КАЛЬКУЛЯТОР КОРНЕЙ НЕЯВНЫХ ФУНКЦИЙ НА БАЗЕ МЕТОДА ЧЕБЫШЕВА 4-ГО ПОРЯДКА С ИСПОЛЬЗОВАНИЕМ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Представляемый пилотный проект является интерактивным (online) калькулятором позволяющим вычисление локального корня (с точностью 15 знаков после запятой) неявной плоской функции и некоторых его дифференциальных характеристик.

Неявная (implicit) плоская функция – это функция, описывающая плоскость в виде неявного уравнения $f(x,y)=0$. Это неявное уравнение связывает две координатные переменные и определяет плоскую неявную кривую. В общем случае любая плоская неявная кривая задаётся указанным уравнением. Следовательно, неявная функция может рассматриваться как множество нулей функции от двух переменных. Неявные функции возникают в различных областях, включая физику, экономику и инженерию. Они позволяют моделировать сложные взаимосвязи между переменными. В этом случае, одной из актуальных задач является задача определения корней уравнения $f(x,y)=0$, которая формулируется так: найти решение указанного уравнения при разных значениях $x=x^*$.

Учитывая тот факт, что функция $f(x,y)$ может быть сложной композитной функцией, аналитическое решешение которой весьма затруднительно или вообще невозможно, на практике применяют различные численные итерационные методы. Например, метод Чебышева 4-го порядка [1], итерационная формула которого имеет вид:

$$y_{i+1} = y_i - [1 + (L_i + L_i^2)/2 - K_i/6]D_i, \qquad i = 0,1,2,\dots, \tag{1}$$

где: $L_i = D_i \cdot f''(x^*,y_i)/f'(x^*,y_i)$, $K_i = D_i^2 \cdot f'''(x^*,y_i)/f'(x^*,y_i)$, $D_i = f(x^*,y_i)/f'(x^*,y_i)$,

$$f'(x^*,y) = \frac{\partial f(x^*,y)}{\partial y},\ f'(x^*,y) = \frac{\partial^2 f(x^*,y)}{\partial y^2}, f'(x^*,y) = \frac{\partial^3 f(x^*,y)}{\partial y^3}$$

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 (1) служит условие $f'(x^*,y) \neq 0$ на рассматриваемом интервале $[a,\ b]$. Второе требование связано с небходимостью наличия способа вычисления $f(x^*,y)$, $f'(x^*,y)$, $f''(x^*,y)$ и $f'''(x^*,y)$ когда $f(x^*,y)$ имеет сложное аналитическое выражение или представлена сложным программный кодом. В этом случае, как правило, используются методы численного дифференцирования с применением различных моделей аппроксимации. Последнее приводит к возникновению 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 (AD) [2], 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. Если использовать гипер-дуальные числа третьего класса (HDN3) [3], то за одно обращение к перезагруженной функции вычисляются значения самой функции и её первых трех производных.

Гипер-дуальные числа (HDN3) находять достаточно широкое применение в самых разнообразных задачах прикладной математики (см., например, веб-ресурс: HDN3).

Эти числа (HDN3) имеют представление в виде $X = x + x_1\boldsymbol{\varepsilon} + x_2\boldsymbol{\omega} + x_3\boldsymbol{\gamma}$, $(x, x_1, x_2, x_3) \in \mathbb{R}$. Параметр x называется главной (*Re* - действительной) частью гипер-дуального числа, а x_1, x_2, x_3 – его мнимыми (*Im1*, *Im2*, *Im3* - инфинитезимальными) частями. Абстрактные элементы $\boldsymbol{\varepsilon}, \boldsymbol{\omega}, \boldsymbol{\gamma}$ образуют базис мнимых частей гипер-дуального числа, отвечающий следующими правилам: $\boldsymbol{\varepsilon}^2 = 2\boldsymbol{\omega}$, $\boldsymbol{\varepsilon}\boldsymbol{\omega} = 3\boldsymbol{\gamma}$, $\boldsymbol{\varepsilon}\boldsymbol{\gamma} = \boldsymbol{\omega}\boldsymbol{\gamma} = \boldsymbol{\gamma}^2 = \boldsymbol{\omega}^2 = 0$. Алгебраические операции над дуальными числами HDN3 определяются формулами:

