ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ МЕТОДА ЧЕБЫШЕВА 4-ГО ПОРЯДКА С ИСПОЛЬЗОВАНИЕМ СУПЕР-ДУАЛЬНЫХ ЧИСЕЛ 3-ГО КЛАССА

Олифер В. И.

Методы Чебышёва для решения уравнения $y = f(x)$ основаны на разложении по формуле Тейлора функции $x = \varphi(y)$, обратной к $f(x)$. Они могут иметь произвольно высокий порядок точности, определяемый количеством членов разложения для функции $\varphi(y)$ [1, 2, 3].

Из предположения, что вещественная функция $f(x)$ является гладкой и монотонной на интервале $[a, b]$, так что она взаимно однозначно отображает этот интервал в некоторый интервал $[\alpha, \beta]$, следует существование обратной к $f(x)$ функции $x = \varphi(y)$: $[\alpha, \beta] \to [a, b]$, которая имеет ту же гладкость, что и $f(x)$. Тогда разложение функции $\varphi(y)$ в окрестности $y = 0$ по формуле Тейлора имеет вид:

$$\varphi(0) \approx \sum_{k=0}^{p} (-1)^k \varphi^{(k)}(y) \frac{y^k}{k!}, \tag{1}$$

где $\varphi^{(k)} = \dfrac{d^k \varphi}{dy^k}$.

Заменяя в (1) $y \to f(x)$ получим итерационную формулу Чебышёва $(p+1) -$ го порядка:

$$x_{i+1} = \sum_{k=0}^{p} (-1)^k \varphi^{(k)}\big(f(x_i)\big) \frac{f(x_i)^k}{k!}, \qquad i = 0, 1, 2, \ldots \tag{2}$$

Для определения производных функции $\varphi^{(k)}(y)$ необходимо проследовательно дифференцировать тождество $x = \varphi(y)$, используя правило дифференцирования сложных функций:

$$\begin{aligned}
&\varphi^{(0)} = x_i,\\
&\varphi^{(1)}(f(x))f'(x_i) = 1,\\
&\varphi^{(2)}(f(x))f'(x_i)^2 + \varphi^{(1)}(f(x))f''(x_i) = 0,\\
&\varphi^{(3)}(f(x))f'(x_i)^3 + 3\varphi^{(2)}(f(x))f'(x_i)f''(x_i) + \varphi^{(1)}(f(x))f'''(x_i) = 0,\\
&\ldots
\end{aligned} \tag{3}$$

Эта система соотношений (3) начиная со второй строки имеет треугольный вид, позволяющий последовательно найти $\varphi^{(k)}$ при $k > 0$:

$$
\begin{aligned}
&\varphi^{(1)}(f(x)) = \frac{1}{f'(x_i)},\\
&\varphi^{(2)}(f(x)) = -\frac{f''(x_i)}{f'(x_i)^2},\\
&\varphi^{(3)}(f(x)) = -3\frac{f''(x_i)^2}{f'(x_i)^5} - \frac{f'''(x_i)}{f'(x_i)^4},\\
&\dots
\end{aligned}
\qquad (4)
$$

Для $p = 3$ имеем расчетную формулу Чебышева 4-го порядка [2]:

$$
\begin{aligned}
x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\left(\frac{f(x_i)}{f'(x_i)}\right)^2 \frac{f''(x_i)}{f'(x_i)} - \frac{1}{2}\left(\frac{f(x_i)}{f'(x_i)}\right)^3 \left[\left(\frac{f''(x_i)}{f'(x_i)}\right)^2 - \frac{1}{3}\frac{f'''(x_i)}{f'(x_i)}\right] = \\
&= x_i - [1 + (L_i + L_i^2)/2 - K_i/6]D_i, \quad i = 0, 1, 2, \dots ,
\end{aligned}
\qquad (5)
$$

где: $L_i = D_i \frac{f''(x_i)}{f'(x_i)}$, $K_i = D_i^2 \frac{f'''(x_i)}{f'(x_i)}$, $D_i = \frac{f(x_i)}{f'(x_i)}$

