

РЕАЛИЗАЦИЯ МЕТОДА ЧЕБЫШЕВА 5-го ПОРЯДКА НА БАЗЕ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ 4-го ПОРЯДКА

Олифер В.И.

Расчетная схема одношаговой (одноточечной) численной реализации эталонной итерационной формулы Чебышева 5-го порядка имеет вид [1, 2]:

$$\begin{aligned} x_{i+1} = x_i - D_i - \frac{1}{2}(D_i)^2 \left[\frac{f''(x_i)}{f'(x_i)} \right] - (D_i)^3 \left[\frac{1}{2} \left(\frac{f''(x_i)}{f'(x_i)} \right)^2 - \frac{1}{6} \frac{f'''(x_i)}{f'(x_i)} \right] \\ - (D_i)^4 \left[\frac{5}{8} \left(\frac{f''(x_i)}{f'(x_i)} \right)^3 - \frac{5}{12} \frac{f''(x_i)f'''(x_i)}{f'(x_i)^2} + \frac{f^{IV}(x_i)}{f'(x_i)} \right] = \\ = x_i - D_i \left[1 + \frac{1}{2}(L_i + M_i) - \frac{1}{6}K_i + Q_i \right], \end{aligned} \quad (1)$$

где: $D_i = \frac{f(x_i)}{f'(x_i)}$, $L_i = D_i \frac{f''(x_i)}{f'(x_i)}$, $M_i = L_i^2$, $K_i = D_i^2 \frac{f'''(x_i)}{f'(x_i)}$,
 $Q_i = D_i^3 * \left[\frac{L_i}{D_i^3} \left(\frac{5}{8}M_i - \frac{5}{12}K_i \right) + \frac{f^{IV}(x_i)}{f'(x_i)} \right]$, $\forall i = 0, 1, 2, \dots$

Итерационная схема (1) получена путем разложения в ряд Тейлора функции обратной к исходной $f(x)$. По мнению самого Чебышева П. Л. [1] этот путь является наиболее естественным и общим для разработки итерационных схем. Удержание разных членов в (1) дает формулы для методов Чебышева со 2-го по 5-й порядок [2]. Главный недостаток метода (1) состоит в том, что в нём участвуют производные функции $f(x)$ с 1-го до 4-го порядка, вычисление которых весьма затруднительно.

Традиционные способы вычисления даже производной $f'(x)$ на практике сопряжены с трудностями и имеют существенные недостатки. Реализация вычислений $f'(x)$ в виде отдельной программной процедуры является избыточным решением и нуждается в предварительном анализе $f'(x)$. Использование метода численного дифференцирования (разделенных разностей) требует компромисса между погрешностями усечения и округления, т. е. нахождения оптимальной величины приращения аргумента x . Более того, задача численного дифференцирования является некорректной и ее нужно решать методами регуляции [3]. Еще большие проблемы возникают при вычислении производных высших порядков. О гипер-сложных функциях, например,

$f(x) = f(f_1(f_2(f_3(f_4(f_5(\dots))))))$ и говорить нечего. Пример такой функции рассмотрен далее в численном эксперименте.

Полностью обойти указанные трудности позволяет метод автоматическое дифференцирование (AD), связанный с вычислением точных (с машинной точностью) значений производных функции, представленной компьютерным кодом [4, 5]. Суть AD заключается в создании нового типа данных и переопределении операций и базовых функций над ними. В нашем случае, в качестве такого нового типа данных могут быть использованы гипер-дуальные числа 4-го класса (порядка) [5].

