К ЧИСЛЕННОМУ РЕШЕНИЮ НЕЯВНЫХ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Олифер В. И.

Если каждому значению $x \in X^{\pm}$ соответствует значение $y \in Y^{\pm}$, которое вместе с x удовлетворяет уравнению $F(x, y) = 0$, то говорят, что это уравнение определяет на множестве $X^{\pm}$ *неявную функцию* (*implicit function)* одной переменной $y = y(x)$ [1]. Таким образом, для неявной функции имеет место тождество $\forall x \in X^{\pm}\colon\ F\big(x, y(x)\big) \equiv 0$.
Неявные функции возникают в различных областях, включая физику, экономику и инженерию. Они позволяют моделировать сложные взаимосвязи между переменными.

Ответ на вопрос, каким условиям должна удовлетворять функция $F(x, y)$, чтобы уравнение $F(x, y) = 0$ определяло единственную функцию $y = y(x)$, дает теорема о существовании неявной функции.

ТЕОРЕМА О НЕЯВНОЙ ФУНКЦИИ [1]. Пусть функция $F(x, y)$ и ее *частные* производные $F'_x(x, y)$, $F'_y(x, y)$ непрерывны в *некоторой окрестности точки* $M(x_0, y_0)$ и при этом $F(x_0, y_0) = 0$, $F'_y(x_0, y_0) \neq 0$. Тогда уравнение $F(x, y) = 0$ определяет в этой окрестности точки $M(x_0, y_0)$ единственную неявную функцию, монотонную, непрерывную и дифференцируемую в некотором интервале, содержащем точку x_0, причем $y(x_0) = y_0$.

Из этой теоремы следует, что равенство $F(x, y) = 0$ определяет неявную функцию $y = y(x)$, для которой в окрестности точки $M(x_0, y_0)$ имеет место тождество $F\big(x, y(x)\big) \equiv 0$. Тогда *полные* производные $F(x, y)$ по x и y равны нулю

$$
\begin{aligned}
&\frac{dF}{dx} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial y}\frac{dy}{dx} = 0 \Rightarrow \frac{dy}{dx} = y'_x(x) = -\frac{F'_x(x, y(x))}{F'_y(x, y(x))},\\
&\frac{dF}{dy} = \frac{\partial F}{\partial y} + \frac{\partial F}{\partial x}\frac{dx}{dy} = 0 \Rightarrow \frac{dx}{dy} = x'_y(y) = -\frac{F'_y(x, y(x))}{F'_x(x, y(x))}
\end{aligned}
\qquad (1)
$$

Последовательно дифференцируя $y'_x(x)$ и $x'_y(y)$ получим производные более высоких порядков:

$$
\begin{aligned}
&y'_x = -\frac{F'_x}{F'_y}, \qquad y''_x = -\frac{F''_x \cdot F'_y - F'_x \cdot F''_y \cdot y'_x(x)}{{F'_y}^2},\\
&y'''_x = -\frac{F'''_x {F'_y}^4 - F'''_y F'_y {F'_x}^3 + 2F'_x\left(F''_x {F'_y}^2 F''_y + {F'_x}^2 {F''_y}^2\right) - y''_x F'_x F''_y {F'_y}^3}{{F'_y}^5},
\end{aligned}
\qquad (2)
$$

$$\ldots$$
$$y_x^{(n)} = \left[y_x^{(n-1)}\right]'_x, \quad n = 2, 3, \ldots$$
$$x_y^{(n)} = \left[x_y^{(n-1)}\right]'_y, \quad n = 2, 3, \ldots$$

Как следует из теоремы о неявной функции, соотношения (2) справедливы только для точек (x, y), лежащих на кривой, описываемой функцией $F(x, y) = 0$, т.е. точки (x, y) должны быть корнями уравнения $F(x, y) = 0$.