Если положить $L_i^2 = K_i = 0$, то получим формулу Чебышева с кубической сходимостью, а при $L_i = L_i^2 = K_i = 0$ формулу Ньютона с квадратичной сходимостью.
Первым требованием к реализации итерационной формулы (5) является условие $f'(x) \neq 0$ на интервале $[a, b]$. Второе требование связано с небходимостью наличия способа вычисления $f(x)$, $f'(x)$, $f''(x)$ и $f'''(x)$, когда $f(x)$ имеет сложное аналитическое выражение или представлена сложным программный кодом. В этом случае используются методы численного дифференцирования с применением различных моделей аппроксимации. Последнее приводит к появлению неустранимой вычислительной погрешности, связанной с погрешностью самой модели аппроксимации.

С появлением современных вычислительных устройств и высокоуровневых алгоритмических языков программирования, позволяющих работать с нестандартными типами данных, стал успешно применяться метод автоматического дифференцирования (АД) [4], суть которого заключается в создании нового типа данных и переопределение операций и базовых функций над ними. Этот подход позволяет вычислять точные (с машинной точностью) значения функции и ее производных. Для рассматриваемого итерационного уравнения (5) в качестве такого нового типа данных могут быть исползованы супер-дуальные числа [5].

Супер-дуальные числа n-го класса $X_{[n]}$ определяются следующей формулой

$$X_{[n]} = x + \sum_{k=1}^{n} x_k\, \boldsymbol{\varepsilon}_k, \tag{6}$$

где: x и x_k – вещественные числа, $\boldsymbol{\varepsilon}_k$ – абстрактные элементы мнимого базиса (мнимые символы).

Величина x называется главной частью $X_{[n]}$ (с базисом равным 1), а $\sum_{k=1}^{n} x_k\, \boldsymbol{\varepsilon}_k$ – его мнимой (инфинитезимальной) частью.

Алгебраические операции сложения и вычитания над супер-дуальными числами *n*-го класса определяются формулами:

$$X_{[n]} \pm Y_{[n]} = (x \pm y) + \sum_{k=1}^{n} (x_k \pm y_k)\, \boldsymbol{\varepsilon}_k, \tag{7}$$

Правило умножения супер-дуальных чисел *n*-го класса должно исходить из требования конечности разложения функции супер-дуального числа *n*-го класса в ряд Тейлора:

$$F\big(X_{[n]}\big) = F\big(x + Im_{[n]}\big) = f(x) + \sum_{k=1}^{n} \frac{Im_{[n]}^{k}}{n!} f^{(k)}(x)\,, \qquad \text{где: } Im_{[n]} = \sum_{k=1}^{n} x_k\, \boldsymbol{\varepsilon}_k \tag{8}$$

Этому требованию отвечает следующее правило умножения элементов мнимого базиса:

$$\boldsymbol{\varepsilon}_l \boldsymbol{\varepsilon}_k = \Lambda_{lk} \boldsymbol{\varepsilon}_k, \qquad \Lambda_{lk} = \begin{cases} \dbinom{l+k}{k} = \dfrac{(l+k)!}{l!\,k!}, & \text{при } l+k \le n \\ 0, & \text{при } l+k > n \end{cases} \tag{9}$$

Тогда операция умножения супер-дуальных чисел *n*-го класса представима следующей формулой:

$$X_{[n]} \cdot Y_{[n]} = x \cdot y + \sum_{k=1}^{n} \left\{ (x \cdot y_k + y \cdot x_k) + \sum_{l=1}^{n} \Lambda_{lk} \cdot y_l \cdot x_k \right\} \boldsymbol{\varepsilon}_k \tag{10}$$

Таблица умножения элементов мнимого базиса любого класса супер-дуальных чисел симметрична относительно диагонали, проведенной из её верхнего левого угла; главная диагональ и всё что ниже её состоит только из нулевых элементов.

Интересно отметить, что треугольник Паскаля [6], обладающий рядом замечательных свойств (последовательности треугольных, тетраэдрических чисел, биномиальные коэффициенты, числа Фибоначчи и т.д.) фактически является таблицей умножения ненулевых элементов мнимого базиса супер-дуальных чисел любого класса. Например (см.

