РЕАЛИЗАЦИЯ МЕТОДОВ ЧЕБЫШЕВА-ХЭЛЛИ 5-ГО ПОРЯДКА СХОДИМОСТИ НА ОСНОВЕ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Олифер В. И.

Одной из важнейших задач численного анализа является решение нелинейных уравнений $f(x) = 0$. Для решения этих уравнений часто используются различные итерационные методы. Грубо говоря, итерационный метод начинается с начального предположения x_0 (называемого порной точкой), которое улучшается посредством итерации $x_{i+1} = \varphi(x_i)$, $i = 0,1,2,\dots$.

В общем случае, итерационный метод $x_{i+1} = \varphi(x_i)$ имеет порядок q, если ошибка $|x_* - x_{i+1}|$ пропорционалена $|x_* - x_i|^q$ при $i \to \infty$, где x_* - решение $f(x) = 0$.

Следовательно, чем выше порядок, тем выше скорость сходимости. Однако эксплуатационные затраты метода также возрастают с его порядком. Этот факт приводит к поиску баланса между высокой скоростью и эксплуатационной стоимостью.

Основой итерационных методов (типа метода Ньютона) служит разложение исходной функции $f(x)$ в ряд Тейлора и приравниванием его нулю:

$$f(x_i) + \sum_{j}^{J} \frac{1}{j!} f^{(j)}(x_i) \left[\sum_{k=0}^{j} \binom{j}{k} x_{i+1}^{j-k}\, x_i^k \right] = 0 \qquad (1)$$

Левая часть уравнения (1) является алгебраическим многочленом J-й степени относительно x_{i+1}. При $J = 1$ получим формулу метода Ньютона. Если $J > 1$ возникает проблема с выбором корня. Более того, возможно получение комплексных корней.

В работе [1] описан вариант двухшагового итерационного метода на основе формулы Ньютона, но с гипер-касательной, которая описывалась многочленом степени больше 1. На первом шаге использовался упращенный вариан получения $x_{i+1} = \varphi_1(x_i)$, например, метод Ньютона с линеной касательной. Второй шаг уточнял x_{i+1} с применением параболической касательной. Аналогичная двухшаговая итерационная процедура была применена для получения метода Чебышева-Хэлли 5-го порядка сходимости [2]:

■ Первый шаг:

$$x_{i+1} = x_i - \left[1 + \frac{1}{2} \cdot \frac{L(x_i)}{1 - \alpha L(x_i)}\right] D(x_i), \qquad \alpha \in R, \qquad i = 0, 1, 2, \dots \qquad (2)$$

где: $L(x_i) = \frac{f(x_i)f''(x_i)}{f'(x_i)^2} = D(x_i)\cdot f''(x_i)/f'(x_i)$, а $D(x_i) = f(x_i)/f'(x_i)$ – ньютоновская корректировка.

■ Второй шаг:

$$\tilde{x}_{i+1} = x_{i+1} - \left[1 + \frac{M(x_i, x_{i+1})}{1 - \beta M(x_i, x_{i+1})}\right] D(x_i, x_{i+1}), \qquad \tilde{x}_{i+1} \Rightarrow x_{i+1}, \tag{3}$$

где: $M(x_i, x_{i+1}) = L(x_i)(1 - f(x_{i+1})/f(x_i))$, $D(x_i, x_{i+1}) = f(x_{i+1})/f'(x_i)$, $\beta \in R$

Формула (2) первого шага суть – семейство методов Чебышева-Хэлли 3-го порядка сходимости, а формула (3) второго шага была получена на основе классического метода Ньютона с гипер-касательной в виде гиперболы. Величины α и β являются параметрами 1-го и 2-го шагов. При $\alpha = 0\,, 0.5,\ 1$ имеем формулы: метода Чебышева, метода Хэлли и супер-метода Хэлли соответственно. Параметр β входит в формулу, описывающую гипер-касательную (гиперболу).

В [2] показано, что (2) и (3) реализуют метода Чебышева-Хэлли 5-го порядка сходимости.

Определение $L(x_i)$, $D(x_i)$ и $D(x_i, x_{i+1})$, входящих в соотношения (2) и (3), связано с небходимостью вычисления $f(x_i)$, $f'(x_i)$ и $f''(x_i)$. В ряде случаев вычисление $f'(x_i)$ и особенно $f''(x_i)$ весьма ресурсозатратно, а иногда и невыполнимо.