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

а функция 2-х гипер-дуальных аргументов HDN3, после ее разложения в ряд Тейлора, имеет вид:

$$
\begin{aligned}
&F(X, Y) = f +\\
&+ [x_1 f'_x + y_1 f'_y]\boldsymbol{\varepsilon} +\\
&+ [x_2 f'_x + y_2 f'_y + x_1^2 f''_{xx} + y_1^2 f''_{yy} + 2x_1 y_1 f''_{xy}]\boldsymbol{\omega} +\\
&+ [x_3 f'_x + y_3 f'_y + x_1^3 f'''_{xxx} + y_1^3 f'''_{yyy} + 3(x_1 x_2 f''_{xx} + y_1 y_2 f''_{yy} +\\
&+ (x_1 y_2 + x_2 y_1) f''_{xy} + x_1^2 y_1 f'''_{xxy} + x_1 y_1^2 f'''_{xyy})]\boldsymbol{\gamma},
\end{aligned}
\tag{3}
$$

где f, $f'_{\dots}$, $f''_{\dots}$, $f'''_{\dots}$– функция и ее соответствующие частные производные в точке (x, y). Разные комбинации x_1, x_2, x_3, y_1, y_2 и y_3 позволяют сохранить в (3) те либо иные члены. Наиболее актуальные комбинации (фильтры) x_k и y_k $(dX, dY$ и $dXY)$ приведены в следующей таблице:

	X, Y	$F(X, Y)$
dX	$X = (x, 1, 0{,}0), \quad Y = (y, 0, 0{,}0)$	$f + f_x'\boldsymbol{\varepsilon} + f_{xx}''\boldsymbol{\omega} + f_{xxx}'''\boldsymbol{\gamma}$
dY	$X = (x, 0, 0{,}0), \quad Y = (y, 1, 0{,}0)$	$f + f_y'\boldsymbol{\varepsilon} + f_{yy}''\boldsymbol{\omega} + f_{yyy}'''\boldsymbol{\gamma}$
dXY	$X = (x, 1, 0{,}0), \quad Y = (y, 1, 0{,}0)$	$f + [f_x' + f_y']\boldsymbol{\varepsilon} + [f_{xx}'' + 2f_{xy}'' + f_{yy}'']\boldsymbol{\omega} +$ $+ [f_{xxx}''' + f_{yyy}''' + 3(f_{xxy}''' + f_{xyy}''')]\boldsymbol{\gamma}$

Таблица 1. Актуальные фильтры $\boldsymbol{dX}, \boldsymbol{dY}$ и $\boldsymbol{dXY}$ *для HDN3*

Из таблицы 1 видно, что главные части dX, dY и dXY содержать значение функции в точке (x, y), а мнимые части – частные производные:

- $Im_1\ (dX) = f_x'$, $Im_2(dX) = f_{xx}''$, $Im_3(dX) = f_{xxx}'''$
- $Im_1\ (dY) = f_y'$, $Im_2(dY) = f_{yy}''$, $Im_3(dY) = f_{yyy}'''$
- $Im_1\ (dXY) = f_x' + f_y'$, $Im_2(dXY) = f_{xx}'' + 2f_{xy}'' + f_{yy}''$, $Im_3(dX) = f_{xxx}''' + f_{yyy}''' + 3(f_{xxy}''' + f_{xyy}''')$

Если исходная функция удовлетворяет условиям теоремы Шварца [4], т.е. порядок дифференцирования не имеет значения, то $f_{xy}'' = f_{yx}''$ и

$$
\begin{aligned}
f_x' &= Im_1(dX), \qquad & f_y' &= Im_1(dY),\\
f_{xx}'' &= Im_2(dX), \qquad & f_{yy}'' &= Im_2(dY),\\
f_{xxx}''' &= Im_3(dX), \qquad & f_{yyy}''' &= Im_3(dY),\\
f_{xy}'' &= f_{yx}'' = [Im_2(dXY) - Im_2(dX) - Im_2(dY)]/2
\end{aligned}
$$

Для вычисления f_{xyy}''' и f_{xxy}''' введем фильтер $dXY^- = dXY$ при $X = (x, 1, 0, 0)$, $Y = (y, -1,\ 0, 0)$, тогда сумма и разность dXY и dXY^- равны

$$dXY + dXY^- = 2\{f + [f_{xxx}''' + 3f_{xyy}''']\boldsymbol{\gamma}\}, \quad XY - dXY^- = 2[f_{yyy}''' + 3f_{xxy}''']\boldsymbol{\gamma}\},$$

откуда

$$f_{xyy}''' = [Im_3\,(dXY + dXY^- - Im_3(dX))]/3, \qquad f_{xxy}''' = [Im_3\,(dXY - dXY^- - Im_3(dY))]/3$$

На практике использование AD предполагает что целевая функция $f(x, y)$ уже определена до этапа компеляции программы. Для возможности изменения целевой функции во время работы уже откомпелированной программы будем использовать парсинг/разбор математического выражения. Ниже представлена функциональная схема предлагаемого калькулятора.

