// (t, x, i) – solution and number of iterations

func CHB_P(f:(Double) -> Sdn3,  $\varphi$ :(Double)->Double, t0:Double)->(t:Double,x:Double,i:Int){
    var i = -1, F = f(t0), t = t0, 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;
        t -= (1.0 + 0.5*(L + L*L) - K/6.0)*D;
        i += 1; F = f(t);
    } while (abs(F.re) >=  $\delta$ );
    return (t,  $\varphi$ (t), i);
}

// INPUT DATA:
// f – function pointer, t – root of f(t)
```

```
// OUTPUT DATA:
// k – root multiplicity
// k = -1; t is not root
// k = 0, 1, 2, 3 - not less 3

func RootK(f:(Double) -> Sdn3, t: Double)->Int{
    var k = -1, F = f(t);
    if F.re != 0 {return k} else {k = 0}
    if F.im1 != 0 {k = 1}
    else if F.im2 != 0 {k = 2} else if F.im3 != 0 {k = 3}
    else {k = 3}
    return k;

// call function CHB_P()
let t = CHB_P(f: f_p, φ: φ_p, t0: 3.0); // t = (t:Double,x:Double,i:Int)

// examples of functions under study
func f_p(t:Double)->Sdn3{return Sdn3.sh(X: Sdn3(re: t, im1: 1.0, im2: 0.0, im3: 0.0))}

func φ_p(t:Double)->Double{return Sdn3.ch(x: t).re}

func φ_p2(t:Double)->Double{return 1.0 - Darwin.exp(-t)}

func f_p2(t:Double)->Sdn3{
    let T = Sdn3(re: t, im1: 1.0, im2: 0.0, im3: 0.0);
    return 2.0*(Sdn3(re: 1.0) - Sdn3.exp(X: -T)) - T}

// INPUT DATA:
// f and φ – functions pointers, to – start point, α – parameter of method
// OUTPUT DATA:
// (t, x, f, extr, i) – solution and number of iterations

func MinMax_P(f:(Double) -> Sdn3, φ:(Double)->Double, t0: Double, α:Double)->
(t:Double, x:Double, f:Double, extr:minmax, i:Int){
var i = -1, ti = t0, D = 0.0, L = 0.0, F = f(t0), extr = minmax.Inf;
repeat{
    if F.im1 == 0.0 || F.im2 == 0.0 { break }
    D = F.im1/F.im2; L = D*F.im3/F.im2;
    ti -= (1.0 + 0.5*L/(1.0 - α*L))*D;
    i += 1; F = f(ti);
    if i > Nmax {break}
} while (abs(F.im1) >= δ);
if F.im2 > 0 {extr = minmax.Min} else if F.im2 < 0 {extr = minmax.Max}
return (t:ti, x: φ(ti), f:F.re, extr: extr, i:i);
}
```

```
// call function MinMax_P ()  
let tmax = MinMax_P(f: f_p2, φ: φ_p2, t0: 0.0, α: 3/4);  
  
// call function RootK ()  
let k = RootK (f: f_p2, t: 0.577350269189625);
```

ЛИТЕРАТУРА

1. *Harris J.* Handbook of mathematics and computational science, Springer, New York, 1998.
2. *Олифер В. И.* Численное исследование метода Чебышева 4-го порядка с использованием супер-дуальных чисел 3-го класса
– URL: http://viosolutions.amerihomesrealty.com/pdf/Численное_исследование_метода_чебышева_4-го_порядка.pdf, 2023
3. *Олифер В. И.* Дифференциальная геометрия кривых в терминах гипер-дуальных чисел
– URL: http://viosolutions.amerihomesrealty.com/pdf/Дифференциальная_геометрия_кривых_в_терминах_гипер-дуальных_чисел.pdf, 2022
4. *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.
5. *Olifer V. I.* Implementation of Chebyshev-Halley type methods based on hyper-dual numbers.– URL: http://viosolutions.amerihomesrealty.com/pdf/Implementation_of_Chebyshev-Halley_type_methods_based_on_hyper-dual_numbers.pdf, 2023.
6. *Балаганский И. А.* Основы баллистики и аэродинамики.– Новосибирск : Изд-во НГТУ, 2017. – 200 с.

Абстракт

В данной публикации рассматривается метод решения параметрических уравнений на основе итерационных формул Чебышева-Хэлли и автоматического дифференцирования с использованием специальных дуальных чисел (супер-дуальных чисел 3-го класса). Представлена компьютерная реализация этого подхода для языка SWIFT операционной системы macOS. Проведены численные эксперименты.

Ключевые слова: *итерационные методы решения параметрических уравнений, автоматическое дифференцирование, супер-дуальные числа, iterative methods for solving parametric equations, automatic differentiation, super-dual numbers.*

20 апреля 2024 г.