Рис.1), для 3-го класса это будут треуголник 2, 3; 3; для 4-го класса: треугольник 2, 3, 4; 3, 6; 4; для 5-го класс: 2, 3, 4, 5; 3, 6, 10; 4, 10; 5; и т.д. Для большей визуальной наглядности достаточно поверную треугольник Паскаля на угол в 45 градусов против часовой стрелки.

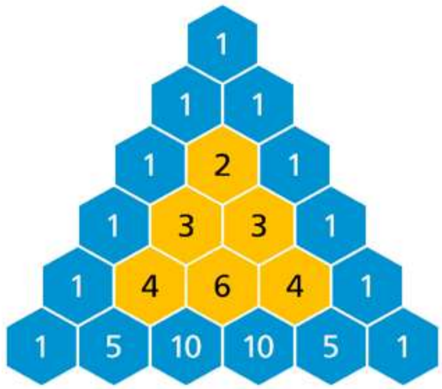

Рис. 1. Связь треугольника Паскаля с правилами умножения элементов мнимого базиса.

Так как для численной реализации формулы Чебышева 4-го порядка (5) необходимо вычисление только $f(x)$, $f'(x)$, $f''(x)$ и $f'''(x)$, то далее будем использовать супер-дуальные числа 3-го класса с базисом $\{1, \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_1, \boldsymbol{\omega} = \boldsymbol{\varepsilon}_2, \ \boldsymbol{\gamma} = \boldsymbol{\varepsilon}_3\}$ и нижний индекс в $X_{[3]}$ будем опускать. В этом случае правила умножения базисов мнимых частей $\boldsymbol{\varepsilon}, \boldsymbol{\omega}$ и $\boldsymbol{\gamma}$ ($\boldsymbol{\varepsilon}^2 = 2\boldsymbol{\omega}$, $\boldsymbol{\varepsilon\omega} = 3\boldsymbol{\gamma}$, $\boldsymbol{\varepsilon\gamma} = \boldsymbol{\omega\gamma} = \boldsymbol{\gamma}^2 = \boldsymbol{\omega}^2 = 0$) можно представить в виде таблице:

$\times$	$\boldsymbol{\varepsilon}$	$\boldsymbol{\omega}$	$\boldsymbol{\gamma}$
$\boldsymbol{\varepsilon}$	$2\boldsymbol{\omega}$	$3\boldsymbol{\gamma}$	0
$\boldsymbol{\omega}$	$3\boldsymbol{\gamma}$	0	0
$\boldsymbol{\gamma}$	0	0	0

Табл. 1. Умножение базисов мнимых частей для 3-го класса супер-дуальных чисел ($\boldsymbol{\varepsilon} = \ \boldsymbol{\varepsilon}_1, \boldsymbol{\omega} = \ \boldsymbol{\varepsilon}_2$ и $\boldsymbol{\gamma} = \ \boldsymbol{\varepsilon}_3$).

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

$$\begin{aligned}
&X = x + x_1\boldsymbol{\varepsilon} + x_2\boldsymbol{\omega} + x_3\boldsymbol{\gamma}, \qquad Y = y + y_1\boldsymbol{\varepsilon} + y_2\boldsymbol{\omega} + y_3\boldsymbol{\gamma} \\
&X + Y = x + y + (x_1 + y_1)\boldsymbol{\varepsilon} + (x_2 + y_2)\boldsymbol{\omega} + (x_3 + y_3)\boldsymbol{\gamma}, \\
&X{\cdot}Y = x{\cdot}y + (x{\cdot}y_1 + y{\cdot}x_1)\boldsymbol{\varepsilon} + (x{\cdot}y_2 + 2x_1{\cdot}y_1 + y{\cdot}x_2)\boldsymbol{\omega} + \\
&\qquad + (x{\cdot}y_3 + y{\cdot}x_3 + 3(x_1{\cdot}y_2 + y_1{\cdot}x_2))\boldsymbol{\gamma}, \\
&\mathrm{Y}^{-1} = y^{-1} - y^{-2}{\cdot}y_1\boldsymbol{\varepsilon} + y^{-2}(\,2{\cdot}y_1^2{\cdot}y^{-1} - y_2)\boldsymbol{\omega} + \\
&\qquad + a^{-2}(6y_1{\cdot}y^{-1}(y_2 - y_1^2{\cdot}y^{-1}) - y_3)\boldsymbol{\gamma}, \\
&X/Y = \mathrm{X}{\cdot}\mathrm{Y}^{-1},
\end{aligned} \tag{11}$$

