

РЕАЛИЗАЦИЯ МЕТОДА ЧЕБЫШЕВА 6-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 была представлена П. Л. Чебышевым в 1840 году [1, 2]. Она имеет вид:

$$\begin{aligned} x_{i+1} &= x_i - D_i - \frac{1}{2}(D_i)^2 \left[\frac{f''(x_i)}{f'(x_i)} \right] - (D_i)^3 \left[\frac{1}{2} \left(\frac{f''(x_i)}{f'(x_i)} \right)^2 - \frac{1}{6} \frac{f'''(x_i)}{f'(x_i)} \right] \\ &\quad - (D_i)^4 \left[\frac{5}{8} \left(\frac{f''(x_i)}{f'(x_i)} \right)^3 - \frac{5}{12} \frac{f''(x_i)f'''(x_i)}{f'(x_i)^2} + \frac{f^{IV}(x_i)}{24f'(x_i)} \right] = \\ &= x_i - D_i \left[1 + \frac{1}{2}(L_i + M_i) - \frac{1}{6}K_i + Q_i \right], \end{aligned} \quad (1)$$

где: $D_i = \frac{f(x_i)}{f'(x_i)}$, $L_i = D_i \frac{f''(x_i)}{f'(x_i)}$, $M_i = L_i^2$, $K_i = D_i^2 \frac{f'''(x_i)}{f'(x_i)}$,
 $Q_i = D_i^3 * \left[\frac{L_i}{D_i^3} \left(\frac{5}{8}M_i - \frac{5}{12}K_i \right) + \frac{f^{IV}(x_i)}{24f'(x_i)} \right]$, $\forall i = 0, 1, 2, \dots$

В 1870 году Е. Schröder в своей работе [3] фактически представил формулу Чебышева 6-го порядка:

$$x_{i+1} = x_i - D_i \left[1 + \frac{1}{2}(L_i + M_i) - \frac{1}{6}K_i + Q_i + P_i \right], \quad (2)$$

где:

$$\begin{aligned} P_i &= D_i^4 \left\{ \frac{7}{8} \left(\frac{f''(x_i)}{f'(x_i)} \right)^2 \left[\left(\frac{f''(x_i)}{f'(x_i)} \right)^2 - \frac{f'''(x_i)}{f'(x_i)} \right] + \frac{1}{12} \left(\frac{f'''(x_i)}{f'(x_i)} \right)^2 + \right. \\ &\quad \left. + \frac{f''(x_i)f^{IV}(x_i)}{8f'(x_i)^2} - \frac{1}{120} \frac{f^V(x_i)}{f'(x_i)} \right\} = \\ &= \frac{7}{8}M_i(M_i - K_i) + \frac{1}{12}K_i^2 + D_i^4 \left(\frac{f''(x_i)f^{IV}(x_i)}{8f'(x_i)^2} - \frac{f^V(x_i)}{120f'(x_i)} \right), \quad \forall i = 0, 1, 2, \dots \end{aligned}$$

Итерационная схема (2) описывает семейство методов Чебышева от 2-го до 6-го порядков (см. таблицу 1).

Порядок	D	L	M	K	Q	P
2	●	0	0	0	0	0
3	●	●	0	0	0	0
4	●	●	●	●	0	0
5	●	●	●	●	●	0
6	●	●	●	●	●	●

Таблица 1. Семейство методов Чебышева

Итерационные схемы (1) и (2) получены 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.

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, а иногда просто невозможно. В схеме (2) уже необходимо дополнительно вычислять ещё и 5-ю производную.

В работе [4] была рассмотрена реализация расчетной схемы (1) с использованием автоматического дифференцирования (AD) на базе гипер-дуальных чисел 4-го порядка [5]. Суть AD (при компьютерной реализации) заключается в создании нового типа данных и переопределении операций и базовых функций над ними.

Гипер-дуальное число 4-го порядка [5] имеет представление в виде $Z = z + z_1\epsilon + z_2\omega + z_3\gamma + z_4\eta$, $\forall z, z_1, z_2, z_3, z_4 \in \mathbb{R}$. 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 $\epsilon, \omega, \gamma, \eta$ образуют базис мнимых частей гипер-дуального числа, отвечающий следующими правилам: $\epsilon^2 = 2\omega$, $\epsilon\omega = 3\gamma$, $\epsilon\gamma = 4\eta$, $\omega^2 = 6\eta$, $\epsilon\eta = \omega\gamma = \omega\eta = \gamma\eta = \gamma^2 = \eta^2 = 0$.