Проблема точности компьютерного вычисления производных функции может быть успешно решена путем использования метода автоматического дифференцирования (АД), суть которого заключается в создании нового типа данных и переопределения операций над ними. Этот подход позволяет определять точные (с машинной точностью) значения функции и ее производных. Для рассматриваемого случая в качестве такого нового типа данных могут быть исползованы усеченные гипер-дуальные числа [3].

Согласно [3], усеченное гипер-дуальное число (truncated hyper-dual number) определяется выражением $X = x + x_1\boldsymbol{\varepsilon} + x_2\boldsymbol{\omega}$, где x, x_1 и x_2 – вещественные числа, $\boldsymbol{\varepsilon}$ и $\boldsymbol{\omega}$ – мнимые символы. Пространство усеченных гипер-дуальных чисел отвечает трехмерной алгебре с правилом умножения элементов базиса $\{\mathbf{1},\ \boldsymbol{\varepsilon},\ \boldsymbol{\omega}\}$:

$\times$	$\mathbf{1}$	$\boldsymbol{\varepsilon}$	$\boldsymbol{\omega}$
$\mathbf{1}$	1	$\boldsymbol{\varepsilon}$	$\boldsymbol{\omega}$
$\boldsymbol{\varepsilon}$	$\boldsymbol{\varepsilon}$	$2\boldsymbol{\omega}$	0
$\boldsymbol{\omega}$	$\boldsymbol{\omega}$	0	0

(4)

Таблица 1. Правила умножения элементов базиса усечённых гипер-дуальных чисел

Число $x = Re(X) = X.Re$ называется главной частью X, а $x_1 = Im_1(X) = X.Im1$ и $x_2 = Im_2(X) = X.Im2$ – мнимыми частями X.
Алгебраические операции сложения, умножения, обращения и деления (с учетом табл. 1) определены по правилам:

$$\begin{aligned}
&A = a + a_1\boldsymbol{\varepsilon} + a_2\boldsymbol{\omega}, \qquad B = b + b_1\boldsymbol{\varepsilon} + b_2\boldsymbol{\omega}\\
&A + B = a + b + (a_1 + b_1)\boldsymbol{\varepsilon} + (a_2 + b_2)\boldsymbol{\omega},\\
&A{\cdot}B = a{\cdot}b + (a{\cdot}b_1 + b{\cdot}a_1)\boldsymbol{\varepsilon} + (a{\cdot}b_2 + 2{\cdot}a_1{\cdot}b_1 + b{\cdot}a_2)\boldsymbol{\omega},\\
&A^{-1} = a^{-1} - a_1{\cdot}a^{-2}\boldsymbol{\varepsilon} + (2{\cdot}a_1^2{\cdot}a^{-3} - a_2{\cdot}a^{-2})\boldsymbol{\omega},\\
&{}^{\mathrm{A}}\!/_{B} = \mathrm{A}{\cdot}\mathrm{B}^{-1} = a{\cdot}b^{-1} + (a_1{\cdot}b^{-1} - a{\cdot}b_1{\cdot}b^{-2})\boldsymbol{\varepsilon} + [2{\cdot}(a{\cdot}b_1^2{\cdot}b^{-3} - a_1{\cdot}b_1{\cdot}b^{-2}) -\\
&\qquad\qquad a{\cdot}b_2{\cdot}b^{-2} + a_2{\cdot}b^{-1}]\boldsymbol{\omega}
\end{aligned} \tag{5}$$

Функция усеченного гипер-дуального аргумента реализуется выражением

$$F(X) = f(x) + x_1{\cdot}f'(x)\boldsymbol{\varepsilon} + (x_2{\cdot}f'(x) + x_1^2{\cdot}f''(x))\boldsymbol{\omega} \tag{6}$$

При $x_1 = 1$ и $x_2 = 0$ формула (6) принимает вид:

$$F(X) = f(x) + f'(x)\boldsymbol{\varepsilon} + f''(x)\boldsymbol{\omega} \tag{7}$$