Гипер-дуальное число 4-го класса (порядка) имеет представление в виде $Z = z + z_1\varepsilon + z_2\omega + z_3\gamma + z_4\eta$, $\forall z, z_1, z_2, z_3, z_4 \in \mathbb{R}$. Параметр z называется главной (Re – действительной) частью гипер-дуального числа, а z_1, z_2, z_3, z_4 – его мнимыми ($Im1, Im2, Im3, Im4$ – инфинитезимальными) частями. Абстрактные элементы $\varepsilon, \omega, \gamma, \eta$ образуют базис мнимых частей гипер-дуального числа, отвечающий следующими правилам: $\varepsilon^2 = 2\omega$, $\varepsilon\omega = 3\gamma$, $\varepsilon\gamma = 4\eta$, $\omega^2 = 6\eta$, $\varepsilon\eta = \omega\gamma = \omega\eta = \gamma\eta = \gamma^2 = \eta^2 = 0$. Таблица умножения элементов мнимого базиса симметрична относительно диагонали, проведенной из её верхнего левого угла, а главная диагональ и всё что ниже её состоит только из нулевых элементов. Интересно отметить, что треугольник Паскаля, обладающий рядом замечательных свойств (последовательности треугольных, тетраэдрических чисел, биномиальных коэффициентов, чисел Фибоначчи и т.д.) фактически является таблицей умножения ненулевых элементов мнимого базиса гипер-дуальных чисел любого порядка. Например, для 3-го порядка это будут треугольник 2, 3; 3; для 4-го порядка: треугольник 2, 3, 4; 3, 6; 4; для 5-го порядка: 2, 3, 4, 5; 3, 6, 10; 4, 10; 5; и т. д.

$\times$	ε	ω	γ	η
ε	2ω	3γ	4η	0
ω	3γ	6η	0	0
γ	4η	0	0	0
η	0	0	0	0



Связь таблицы умножения элементов базиса мнимых частей с треугольником Паскаля

Алгебраические операции над гипер-дуальными числами 4-го порядка представляются формулами:

$$\begin{aligned}
 X &= x + x_1\varepsilon + x_2\omega + x_3\gamma + x_4\eta, & Y &= y + y_1\varepsilon + y_2\omega + y_3\gamma + y_4\eta, \\
 X + Y &= x + y + (x_1 + y_1)\varepsilon + (x_2 + y_2)\omega + (x_3 + y_3)\gamma + (x_4 + y_4)\eta, \\
 X \cdot Y &= x \cdot y + (x \cdot y_1 + y \cdot x_1)\varepsilon + (x \cdot y_2 + 2x_1 \cdot y_1 + y \cdot x_2)\omega + (x \cdot y_3 + y \cdot x_3 + 3(x_1 \cdot y_2 +
 \end{aligned}
 \tag{2}$$

$$+ y_1 \cdot x_2))\boldsymbol{\gamma} + (xy_4 + x_4y + 6x_2y_2 + 4(x_1y_3 + x_3y_1))\boldsymbol{\eta},$$

$$X/Y = X \cdot Y^{-1},$$

$$Y^{-1} = z + z_1\boldsymbol{\varepsilon} + z_2\boldsymbol{\omega} + z_3\boldsymbol{\gamma} + z_4\boldsymbol{\eta},$$

$$\text{где: } z = y^{-1}, \quad z_1 = -z^2y_1, \quad z_2 = -z(zy_2 + 2z_1y_1), \quad z_3 = -z(zy_3 + 3(z_1y_2 + \\ + z_2y_1)), \quad z_4 = -z(zy_4 + 6z_2y_2 + 4(z_1y_3 + z_3y_1))$$

Для определения функции от гипер-дуального аргумента 4-го порядка, разложим её в ряд Тейлора:

$$F(x + \Delta) = f(x) + \Delta \cdot f'(x) + \Delta^2 \cdot f''(x)/2 + \Delta^3 \cdot f'''(x) / 6 + \Delta^4 \cdot f^{IV}(x) / 24,$$

$$\text{где: } \Delta = x_1\boldsymbol{\varepsilon} + x_2\boldsymbol{\omega} + x_3\boldsymbol{\gamma} + x_4\boldsymbol{\eta}, \quad \Delta^2 = 2x_1^2\boldsymbol{\omega} + 6x_1x_2\boldsymbol{\gamma} + (8x_1x_3 + 6x_2^2)\boldsymbol{\eta}, \quad (3)$$

$$\Delta^3 = 6x_1^3\boldsymbol{\gamma} + 36x_1^2x_2\boldsymbol{\eta}, \quad \Delta^4 = 24x_1^4\boldsymbol{\eta}$$

Постановку Δ , Δ^2 , Δ^3 и Δ^4 в (3) даёт:

$$F(X) = f(x) + x_1f'(x)\boldsymbol{\varepsilon} + [x_2f'(x) + x_1^2f''(x)]\boldsymbol{\omega} + \\ + [x_3f'(x) + 3x_1x_2f''(x) + x_1^3f'''(x)]\boldsymbol{\gamma} + \\ + [x_4f'(x) + (4x_1x_3 + 3x_2^2)f''(x) + 6x_1^2x_2f'''(x) + x_1^4f^{IV}(x)]\boldsymbol{\eta} \quad (4)$$