а произвольная функция супер-дуального аргумента 3-го класса после ее разложения в ряд Тейлора принимает вид:

$$
\begin{aligned}
&F(X) = f(x) + x_1 f'(x)\boldsymbol{\varepsilon} + [x_2 f'(x) + x_1^2 f''(x)]\boldsymbol{\omega} + \\
&\qquad + [x_3 f'(x) + 3x_1 x_2 f''(x) + x_1^3 f'''(x)]\boldsymbol{\gamma}, \\
&F(x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}) = f(x) + f'(x)\boldsymbol{\varepsilon} + f''(x)\boldsymbol{\omega} + f'''(x)]\boldsymbol{\gamma}, \\
&\text{где } f(x) = F(x + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma})
\end{aligned} \tag{12}
$$

Для выделения главной и мнимых частей X вводятся соответствующие операции:

$$x = X.Re, \;\; x_1 = X.Im1, \;\; x_2 = X.Im2, \;\; x_3 = X.Im3 \tag{13}$$

Тогда

$$f\,(x) = F(\tilde{X}).Re, \; f'(x) = F(\tilde{X}).Im1, \; f''(x) = F(\tilde{X}).Im2, \; f'''(x) = F(\tilde{X}).Im3 \tag{14}$$

где: $\tilde{X} = x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}$ и компоненты формулы (5) вычисляются по формулам

$$L_i = D_i \frac{F(\tilde{X}_i).Im2}{F(\tilde{X}_i).Im1}, \;\; K_i = D_i^2 \frac{F(\tilde{X}_i).Im3}{F(\tilde{X}_i).Im1}, \;\; D_i = \frac{F(\tilde{X}_i).Re}{F(\tilde{X}_i).Im1} \tag{15}$$

Компьютерная реализация супер-дуальных чисел 3-го класса (Sdn3) была выполнена 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 выполненя численных экспериментов по формулам (5) и (15) в Приложение 1 приводится итерационная процедура Chebychev2_3_4(...), реализирующая метод Чебышева 2-го, 3-го и 4-го порядков. Там же представлены процедуры, описывающие тестируемые функции. В табл. 2 приведено сравнение числа итераций методов Чебышева разных порядков для десяти тестируемых функций. Условие прекращения итерационных процессов определялось выражением $|f(x_i)| \le \delta = 10^{-15}$

№	$f(x)$	x_o	Кол-во итеракций		
			порядок 2	порядок 3	порядок 4
1	$x^3 + 4x^2 - 10$	2.0	4	3	2
2	$\cos(x) - x$	1.0	3	2	2
3	$(x-1)^3 - 1$	2.5	5	3	2

4	$x^3 - \sin^2(x) + 3\cdot\cos(x) + 5$	-1.0	4	3	2
5	$\exp(-x) + \cos(x)$	2.0	3	2	2
6	$x^2 - \exp(x) - 3x + 2$	1.0	3	2	1
7	$\exp(x^2 + 7x - 30) - 1$	5.5	43	29	27
8	$\sin(x)$	0.5	3	2	2
9	$x\cdot\exp(x^2) - \sin^2(x) + 3\cdot\cos(x)$	-5.0	30	20	16
10	$\sin^2(x) - x^2 + 1$	2.0	4	3	2

Табл. 2. Сравнение скорости сходимости методов Чебышева 2-го, 3-го и 4-го порядков по числу итераций

Метод Чебышева 4-го порядка показал лучшие результаты чем 2-го и 3-го порядков, но для него желательна большая степень монотонности функции $f(x)$ на интервале $[a, b]$. Эти три метода требуют более тщательного определения начального приближения, впрочем, как и другие итерационные методы. В любом случае, целесообразно предварительное исследование исходной функции на интересующей области её определения. Наверно, самым простым способом такого анилиза является графический выполненый с помощью того либо иного графического калькулятора, например, Demos calculator.