Для численного поиска корней функции $F(x, y) = 0$ будем использовать идеи итерационного метода Гаусса-Зейделя (метод покоординатного спуска) [2, 4]. Суть метода заключается в том, чтобы на каждой итерации по очереди делать шаг вдоль каждого из компонентов x и y: $x_{i+1} = x_i + QX(x_i, y_i)$, $y_{i+1} = y_i + QY(x_{i+1}, y_i)$, где: $QX(.)$ и $QY(.)$ – процедуры определяющие шаг по x и y соответственно.
Слабым местом этого метода являются медленная сходимость, обусловленная использованием только значений функции в итерационных точках.
Для устранения названного недостатка воспользуемся расчетной формулой Чебышева 4-го порядка [2], которая учитывает первую, вторую и третью производные целевой функции:

$$z_{i+1} = z_i - [1 + (L_i + L_i^2)/2 - K_i/6]D_i, \quad i = 0, 1, 2, \ldots \tag{3}$$

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

Тогда итерационные формулы для x_{i+1} и y_{i+1} примут вид:

$$\begin{aligned} x_{i+1} &= x_i - [1 + (L_{xi} + L_{xi}^2)/2 - K_{xi}/6]D_{xi}\big|(x_i, y_i), \\ y_{i+1} &= y_i - \left[1 + \left(L_{yi} + L_{yi}^2\right)/2 - K_{yi}/6\right]D_{yi}\big|(x_{i+1}, y_i), \end{aligned} \tag{4}$$

где:

$$L_{xi} = D_{xi} \frac{F_x''}{F_x'}, \quad K_{xi} = D_i^2 \frac{F_x'''}{F_x'}, \quad D_{xi} = \frac{F}{F_x'}, \quad (x_i, y_i),$$
$$L_{yi} = D_{yi} \frac{F_y''}{F_y'}, \quad K_{yi} = D_{yi}^2 \frac{F_y'''}{F_y'}, \quad D_{yi} = \frac{F}{F_y'}, \quad (x_{i+1}, y_i)$$

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.
Признаком окончания итерационной процедуры (4) служит выполнение неравенства $\|F(x_{i+1}, y_{i+1})\| \le \delta$, где δ – заданная малая величина.

Первым требованием к реализации итерационной процедуры (4) является условие $F'_x \neq 0$ и $F'_y \neq 0$ на интервале $[a, b]$. Второе требование связано с небходимостью наличия способа вычисления частных производных функции $F(x, y)$, когда последняя 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.

Однако, при компьютерной реализации, вычисление частных производных функции $F(x, y)$ может быть успешно решено, если воспользоваться методом автоматического дифференцирования (АД) [5], суть которого заключается в создании нового типа данных и переопределения операций и базовых функций над ними. Для рассматриваемого случая в качестве такого нового типа данных могут быть исползованы гипер-дуальные числа 3-го класса [5].

Гипер-дуальное число 3-го класса имеет представление в виде $Z = z + z_1\boldsymbol{\varepsilon} + z_2\omega + z_3\boldsymbol{\gamma}$, z, z_1, z_2, $z_3 \in \mathbb{R}$. Параметр z называется главной (*Re* - действительной) частью гипер-дуального числа, а z_1, z_2, z_3 – его мнимыми (*Im1*, *Im2*, *Im3* - инфинитезимальными) частями. Абстрактные элементы $\boldsymbol{\varepsilon}$, ω, $\boldsymbol{\gamma}$ образуют базис мнимых частей гипер-дуального числа, отвечающий следующими правилам: $\boldsymbol{\varepsilon^2} = 2\boldsymbol{\omega}$, $\boldsymbol{\varepsilon\omega} = 3\boldsymbol{\gamma}$, $\boldsymbol{\varepsilon\gamma} = \boldsymbol{\omega\gamma} = \boldsymbol{\gamma^2} = \boldsymbol{\omega^2} = 0$. Алгебраические операции над дуальными числами 3-го класса определяются формулами:

$$\begin{aligned}
&X = x + x_1\boldsymbol{\varepsilon} + x_2\omega + x_3\boldsymbol{\gamma}, \qquad Y = y + y_1\boldsymbol{\varepsilon} + y_2\omega + y_3\boldsymbol{\gamma},\\
&X + Y = x + y + (x_1 + y_1)\boldsymbol{\varepsilon} + (x_2 + y_2)\omega + (x_3 + y_3)\boldsymbol{\gamma},\\
&X{\cdot}Y = x{\cdot}y + (x{\cdot}y_1 + y{\cdot}x_1)\boldsymbol{\varepsilon} + (x{\cdot}y_2 + 2x_1{\cdot}y_1 + y{\cdot}x_2)\omega + (x{\cdot}y_3 + y{\cdot}x_3 + 3(x_1{\cdot}y_2\\
&\qquad\qquad + y_1{\cdot}x_2))\boldsymbol{\gamma},\\
&Y^{-1} = y^{-1} - y^{-2}{\cdot}y_1\boldsymbol{\varepsilon} + y^{-2}(2{\cdot}y_1^2{\cdot}y^{-1} - y_2)\omega + a^{-2}(6y_1{\cdot}y^{-1}(y_2 - y_1^2{\cdot}y^{-1} - y_3)\boldsymbol{\gamma},\\
&X/Y = X{\cdot}Y^{-1},
\end{aligned} \tag{5}$$