При $x_1 = 1$ и $x_2 = x_3 = x_4 = 0$ соотношение (4) принимает форму:

$$F(x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma} + 0\boldsymbol{\eta}) = f(x) + f'\boldsymbol{\varepsilon} + f''(x)\boldsymbol{\omega} + f'''(x)\boldsymbol{\gamma} + f^{IV}(x)\boldsymbol{\eta},$$

$$\forall f(x) = F(x + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma} + 0\boldsymbol{\eta}),$$

Тогда блок-схема, реализующая (1) при $X_i = x_i + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma} + 0\boldsymbol{\eta}$, имеет вид:

$$x_{i+1} = x_i + \Phi(F(X_i)), \quad (5)$$

где $\Phi(F(X_i)) = D_i[1 + (L_i + M_i)/2 - K_i/6, + Q_i]$, вычисленное при $f(x_i) = F(X_i).Re$,

$$f'(x_i) = F(X_i).Im1, \quad f''(x_i) = F(X_i).Im2, \quad f'''(x_i) = F(X_i).Im3, \quad f^{IV}(x_i) = F(X_i).Im4$$

Для поддержки гипер-дуальных чисел 4-го порядка на языке C# была разработана статическая библиотека `Sdn4Lib.dll`, которая включает переопределение операций $+$, $-$, $\times$, $\div$ и основных базовых функций `pow(...)`, `powX(...)`, `exp(...)`, `ln(...)`, `log(...)`, `sqrt(...)`, `sin(...)`, `cos(...)`, `sh(...)`, `ch(...)`, возвращающих гипер-дуальные числа 4-го порядка. Размер библиотеки `Sdn4Lib.dll` составляет всего 7 kb. В Приложении 1 дан

код на языке C# в виде процедуры CH5(...), осуществляющий итерационную процедуру (5). Там же представлены процедуры, описывающие некоторые тестируемые функции.

Не менее важным является критерий окончания итерационного процесса. К сожалению, нет универсального рецепта в этом вопросе. Обычно в качестве этого критерия используется одно из следующих неравенств: $\|f(x_i)\| \leq \delta$, $\|x_i - x_{i-1}\| \leq \delta$ или $i \geq Imax$ (δ – заданная малая величина, $Imax$ – максимально допустимое количество итераций).

Скорость сходимости того или иного итерационного метода, как правило, определяется его порядком. Однако для решения конкретной прикладной задачи в первую очередь важно время нужное для её решения, а не количество итераций необходимое для его достижения. Поэтому в качестве критерия скорости сходимости будем также учитывать и время затраченное на решение.

Численный эксперимент проводился на компьютере с операционной системой Microsoft Windows 10 Pro, CPU 3.40 GHz, RAM 16 MB в среде MS VS 2021 при этом использовались тестируемые функции представленные в следующей таблице.

	Тестируемые функции	x_0
1	$f_1(x) = (\sin(x) - x/2)^2$	2.0
2	$f_2(x) = x^6 - 6x^5 + 50x^3 - 45x^2 - 108x + 108$	4.0
3	$f_3(x) = (x \cdot \exp(x^2) - \sin^2(x) + 3 \cos(x) + 5)^4$	-0.5
4	$f_4(x) = \ln^2(x) \cdot (\exp(x - 3) - 1) \cdot \sin(\pi x/3)$	4.0
5	$f_5(x) = x^3 - 6x^2 + 11x - 6$	4.0
6	$f_6(x) = x^5$	1.0
7	$f_7(x) = \sin(\cos(\operatorname{tg}(\operatorname{sh}(\operatorname{ch}(\operatorname{th}(x))))))$	1.7

Таблица 1. Список тестируемых функций

Результаты численного эксперимента (критерий окончания $\|(x_i) - (x_{i-1})\| \leq 1 \times 10^{-14}$) приведены в таблице 2.