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

$$\begin{aligned}
 X &= x + x_1\epsilon + x_2\omega + x_3\gamma + x_4\eta, & Y &= y + y_1\epsilon + y_2\omega + y_3\gamma + y_4\eta, \\
 X + Y &= x + y + (x_1 + y_1)\epsilon + (x_2 + y_2)\omega + (x_3 + y_3)\gamma + (x_4 + y_4)\eta, \\
 X \cdot Y &= x \cdot y + (x \cdot y_1 + y \cdot x_1)\epsilon + (x \cdot y_2 + 2x_1 \cdot y_1 + y \cdot x_2)\omega + (x \cdot y_3 + y \cdot x_3 + 3(x_1 \cdot y_2 + \\
 &\quad + y_1 \cdot x_2))\gamma + (x \cdot y_4 + y \cdot x_4 + 6x_2 \cdot y_2 + 4(x_1 \cdot y_3 + x_3 \cdot y_1))\eta, \\
 X/Y &= X \cdot Y^{-1}, \\
 Y^{-1} &= z + z_1\epsilon + z_2\omega + z_3\gamma + z_4\eta,
 \end{aligned} \tag{3}$$

где: $z = y^{-1}$, $z_1 = -z^2 y_1$, $z_2 = -z(zy_2 + 2z_1 y_1)$, $z_3 = -z(zy_3 + 3(z_1 y_2 + z_2 y_1))$, $z_4 = -z(zy_4 + 6z_2 y_2 + 4(z_1 y_3 + z_3 y_1))$

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

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

где: $\Delta = x_1 \varepsilon + x_2 \omega + x_3 \gamma + x_4 \eta$, $\Delta^2 = 2x_1^2 \omega + 6x_1 x_2 \gamma + (8x_1 x_3 + 6x_2^2) \eta$, $\Delta^3 = 6x_1^3 \gamma + 36x_1^2 x_2 \eta$, $\Delta^4 = 24x_1^4 \eta$ (4)

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

$$F(X) = f(x) + x_1 f'(x) \varepsilon + [x_2 f'(x) + x_1^2 f''(x)] \omega + [x_3 f'(x) + 3x_1 x_2 f''(x) + x_1^3 f'''(x)] \gamma + [x_4 f'(x) + (4x_1 x_3 + 3x_2^2) f''(x) + 6x_1^2 x_2 f'''(x) + x_1^4 f^{IV}(x)] \eta$$
 (5)

При $x_1 = 1$ и $x_2 = x_3 = x_4 = 0$ соотношение (5) принимает вид:

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

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

Тогда первые четыре производные при $X_i = x_i + \varepsilon + 0\omega + 0\gamma + 0\eta$, определяются по формулам:

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

Компьютерная реализация гипер-дуальных чисел 4-го порядка была выполнена на языке C# в виде статической библиотеки [Sdn4Lib.dll](#), которая включает переопределение операций $+$, $-$, $\times$, $\div$ 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 4-го порядка.

Для нахождения 5-й производной, входящей в (2), воспользуемся подходом, рассмотренном в работе [6]. Суть которого заключается в использовании в качестве целевой функции 4-ю производную $f^{IV}(x)$ исходной функции $f(x)$ с переопределением только базовых функций, что дает возможность вычисления производных с 5-го до 8-го порядков исходной функции. Для этого случая была разработана статическая библиотека [Sdn4Exdll](#), которая переопределяет только базовые функции. Заметим, что библиотеки [Sdn4Lib.dll](#) и [Sdn4Exdll](#) имеют размеры 7 и 3 kb соответственно.

Таким образом, для практической реализации схемы (2) необходимо описать исходную функцию, как в терминах `sdn4`, так и в терминах `sdn4x`. С целью сокращения письма программного кода были введены псевдонимы (alias) `T` и `S` для чисел `Sdn4` и `Sdn4x` соответственно. В этом случае тело исходной функции описывается в терминах `T`, а функция f^{IV} получается из описания f путем замены префиксов `T` на префиксы `S`. (см. Приложение 1).

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	$f_2(x) = (x \cdot \exp(x^2) - \sin^2(x) + 3 \cos(x) + 5)^4$	-1.0
3	$f_3(x) = \ln^2(x) \cdot (\exp(x - 3) - 1) \cdot \sin(\pi x/3)$	6.7
4	$f_4(x) = x^8$	1.0
5	$f_5(x) = \sin(\cos(\operatorname{tg}(\operatorname{sh}(\operatorname{ch}(\operatorname{th}(x)))))$	1.5
6	$f_6(x) = x^6 - 6x^5 + 50x^3 - 45x^2 - 108x$	4.2

Таблица 2. Список тестируемых функций

Цель численного эксперимента – выявить степень влияния f^V на конечный результат в схеме (2).