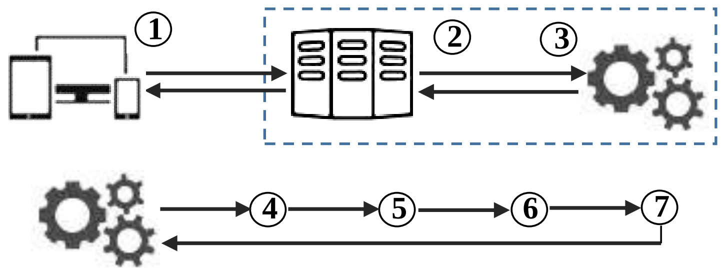

Рис.1. Функциональна схема калькулятора

На рис.1:

① – интерфейс пользователя из которого осуществляется запрос к веб-серверу;
② – веб-сервер;
③ – блок решения задачи;
④ – анализ и форматирование строки математического выражения;
⑤ – разбор математического выражения;
⑥ – токенизация выражения в формате польской нотации (RPN);
⑦ – вычисление токенизированного выражения с использованием автоматического дифференцирования (AD) на базе гипер-дуальных чисел (HDN3) и выполнения итерационной формулы (1);

Для реализации блока ③ используются библиотеки Sdn3Lib.dll и MathParserVIO.dll, написанные на C#, осуществляющие поддержку HDN3 и парсинг исходного строкового выражения функции $f(x, y)$ соответственно. На веб-сервер также реализован метод Чебышева (1) и ряд вспомогательных процедур.

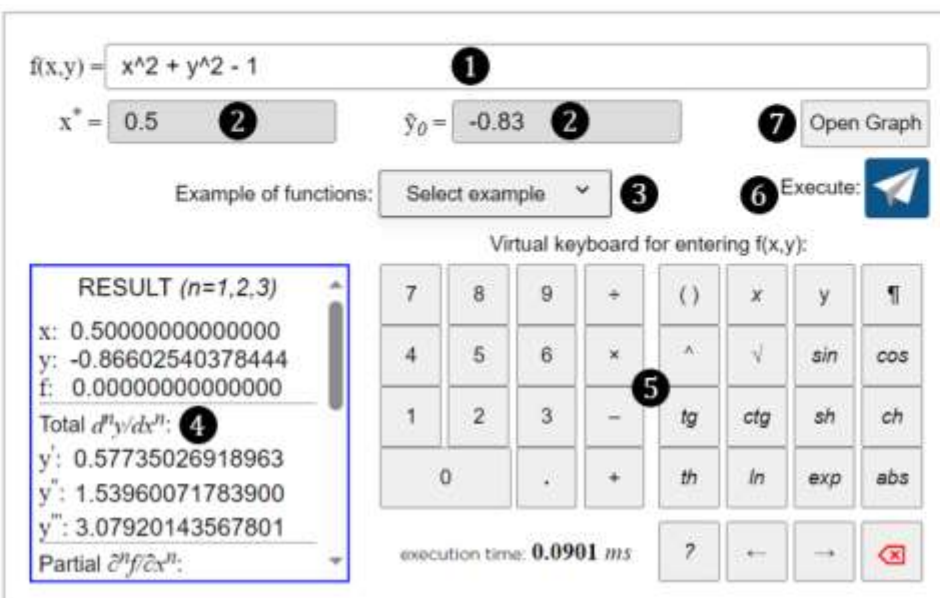

Рис.2. Интерфейс пользователя

Интерфейс пользователя представлен на рис. 2.