а функция гипер-дуального аргумента 3-го класса, после ее разложения в ряд Тейлора, имеет вид

$$\begin{aligned}
\Phi(X) = \varphi(x) + x_1\varphi'(x)\boldsymbol{\varepsilon} + [x_2\varphi'(x) + x_1^2\,\varphi''(x)]\omega +\\
+ [x_3\varphi'(x) + 3x_1x_2\varphi''(x) + x_1^3\varphi'''(x)]\boldsymbol{\gamma}
\end{aligned} \tag{6}$$

При $x_1 = 1$ и $x_2 = x_3 = 0$ соотношение (6) принимает форму:

$\Phi(x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}) = \varphi(x) + \varphi'\boldsymbol{\varepsilon} + \varphi''(x)\boldsymbol{\omega} + \varphi'''(t)\boldsymbol{\gamma}, \quad \varphi(x) = \Phi(x + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma})$,
что позволяет выразить частные производные функции $F(x, y)$ в терминах гипер-дуальных чисел 3-го класса.

Пусть

$$X = x + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}, \quad \bar{X} = \bar{x} + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma},$$
$$Y = y + \boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma}, \quad \bar{Y} = \bar{y} + 0\boldsymbol{\varepsilon} + 0\boldsymbol{\omega} + 0\boldsymbol{\gamma},$$

тогда

$$F(X, \bar{Y}) = F(x, \bar{y}) + F_x'(x, \bar{y})\boldsymbol{\varepsilon} + F_x''(x, \bar{y})\boldsymbol{\omega} + F_x'''(x, \bar{y})\boldsymbol{\gamma},$$
$$F(\bar{X}, Y) = F(\bar{x}, y) + F_y'(\bar{x}, y)\boldsymbol{\varepsilon} + F_y''(\bar{x}, y)\boldsymbol{\omega} + F_y'''(\bar{x}, y)\boldsymbol{\gamma}, \tag{7}$$

где величины с «шапкой» не являются переменными.
Из (7) следует

$$F_x'(x, \bar{y}) = F(X, \bar{Y}).Im1, \quad F_x''(x, \bar{y}) = F(X, \bar{Y}).Im2, \quad F_x'''(x, \bar{y}) = F(X, \bar{Y}).Im3,$$
$$F_y'(\bar{x}, y) = F(\bar{X}, Y).Im1, \quad F_y''(\bar{x}, y) = F(\bar{X}, Y).Im2, \quad F_y'''(\bar{x}, y) = F(\bar{X}, Y).Im3 \tag{8}$$

Таким образом, итерационная процедура (4) с учётом соотношений (8) дает возможность получить численное решения уравнения $F(x, y)$ с заданной точностью.

Компьютерная реализация гипер-дуальных чисел 3-го класса (Sdn3) была выполнена на языке программирования SWIFT 5 для macOS Sonoma 14.6.1 в виде статической библиотеки (static library) SdnLibrary3 [5], которую можно скачать с интернет ресурса (см. Приложение 1), а затем добавить в свой Xcode проект.

Для выполнения численных экспериментов по формулам (4) и (8) в Приложение 1 приводится код итерационной процедуры ImpRoot(...), реализирующей названные формулы, и примеры некоторые из тестируемых функций.
Там же дана процедура Derivatives(...) для нахождения первых трех производных $y(x)$ по формулам (2), и процедура GraphXY(...), использующая ImpRoot(...), для вычисления точек графика функции $y(x)$.

С целью проверки адекватности работы прикладный программы ImpRoot(...) была проведена верификация вычислительных алгоритмов на основе сравнения численных и аналитических решений, полученных для тестовых задач. Расчеты проводились в операционной среде macOS Sonoma 14.6.1 с CPU 2.3 GHz.

В таблице 1 представлены результаты тестируемых функций.