Результаты численного эксперимента (критерий окончания $\|(x_i) - (x_{i-1})\| \leq 1 \times 10^{-14}$) приведены в таблице 3 с учётом $f^V(x)$ (+) и без него (-). Расчет проводился по процедуре `СН6(...)`, реализующей схему (2) (см. Приложение 1).

Функция	$f^V(x)$	x_0	x^*	$f(x^*)$	i	t_{ms}
1	+	2.0	1.8954942670339821	7.8886090522101181E-31	22	0.0261
	-		1.8954942670339821	7.8886090522101181E-31	22	0.0257
2	+	-1.0	-1.2076478271309206	1.199857436841159E-27	24	0.0936
	-		-1.2076478271309206	1.199857436841159E-27	24	0.0182
3	+	6.70	6.0	-8.983421967191978E-15	4	0.0986
	-		6.0	-8.983421967191978E-15	4	0.0909
4	+	1.0	2.608777613530213E-14	2.145341689659686E-109	105	0.0301
	-		2.530629387641176E-14	1.682009238287699E-109	105	0.0215
5	+	1.5	2.0229883146721215	5.0531952754118148E-16	3	0.0301

	-		2.0229883146721210	-1.0489927069340377E-15	3	0.0215
6	+	4.2	4.0000000000000000	0.0	4	0.0217
	-		4.0000000000000009	0.0	4	0.02102

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

Анализ результатов расчета даёт возможность констатировать, что для тестируемых функций (по крайней мере) при $\delta = 1e-14$ влияние учёта $f^V(x)$ в схеме (2) практически не ощутимо, за исключения времени t_{ms} выполнения процедуры CN6(...).

Приложение 1.