Функция	x_0	x^*	$f(x^*)$	i	$t_{ms} \times 10^2$
1	2.0	1.895494267033980	0.0000000000000000	23	2.51
2	4.0	3.000011367610030	0.0000000000000000	15	2.00
3	-0.5	-1.207647827130910	0.0000000000000000	49	14.13

4	4.0	3.000000000000010	0.000000000000000	52	8.62
5	4.0	3.000000000000000	0.000000000000007	4	1.66
6	1.0	0.000000000000012	0.000000000000000	62	8.28
7	1.7	2.022988314672120	0.000000000000001	4	2.85

Таблица 2. Результаты расчета тестируемых функций (x^* – найденный корень, t_{ms} – миллисекунды)

В таблице 3 даны результаты теста при критерии окончания итерационного процесса $\|f(x_i)\| \leq 1 \times 10^{-14}$.

Функция	x_0	x^*	$f(x^*)$	i	$t_{ms} \times 10^2$
1	2.0	1.895494339229670	0.000000000000003	10	1.45
2	4.0	3.000011367610030	0.000000000000000	14	2.25
3	-0.5	-1.207635408737510	0.000000000000004	15	6.62
4	4.0	3.000189843353970	-0.000000000000001	13	2.83
5	4.0	3.000000000000000	-0.000000000000004	3	1.23
6	1.0	0.001352078038146	0.000000000000005	12	2.21
7	1.7	2.022988314672120	0.000000000000001	3	2.28

Таблица 3. Результаты расчета тестируемых функций (x^* – найденный корень, t_{ms} – миллисекунды)

Анализ полученных результатов тестируемых функций по критериям количества итераций и быстродействия t_{ms} показал, что:

1. Количество итераций не всегда связано с минимальным значением t_{ms} ;
2. Суммарное время вычисления всех тестируемых функций в одном пакете не превысило 0.4ms и 0.2ms для таблиц 2 и 3 соответственно;
3. При больших градиентах рывка $f'''(x)$ (больших скачках $f^{IV}(x)$) требуется более тщательной выбор начального приближения;

Использование гипер-дуальных чисел 4-го порядка ещё дает приятный бонус для оценки кратности найденного корня. Если следовать определению кратности корня данное в [6]: корень k -й кратности обращает в нуль не только функцию, но и производные до порядка $(k-1)$ включительно. Так как гипер-дуальная функция 4-го порядка возвращает значения как самой функции, так и её первых четырёх производных, то для тестируемых функций можно оценить кратность полученных корней: корни функций 2, 5 и 7 имеют кратность 1; корень функции 1 имеет кратность 2; корень функции 3 имеет кратность 3, корень

функции 4 имеет кратность 4, а кратность корня функции 6 равна 5. При этом целесообразно использовать критерий окончания итерационного процесса $\|x_i - x_{i-1}\| \leq \delta$.

В заключение отметим, что в некоторых случаях продуктивность метода Чебышева 5-го порядка может быть несколько улучшена, если только на его первом шаге положить $Q = 0$.

Приложение 1.

Исходный код на C#, использующий статическую библиотеку [Sdn4Lib.dll](#) структура которой дана в Приложении 2.

```
using System;
using System.Diagnostics;
using Sdn4Lib;

const double δ = 1e-10; //--1e-14;
const int maxIter = 1000;

struct tuple5
{
    public double x;
    public sdn4 y;
    public int i;
    public tuple5(double value1, sdn4 value2, int value3)
    {
        x = value1;
        y = value2;
        i = value3;
    }
}

//----- метод Чебышева 5-го порядка

tuple5 CH5(Func<double, sdn4> f, double x0)
{
    int i = -1; double D = 0, D4 = 0.0, L = 0, M = 0, K = 0, Q = 0,
    x = x0, x1 = 0;
    sdn4 F = f(x);
    double[] a = { 0.5, 1.0/6.0, 5.0/8.0, 5.0/12.0, 1.0};
    do
    {
        x1 = x;
```

```
        if (double.IsNaN(F.im1)) { return new tuple5(x1, F, i); }
        D = F.re / F.im1; L = D * F.im2 / F.im1;
        M = L * L; K = D * D * F.im3 / F.im1;
        D4 = a[4] * F.im4 / F.im1;
        Q = L * (a[2] * M - a[3] * K) + Math.Pow(D, 3) * D4;
        x -= D * (1.0 + a[0] * (L + M) - K * a[1] + Q);
        i += 1;
        if (double.IsNaN(x)){ x = x1; break;}
        F = f(x);
    } while (Math.Abs(x - x1) > δ && i < maxIter);
    return new tuple5(x, F, i);
}

//----- некоторые тестируемые функции

sdn4 x5 (double x)
{   var X = new sdn4(x, 1.0);
    return sdn4.pow(X,5);
}

sdn4 sinh (double x)
{   var X = new sdn4(x, 1.0);
    sdn4 th = sdn4.sh(X) / sdn4.ch(X);
    sdn4 ch = sdn4.ch(th);
    sdn4 sh = sdn4.sh(ch);
    sdn4 tg = sdn4.sin(sh) / sdn4.cos(sh);
    return sdn4.sin(sdn4.cos(tg));
}

//----- пример вызова метода CH5(...)

Stopwatch sw = Stopwatch.StartNew();
var R = CH5 (x5, 1);
sw.Stop();
var t = sw.Elapsed.TotalMilliseconds.ToString();
```