$F(x, y)$	(x_0, y_0)	x^*	y^*	i	t
$y^2 + x^2 - 1$	$(0.5, \overline{0.0})$	1.0	0.0	4	0.072
	$(\overline{0.0}, 0.5)$	0.0	1.0	4	0.072
	$(0.5, 0.5)$	0.9877175535	0.1562499107	2	0.036

$y - xe^x + 1$	$(0.5, \overline{0.0})$	0.5671432904	0.0	2	0.036
	$(\overline{0.0}, 0.5)$	0.0	-1.0	1	0.036
	$(0.5, 0.5)$	0.7313905789	0.5189063489	1	0.036
$y - x - \dfrac{\sin(y)}{2} - 1$	$(-0.5, \overline{0.0})$	-1.0	0.0	1	0.041
	$(\overline{0.0}, 0.5)$	0.0	1.4987011335	3	0.041
	$(-0.5, 0.5)$	-0.7397127693	0.5	1	0.044
$y^2 - xy + x^2 - 1$	$(0.5, \overline{0.0})$	1.0	0.0	4	0.036
	$(\overline{0.0}, 0.5)$	0.0	1.0	4	0.038
	$(1.0, 0.8)$	1.1218106993	0.7978770067	2	0.039
$x^2 \sin(y) + xy - 1$	$(0.5, \overline{1.5})$	0.5002507275	1.5	1	0.041
	$(\overline{1.5}, 0.5)$	1.5	0.2685974744	2	0.041
	$(0.5, 0.5)$	1.0683837393	0.4608616427	2	0.046
$y + e^y\sqrt{1-x} - 0.5$	$(0.7, \overline{0.0})$	0.75	0.0	1	0.046
	$(\overline{0.0}, 0.0)$	0.0	-0.2662486082	2	0.046
	$(0.0, 0.0)$	0.75	0.0	1	0.046
$e^{yx} - \ln(x^2 + y^2)$	$(1.0, \overline{0.0})$	1.6487212707	0.0	3	0.042
	$(\overline{0.0}, -1.0)$	0.0	-1.6487212707	3	0.059
	$(0.0, 1.0)$	-0.8333330433	0.9404418502	2	0.043

Таблица 1.Результаты тестируемых функций где: (x_0, y_0) *– начальное приближение;* x^*, y^**– корень уравнения;* i *– кол-во итеракций;* t *– затраченное время в миллисекундах*

Рассмотренный прием численного решения неявных уравнения основаный на методе покоординатного спуска и итеррационной формулы Чебышева 4-го порядка с использованием автоматического дифференцирования на базе супер-дуальных чисел 3-го класса показал свою несомненную актуальноность, особенно для функций, являющимися весьма сложными композициями базовых функций.

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS Sonoma 14.6.1). 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-14;
let Nmax = 1000; // max number of iterations

enum rootOf{
   case F //- for R(x,y)  = 0
   case y //- for R(x,y0) = 0
   case x //- for R(x0,y) = 0
}

//-- Root of Implicit function
func ImpRoot(x0:Double,y0:Double, f:(Sdn3,Sdn3) -> Sdn3,
          rf:rootOf)->(x:Double,y:Double,i:Int)
{   let δ:Double = 1E-14;
   var X = Sdn3(re: x0),  Y = Sdn3(re: y0);
   var z = 0.0, i = 1;
   func CHH(X:Sdn3,Y:Sdn3)->Double{
      let Z =  f(X, Y),
          D = Z.re/Z.im1, L = D*Z.im2/Z.im1, K = D*D*Z.im3/Z.im1;
      return (1.0 + 0.5*(L + L*L) - K/6.0)*D;
   }
   while (true){
      if rf == .x {X = Sdn3(re: x0)}
      else {Y.im1 = 0; X.im1 = 1;
           z = CHH(X: X, Y: Y)             if z.isNaN {break}
           X = Sdn3(re: X.re - z, im1: 1, im2: 0, im3: 0);
      }
      if rf == .y {Y = Sdn3(re: y0)}
      else {X.im1 = 0; Y.im1 = 1;
           z = CHH(X: X, Y: Y);            if z.isNaN {break}
           Y = Sdn3(re: Y.re - z, im1: 1, im2: 0, im3: 0);
      }
      if abs(f(X,Y).re) <= δ {break}
      i += 1
      if i > Nmax {break}
   }
   return (x:X.re,y:Y.re,i:i);
}