❶ - поле ввода функции $f(x, y) = 0$;
❷ - поля ввода $(x^*, \tilde{y}_0)$ (осуществляются через графический калькулятор ❼);
❸ - список примеров функций (ввод выбранного примера функции в поле ❶);
❹ - блок результатов расчета;
❺ - клавиатура ввода функции $f(x, y)$ в поле ❶;
❻ - кнопка выполнения расчёта;
❼ - кнопка открытия графического калькулятора;

Исходными данными служат три величины: строковое выражение функции $f(x, y)$, величины x^* и $\tilde{y}_0$ (где: x^* – заданное значение x; $\tilde{y}_0$ – начальное приближение для метода Чебышева). Определение хорошего начального приближения $\tilde{y}_0$ для любых итерационных процессов является весьма важной, сложной и трудно выполнимой задачей. Наиболее часто применяется анализ графика исходной функции. Для этой цели, используя графический пакет JSXGraph на базе Scalable Vector Graphics (SVG), разработан графический калькулятор, интерфейс которого приведен на рис.3.

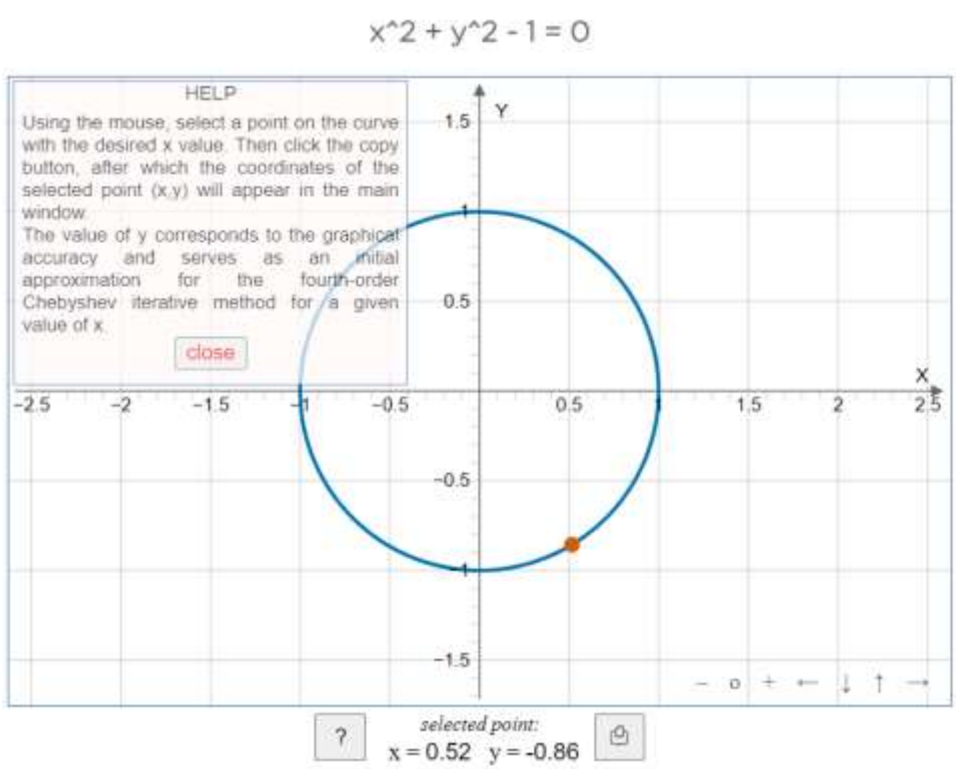

Рис.3. Интерфейс графического калькулятора

С помощью «мыши», перемещая ● указатель можно выбрать точку на кривой $f(x, y)$ с нужным значением x, после чего в полях ❷ интерфейса пользователя появятся координаты выбранной точки (x, y). Значение y соответствует графической точности и служит начальным приближением $\tilde{y}_0$ для итерационного метода Чебышева четвертого порядка при заданном значении x^*, который обеспечивает точность результата до 15 знаков после запятой за минимальное количество итераций.

Результатами расчета являются: x^*, $y(x^*)$, $f(x^*, y)$, $y'(x^*)$, $y''(x^*)$, $y'''(x^*)$, $f_x'(x^*, y)$, $f_x''(x^*, y)$, $f_x'''(x^*, y)$, $f_y'(x^*, y)$, $f_y''(x^*, y)$, $f_y'''(x^*, y)$, $æ(x^*, y)$, $R(x^*, y)$, $runtime$, $iterations$.