Приложение 2.

Структура библиотеки Sdn4Lib

```
namespace Sdn4Lib
{
    public struct sdn4
    {
        public double re;
        public double im1;
        public double im2;
        public double im3;
```

```
public double im4;
public sdn4(double re);
public sdn4(double re, double im1);
public sdn4(double re, double im1, double im2, double im3, double im4);
public static sdn4 ch(double x);
public static sdn4 ch(sdn4 X);
public static sdn4 cos(sdn4 X);
public static sdn4 cos(double x);
public static sdn4 exp(double x);
public static sdn4 exp(sdn4 X);
public static sdn4 Inverse(sdn4 A);
public static sdn4 ln(sdn4 X);
public static sdn4 ln(double x);
public static sdn4 log(sdn4 X, double a);
public static sdn4 log(double x, double a);
public static sdn4 pow(double x, double n);
public static sdn4 pow(sdn4 X, double n);
public static sdn4 powX(double a, double x);
public static sdn4 powX(sdn4 A, double x);
public static sdn4 sh(double x);
public static sdn4 sh(sdn4 X);
public static sdn4 sin(double x);
public static sdn4 sin(sdn4 X);
public static sdn4 sqrt(double x);
public static sdn4 sqrt(sdn4 X);
public static sdn4 operator +(sdn4 A);
public static sdn4 operator +(sdn4 A, sdn4 B);
public static sdn4 operator -(sdn4 A);
public static sdn4 operator -(sdn4 A, sdn4 B);
public static sdn4 operator *(double a, sdn4 B);
public static sdn4 operator *(sdn4 A, double b);
public static sdn4 operator *(sdn4 A, sdn4 B);
public static sdn4 operator /(sdn4 A, double b);
public static sdn4 operator /(sdn4 A, sdn4 B);
    }
}
```

ЛИТЕРАТУРА

1. Чебышев П. Л. Вычисление корней уравнений. – Собр. соч. Т.5, АН СССР, М. – Л., 1951, 7–25 с.
2. Антонюк П. О первой работе П. Л. Чебышева. – Научно-Исслед. Семинар по истории и методологии мат. и мех.: МГУ, 2014. –14 с.
3. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. – М.: Наука, 1979. – 285 с.

4. Naumann U. The Art of Differentiating Computer Programs. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2012.
5. Олифер В. И. Автоматическое дифференцирование на основе супер-дуальных чисел— URL:http://viosolutions.amerihomesrealty.com/pdf/Автоматическое_дифференцирование_на_основе_супер-дуальных_чисел.pdf, 2019.
6. Калиткин Н.Н., Кузьмина Л.В. Вычисление корней уравнения и определение их кратности. – математическое моделирование, 2010 год, том 22, номер 7, стр. 33-52.

Абстракт

В данной публикации производится реализация и анализ быстродействия итерационного метода Чебышева 5-го порядка для решения нелинейных уравнений. Используется автоматическое дифференцирование с гипер-дуальными числами 4-го порядка. Представлена компьютерная реализация этого подхода для языка C# операционной системы Windows 10 Pro. Проведены численные эксперименты.

This publication implements and analyzes the performance of the 5th-order Chebyshev iterative method for solving nonlinear equations. Automatic differentiation with 4th-order hyper-dual numbers is used. 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: *итерационные методы, автоматическое дифференцирование, гипер-дуальные числа, iterative methods, automatic differentiation, hyper-dual numbers.*

20 мая 2025 г