//-- Derivatives of y(x)
func Derivatives(x:Double, y:Double, f:(Sdn3,Sdn3)->(Sdn3))
```

```
   ->(y1:Double,y2:Double,y3:Double){
   var X = Sdn3(re: x, im1: 1, im2: 0, im3: 0), Y = Sdn3(re: y);
      let FX = f(X, Y);
   X = Sdn3(re: x); Y = Sdn3(re: y, im1: 1, im2: 0, im3: 0);
      let FY = f(X, Y);
   let y1 = -FX.im1 / FY.im1;
   let y2 =  (FX.im1*FY.im2*y1 - FX.im2*FY.im1)/(FY.im1*FY.im1);
   let y3 = -(FX.im3*pow(FY.im1, 4) - FY.im3*FY.im1*pow(FX.im1, 3)
          + 2.0*FX.im1*(FX.im2*FY.im2*pow(FY.im1,2) + pow(FX.im1,2)*pow(FY.im2, 2))
          - y2*FX.im1*FY.im2*pow(FY.im1, 3))/pow(FY.im1, 5);
  return (y1:y1, y2:y2, y3:y3);
}

//-- Generating Graph Points
func GraphXY (x1:Double, x2:Double, F:(Sdn3,Sdn3)->(Sdn3))->(x:[Double], y:[Double])
{  var xPoints: [Double] = [], yPoints: [Double] = []
   var r:(x:Double,y:Double,i:Int);
      for x:Double in stride(from: x1, through: x2, by: 0.01) {
         r = ImpRoot(x0: x, y0: 0.5, f: F, rf: .x);
         xPoints.append(r.x);    yPoints.append(r.y);
      }
      for x:Double in stride(from: 1, through: -1, by: -0.01) {
         r = ImpRoot(x0: x, y0: -0.5, f: F, rf: .x);
         xPoints.append(r.x);   yPoints.append(r.y);
      }
   return (x:xPoints, y:yPoints);
}

//-- Example of use
let F1:(Sdn3,Sdn3)->Sdn3={X,Y in X*X + Y*Y - Sdn3(re: 1)}
let F2:(Sdn3,Sdn3)->Sdn3={X,Y in X*X*Sdn3.sin(X: Y) + X*Y - Sdn3(re: 1)}
let F3:(Sdn3,Sdn3)->Sdn3={X,Y in Sdn3.exp(X: X*Y) - Sdn3.ln(X: X*X + Y*Y)}

let r = ImpRoot(x0: 0.5, y0: 0.0, f: F1, rf: .x);
let y123 = Derivatives(x: r.x,  y: r.y, f:F1);
let xyP =  GraphXY(x1:-1.0, x2:1.0, F:F1);
```

ЛИТЕРАТУРА

1. *Зорич В. А.* Математический анализ.Часть I. – Изд. 10-е, испр. – М.: МЦНМО, 2019. – 564 с.
2. *Шарый С. П.* Курс вычислительных методов. – Новосибирск: Новосиб. гос. ун-т.,

2015. – 507 с.
3. *Антонюк П.* О первой работе П. Л. Чебышева. – Научно-Исслед. Семинар по истории и методологии мат. и мех.: МГУ, 2014. –14 с.
4. *Березин И. С., Жидков Н. П.* Методы вычислений. Т.2 – Изд. Физико-математической литературы. – М: 1958. – 620 с.
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.

Абстракт

В данной публикации рассматривается метод численного решения неявных функций, основанный на итерационных формулах Гаусса-Зейделя, Чебышева и автоматического дифференцирования с использованием специальных дуальных чисел (супер-дуальных чисел 3-го класса). Представлена компьютерная реализация этого подхода на языке SWIFT операционной системы macOS. Проведены численные эксперименты.

This publication discusses a method for numerically solving implicit functions based on the iterative formulas of Gauss-Seidel, Chebyshev and automatic differentiation using special dual numbers (super-dual numbers of the 3rd class). A computer implementation of this approach in the SWIFT language of the macOS operating system is presented.
Numerical experiments are conducted.

Ключевые слова*: неявные функции, итерационные методы, автоматическое дифференцирование, супер дуальные числа, корни неявных функций, implicit functions, iterative methods, automatic differentiation, super dual numbers, roots of implicit functions*

20 сентября 2024 г.