Для нахождения частных производных используются фильтры dX, dY (см.табл.1) и dXY^-. Вычисление полных производных y_x' и y_x'' осуществляется по формулам аналогичным [5]:

$$y_x' = -\frac{f_x'}{f_y'}, \qquad y_x'' = -\frac{f_x'' + 2f_{xy}'' \cdot y_x' + f_y'' \cdot {y_x'}^2}{f_y'},$$

Полное выражение для y_x''' весьма громоздко и поэтому здесь не приводится. Исскуственный интеллект Copilot AI предлогает заметно упрощенную формулу для y_x''':

$$y_x''' := \frac{\varphi_1 - \varphi_2}{f_y^2},$$

где:

$$\varphi_1 = f_{xy}''\left(f_x'' + 2f_{xy}'' \cdot y_x' + f_y'' \cdot {y_x'}^2\right),$$

$$\varphi_2 = f_y' \, [f_x''' + 2\left(y_x' \cdot f_{xxy}''' + y_x'' \cdot f_{xy}''\right) + f_{xyy}''' \cdot y_x'^2 + 2\, f_y'' \cdot y_x' \cdot y_x'']$$

Заметим, что полные производные неявных функций находят широкое применение: в оптимизации, экономике, физике, технике, дифференциальной геометрии, эконометрических моделях и других научно-технических областях.

И, наконец, кривизна кривой в точке (x^*, y) находится согласно [6] с использованием фильтров dX, dY, dXY:

$$æ = \left(-f_y' \;\; f_x'\right) \cdot \begin{pmatrix} f_x'' & f_{xy}'' \\ f_{xy}'' & f_y'' \end{pmatrix} \cdot \begin{pmatrix} -f_y \\ f_x \end{pmatrix} / ({f_x'}^2 + {f_y'}^2)^{3/2} = \frac{{f_x'}^2 \cdot f_y'' - 2 f_x' \cdot f_y' \cdot f_{xy}'' + {f_y'}^2 \cdot f_x''}{({f_x'}^2 + {f_y'}^2)^{3/2}},$$

а радиус кривизны $R = 1/|æ|$.

Подводя итоги, отметим:

- Интерфас пользователя выполнен на ASP.NET C#. Для реализации 3-го блока решения задачи (см. рис. 1) на языке C# разработаны две статические библиотеки Sdn3Lib.dll и MathParserVIO.dll. Первая из них реализует гипердуальные числа (HDN3), а вторая – парсинг строки исходного математического выражения.

- Графический калькулятор для определения начального приближения $\tilde{y}_0(x^*)$ разработан на основе графического пакета JSXGraph.

- Для тестируемых примеров, в зависимости от сложности исходной функции, время выполнения запроса находилось в диапазоне 0.03 ÷ 8.0 ms. Точность получаемых результатов составляет 15 знаков после запятой.
- Рабочая версия представленного калькулятора находится по адресу Calculator_Imp.

Литература

1. *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-го_порядка.pdf, 2023.

2. *Fike J.A.* Multi-objective optimization using hyper-dual numbers. Ph.D. Dissertation, Stanford University, 2013.

3. *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.

4. *Тер-Криков А. М., Шабунин М. И.* Курс математического анализа. – Изд-во физмат-лит., М., 2001. – 672 с.

5. *Олифер В. И.* К численному решению неявных функций с использованием гипер-дуальных чисел. – URL:http://viosolutions.amerihomesrealty.com/pdf/Неявные_функции_z.pdf, 2020.

6. *Goldman R.* Curvature formulas for implicit curves and surfaces. – Computer Aided Geometric Design 22 (2005) 632–658.

Абстракт

В данной публикации рассматривается пилотный проект интерактивного (online) калькулятора, позволяющего вычисление значения локального корня невной функции и её первых трех частных и полных производных. Решение строится на основе автоматического дифференцирования с использованием специальных дуальных чисел (супер-дуальных чисел 3-го класса), итерационной процедуры Чебышева четвертого порядка и парсинга строкового математического выражения.

This publication discusses a pilot project of an interactive (online) calculator that allows calculating the value of a local root of a variable implicit function and its first three partial and total derivatives. The solution is based on automatic differentiation using special dual numbers (super-dual numbers of the 3rd class), the fourth-order Chebyshev iteration procedure, and parsing of a string mathematical expression..

Ключевые слова: *дуальные числа, гипер-дуальные числа, супер-дуальные числа, автоматическое дифференцирование, калькулятор, корни неявного уравнения, метод Чебышева, парсинг, токенизация, dual numbers, hyper-dual numbers, super-dual numbers, automatic differentiation, calculator, roots of an implicit equation, Chebyshev's method, parsing, tokenization*

20 декбря 2024 г.