Описание элементарных (базовых) функций усечённого гипер-дуального аргумента приведены в [3]. Например, $\ln(X) = \ln(x) + x^{-1}\boldsymbol{\varepsilon} - x^{-2}\boldsymbol{\omega}$, откуда $Re(X) = f(x) = \ln(x)$, $Im_1(X) = f'(x) = x^{-1}$ и $Im_2(X) = f''(x) = -x^{-2}$.
Вычисление сложной усеченной гипер-дуальной функции (function composition) вида $F = f_1(f_2(\dots f_k(X)\dots), X), X)$ (где значение f_k используется в качестве аргумента для f_{k-1}) необходимо начать с вычисления $F_k = f_k(X)$, продолжить вычислениями $F_{k-1} = f_{k-1}(F_k, X)$, $F_{k-2} = f_{k-2}(F_{k-1}, X)$, … , $F = f_1(F_2, X)$.
Для применения АД (на базе усеченных гипер-дуальных чисел и усеченных гипер-дуальных функций) к соотношениям (2) и (3) необходимо применить следующие отображения

$$x \;\rightarrow\; X = x + 1{\cdot}\boldsymbol{\varepsilon} + 0{\cdot}\boldsymbol{\omega}, \qquad f(x) \rightarrow F(X) = f(x) + f'(x){\cdot}\boldsymbol{\varepsilon} + f''(x){\cdot}\boldsymbol{\omega}$$

Тогда для (2), (3) имеем

$$\begin{aligned}
&D(x_i) = F(X_i).Re/\,F(X_i).Im_1, \quad D(x_i, x_{i+1}) = F(X_{i+1}).Re/F(X_i).Im_1,\\
&L(x_i) = D(x_i)[F(X_i).Im_1/F(X_i).Im_2],
\end{aligned} \tag{8}$$

где: $X_i = x_i + 1{\cdot}\boldsymbol{\varepsilon} + 0{\cdot}\boldsymbol{\omega}$.

Компьютерная реализация усеченное гипер-дуальных чисел (Thdn) была выполнена на языке программирования SWIFT 5 для macOS 13.3 в виде статической библиотеки (static library) ThdnLibraryX, которую можно скачать с интернет ресурса (см. Приложение 1), а затем добавить в свой Xcode проект. Для выполненя численных экспериментов по формулам (2), (3) и (8) в Приложение 1 приводится итерационная процедура ICH(...), реализирующая семейство методов Чебышева-Хэлли 5-го порядка сходимости. Там же представлены процедуры, описывающие тестируемые функции.
Для двенадцати тестируемых функций (табл. 2) в табл. 3 дано сравнение числа итераций при разных значениях α и β. Условие прекращения итерационных процессов определялось выражением $|f(x_i)| \le \delta = 10^{-14}$.

$f(x)$	x_0	x_*
$f_1(x) = x^3 + 4x^2 - 10$	0.3	1.3652300134140969
$f_2(x) = \cos(x) - x$	0.0	0.7390851332151607
$f_3(x) = x^3 - 10$	1.7	2.1544346900318841
$f_4(x) = x^2 - e^x - 3x + 2$	0.0	0.2575302854398608
$f_5(x) = \sin^2(x) - x^2 + 1$	1.2	1.4044916482153411
$f_6(x) = x^2 + \sin\left(\frac{x}{5}\right) - \frac{1}{4}$	0.1	0.4099920179891371
$f_7(x) = e^x - 4x^2$	1.0	0.7148059123627778
$f_8(x) = e^{-x} + \cos(x)$	1.5	1.7461395304080070
$f_9(x) = e^{x^2+7x-30} - 1$	3.1	3.0
$f_{10}(x) = (x-1)^3 - 1$	1.5	2.0
$f_{11}(x) = e^x \sin(x) + \log(x^2+1)$	1.0	0
$f_{12}(x) = (x-2)(x^{10}+x+1)e^{-(x+1)}$	2.5	2.0

Таблица 2. Тестируемые функции.

$f(x)$	$\alpha = 0$			$\alpha = 1/2$			$\alpha = 1$		
	$\beta = 0$	$\beta = 3/4$	$\beta = 1$	$\beta = 0$	$\beta = 3/4$	$\beta = 1$	$\beta = 0$	$\beta = 3/4$	$\beta = 1$
f_1	25	20	17	21	4	3	8	4	4
f_2	3	3	3	3	3	3	3	3	3

f_3	3	3	3	3	3	3	3	3	3
f_4	2	2	2	2	2	2	2	2	2
f_5	3	3	3	3	3	3	3	3	3
f_6	4	4	4	4	3	3	4	3	3
f_7	3	3	3	3	3	3	3	3	3
f_8	3	2	2	2	2	2	2	2	2
f_9	3	3	3	3	3	3	4	3	3
f_{10}	8	20	14	32	3	3	5	4	3
f_{11}	4	4	4	4	4	4	4	3	3
f_{12}	4	4	4	4	4	4	4	4	4

Таблица 3. Сравнение количества итераций ***n*** *в зависимости от величины* α *и* β.