Исходный код на C#, использующий статические библиотеки [Sdn4Lib.dll](#) и [Sdn4Exdll](#).

```
using System;
using System.Diagnostics;
using Sdn4Lib;
using T = Sdn4Lib.sdn4;
using S = Sdn4Ex.sdn4Ex;

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

struct tuple6
{
    public double x;
    public sdn4 y;
    public double y5;
    public int i;
    public tuple6(double value1, sdn4 value2, double value3, int value4)
    {
        x = value1;
        y = value2;
        y5 = value3;
        i = value4;
    }
}

struct tuple10
{
    public double f;
    public double f1;
    public double f2;
    public double f3;
    public double f4;
    public double f5;
    public double f6;
    public double f7;
    public double f8;
    public double x;
    public tuple10(double x, double f, double f1, double f2, double f3,
```

```
        double f4, double f5, double f6, double f7, double f8);
    }

//----- тестируемые функции

public class funcs
{ public funcs(){}

//-- 1) (sin(x) - x/2)^2
    public static T sin4(T X){ var sX = (T.sin(X) - 0.5 * X); return sX * sX;}
    public static T sin4x(T X){var sX = (S.sin(X) - 0.5 * X); return sX * sX;}

//-- 2) (x*exp(x^2) -sin(x)^2 + 3cos(x) + 5 )^2
    public static T exp(T X)
    {   var A = new sdn4 (5.0);
        var Z = X * T.exp(X * X) - T.sin(X)* T.sin(X) + 3.0 * T.cos(X) + A;
        return Z * Z ;
    }
    public static T expx(T X)
    {   var A = new sdn4 (5.0);
        var Z = X * S.exp(X * X) - S.sin(X)*S.sin(X) + 3.0 * S.cos(X) + A;
        return Z * Z;
    }

//-- 3) ln(x)^2*(exp(x-3) - 1)sin(π*x/3)
    public static T ln(T X)
    {   var A1 = new sdn4(1.0);
        var A2 = new sdn4(2.0);
        var A3 = new sdn4(3.0);
        var sin = T.sin((Math.PI / 3.0) * X);
        var ln = T.ln(X - A2);
        return ln * ln * (T.exp(X - A3) - A1) * sin;
    }
    public static T lnX(T X)
    {   var A1 = new sdn4(1.0);
        var A2 = new sdn4(2.0);
        var A3 = new sdn4(3.0);
        var sin = S.sin((Math.PI / 3.0) * X);
        var ln = S.ln(X - A2);
        return ln * ln * (S.exp(X - A3) - A1) * sin;
    }

//-- 4) x^8
    public static T x8(T X){return T.pow(X, 8);}
    public static T x8x(sdn4 X){return S.pow(X, 8);}

//-- 5) sin2(cos(tg(sh(ch(th(x))))))
    public static T sinh4(T X)
    {   var th = T.sh(X) / T.ch(X);
        var ch = T.ch(th);
        var sh = T.sh(ch);
        var tg = T.sin(sh) / T.cos(sh);
```

```
        return T.sin(T.cos(tg));
    }
    public static T sinh4x(T X)
    {
        var th = S.sh(X) / S.ch(X);
        var ch = S.ch(th);
        var sh = S.sh(ch);
        var tg = S.sin(sh) / S.cos(sh);
        return S.sin(S.cos(tg));
    }

    //-- 6)  $x^6 - 6x^5 + 50x^3 - 45x^2 - 108x$ 
    public static sdn4 x6(sdn4 X)
    {
        var A = new sdn4(108.0);
        var X2 = T.pow(X, 2);
        var X3 = T.pow(X, 3);
        var X5 = T.pow(X, 5);
        var X6 = T.pow(X, 6);
        return X6 - 6.0 * X5 + 50.0 * X3 - 45.0 * X2 - A * X;
    }

    public static sdn4 x6x(sdn4 X)
    {
        var A = new sdn4(108.0);
        var X2 = S.pow(X, 2);
        var X3 = S.pow(X, 3);
        var X5 = S.pow(X, 5);
        var X6 = S.pow(X, 6);
        return X6 - 6.0 * X5 + 50.0 * X3 - 45.0 * X2 - A * X;
    }
}

/-- метод Чебышева 6-го порядка
tuple6 CH6(Func<T,T> f, Func<T,T> fx, double x0)
{
    int i = -1; double D = 0.0, L = 0.0, M = 0.0, K = 0.0,
    Q = 0.0, P = 0.0, F5 = 0.0, xi = x0, x1 = 0.0;
    double[] a = { 0.5, 1.0 / 6.0, 5.0 / 8.0, 5.0 / 12.0, 1.0/24.0,
        7.0 / 8.0, 1.0/12.0, 1.0/8.0, 1.0/120.0 };
    tuple10 R = new tuple10();
    while (true)
    {
        x1 = xi; i += 1;
        R = S.Derivatives(x: xi, f: f, fx: fx);
        D = R.f / R.f1; L = D * R.f2 / R.f1; M = L * L; K = D * D * R.f3 / R.f1;
        Q = L * (a[2] * M - a[3] * K) + a[4] * Math.Pow(D, 3) * R.f4;
        F5 = a[8] * R.f5 / R.f1;
        P = a[5] * M * (M - K) + a[6] * K * K +
            Math.Pow(D, 4) * (a[7] * R.f2 * R.f4 / (R.f1 * R.f1) - F5);
        xi -= D * (1.0 + a[0] * (L + M) - a[1] * K + Q + P);
        if (Math.Abs(xi - x1) <= delta || i > maxIter || double.IsNaN(xi)){break;}
    }
    R = S.Derivatives(x: xi, f: f, fx: fx);
    return new tuple6(xi, new T(R.f, R.f1, R.f2, R.f3, R.f4), R.f5, i);
}
```

```
}  
  
/-- пример вызова метода CH6(...)  
Stopwatch sw = Stopwatch.StartNew();  
var R6 = CH6(f: funcs.sinh4, fx: funcs.sinh4x, x0: 1.5);  
sw.Stop();  
var t = sw.Elapsed.TotalMilliseconds.ToString();
```

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. Schröder E. Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen. – Math. Ann. Vol. 2, pp 317-365, 1870.
4. Олифер В. И. Реализация метода Чебышева 5-го порядка на базе гипер-дуальных чисел 4-го порядка. – URL:<http://viosolutions.amerihomesrealty.com/pdf/...>, 2025.
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. Олифер В. И. К вычислению производных высокого порядка на основе гипер-дуальных чисел. – URL:http://viosolutions.amerihomesrealty.com/pdf/Вычислению_производных_высокого_порядка.pdf, 2025.

Абстракт

В данной публикации производится реализация и анализ быстродействия итерационного метода Чебышева 6-го порядка для решения нелинейных уравнений. Используется автоматическое дифференцирование с гипер-дуальными числами 4-го порядка с расширением. 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 6th-order Chebyshev iterative method for solving nonlinear equations. Automatic differentiation with 4th-order hyper-dual numbers with expansion 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 июля 2025 г