Рассмотренный прием реализации семейства итеррационной формулы Чебышева (5) с использованием автоматического дифференцирования на базе супер-дуальных чисел 3-го класса показал свою несомненную актуальнольность, особенно для функций, являющимися весьма сложными композициями базовых функций.

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS 13.3). Тип данных Sdn3 (супер-дуальные числа 3-го класса) приведен в [5], 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 δ:Double = 1E-15;  // allowable error
// Order of method

enum Order:Int{
   case order2 = 2
   case order3
   case order4
}

// INPUT DATA:
// f – function pointer, xo – start point, order – order of method
// OUTPUT DATA:
// (x, i) – solution and number of iterations

func Chebyshev2_3_4(f:(Double) -> Sdn3, x0:Double, order:Order)->(x:Double,i:Int){
   var i = -1,    x = x0,   F = f(x),   D = 0.0,   L = 0.0,   K = 0.0;
   repeat{
      D = F.re/F.im1;     L = D*F.im2/F.im1;
      switch order {
         case Order.order2:   x -= D;
         case Order.order3:   x -= (1.0 + 0.5*L)*D;
         case Order.order4:   K = D*D*F.im3/F.im1;
               x -= (1.0 + 0.5*(L + L*L) - K/6.0)*D;
      }
      i += 1;    F = f(x);
   } while  (abs(F.re) > δ);
   return (x, i);
}

// call function Chebyshev2_3_4()

let x = Chebyshev2_3_4(f: f10, x0: 2.0, order: Order.order4);
print("x=\(x.x), i=\(x.i) ");

//  functions under study

func f1(x:Double)->Sdn3{
   let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
   return X**3 + 4*X*X - Sdn3(re: 10.0);
}

func f2(x:Double)->Sdn3{
   let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
   return Sdn3.cos(X: X) - X;
}

func f3(x:Double)->Sdn3{
```

```
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0) - Sdn3(re: 1.0);
    return X**3  - Sdn3(re: 1.0);
}

func f4(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    return X**3 - Sdn3.sin(X: X)**2 + 3*Sdn3.cos(X: X) + Sdn3(re: 5.0);
}

func f5(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    return Sdn3.exp(X: -X) + Sdn3.cos(X: X);
}

func f6(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    return X**2 - Sdn3.exp(X: X) - 3*X + Sdn3(re: 2.0);
}

func f7(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    let Z = X**2 + 7.0*X -  Sdn3(re: 30.0);
    return Sdn3.exp(X: Z) -  Sdn3(re: 1.0);
}

func f8(x:Double)->Sdn3{
    return Sdn3.sin(X: Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0));
}

func f9(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    return X*Sdn3.exp(X: X**2) - Sdn3.sin(X: X)**2 + 3.0*Sdn3.cos(X: X) ;
}
func f10(x:Double)->Sdn3{
    let X = Sdn3(re: x, im1: 1.0, im2: 0.0, im3: 0.0);
    return Sdn3.sin(X: X)**2 - X**2 + Sdn3(re: 1.0);
}
```

ЛИТЕРАТУРА

1. *Шарый С. П.* Курс вычислительных методов. – Новосибирск: Новосиб. гос. ун-т., 2015. – 507 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. *Березин И. С., Жидков Н. П.* Методы вычислений. Т.2 – Изд. Физико-математической литературы. – М: 1958. – 620 с.
4. *Naumann U.* The art of differentiating computer programs. Society for industrial and applied mathematics, Philadelphia, 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 , 2020.
6. *Успенский В. А.* Треугольник Паскаля. – М.: Наука, 1979. – 48 с.

Абстракт

В данной публикации рассматривается метод реализации итерационных формул Чебышева на основе 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 нелинейных уравнений, автоматическое дифференцирование, супер-дуальные числа, iterative methods for solving nonlinear equations, automatic differentiation, super-dual numbers.*

20 сентября 2023 г.