В табл. 3 видно, что столбцы ($\alpha = 1/2,\ \beta = 1$) и ($\alpha = 1,\ \beta = 1$) имеют наименьшее количество итераций. Из чего следует, что на первом шаге целесообразно использовать метод Хэлли или супер-метод Хэлли.

Для некоторых тестируемых функций результаты, приведенные в табл. 3, оказались несколько хуже (на одну итерацию больше), чем дает одношаговый метода Чебышева 4-го порядка, учитывающий третьи производные и использующий гипер-дуальные числа более высокого порядка [4].

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS 13.3). Тип данных Thdn (truncated hyper-dual number) приведен в [3], или можно скачать статическую библиотеку ThdnLibraryX.zip и после распаковки добавить ThdnLibraryX в свой проект (как это сделать см. ReadMe.txt). Download ThdnLibraryX.zip

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

let δ:Double = 1E-14;

// INPUT DATA:
// f – function pointer; xo – start point; α, β – parameters of the 1st and 2nd steps
```

```
// OUTPUT DATA:
// (x, i) – solution and number of iterations

func ICH(f:(Double) -> Thdn, x0:Double, α:Double, β:Double ) -> (x:Double, i:Int){
   var xi = x0,  F  = Thdn(), Φ = Thdn();
   var L, D, M: Double, i = 0;
   while (true){
  //-----STEP 1:
         i += 1;
         F = f(xi);      D = F.re/F.im1;      L = D*F.im2/F.im1;
         xi -= (1.0 + 0.5*L/(1.0 - α*L))*D;
  //-----STEP 2:
         Φ = f(xi);
         if abs(Φ.re) <= δ {break}
         M = L*(1.0 - Φ.re/F.re);
         xi -= (1.0 + M/(1.0 - β*M))*Φ.re/F.im1;
   }
   return (xi, i);
}

//----- ICH () function call example

var r = ICH(f: func12, x0: 2.5, α: 1.0, β: 0.0);
print("x=\(r.x), i=\(r.i)”);


//----- some functions under study

func func9(x:Double)->Thdn{
   let X = Thdn(re: x, im1: 1, im2: 0);
   let Z = X*X + 7.0*X - Thdn(re: 30.0);
   return  Thdn.exp(X: Z) - Thdn(re: 1.0);
}

func func11(x:Double)->Thdn{
   let X = Thdn(re: x, im1: 1, im2: 0);
   let Z = Thdn.exp(X: X)*Thdn.sin(X: X);
   return  Z + Thdn.log(X: X*X + Thdn(re: 1.0));
}

func func12(x:Double)->Thdn{
   let X = Thdn(re: x, im1: 1, im2: 0);
   let Z = (X - Thdn(re: 2.0))*(Thdn.pow(X: X, n: 10) + X +  Thdn(re: 1.0));
   return  Z*Thdn.exp(X: Thdn(re: -1.0) - X);
}
```

ЛИТЕРАТУРА

1. *H.H.H. Homeier*. On Newton-type methods with cubic convergence, J. Comput. Appl. Math. 176 (2005) 425–432

2. *Jisheng Kou, Yitian Li*. The improvements of Chebyshev–Halley methods with fifth-order convergence. Applied Mathematics and Computation 188 (2007) 143–147

3. *Олифер В. И.* Усеченные гипер-дуальные числа в автоматическом дифференцировании. – URL: http://viosolutions.amerihomesrealty.com/pdf/ Усеченные_гипер-дуальные_числа_в_автоматическом_дифференцировании.pdf, 2020.

4. *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.pdf, 2023.

Абстракт

В данной публикации рассматривается метод реализации итерационных формул Чебышева-Хэлли пятого порядка сходимости на основе автоматического дифференцирования с использованием усеченных гипер-дуальных чисел. Представлена компьютерная реализация этого подхода для языка SWIFT операционной системы macOS. Проведены численные эксперименты.

Ключевые слова*: итерационные методы решения нелинейных уравнений, автоматическое дифференцирование, гипер-дуальные числа, iterative methods for solving nonlinear equations, automatic differentiation, hyper-dual numbers.*

20 октября 2023 г