DOI: 10.17117/nv.2015.04.039

<http://ucom.ru/doc/nv.2015.04.039.pdf>

Поступила (Received): 12.11.2015

Заика И.В., Тюшнякова И.А.
Программное определение нулей функций и
разностных решений систем обыкновенных
дифференциальных уравнений на основе
сортировки

Zaika I.V., Tjushnyakova I.A.
Software definition of functions zeroes
and difference solutions of systems ordinary
differential equations on the basis of the sort

В статье описан метод вычисления нулей многочленов и функций с учетом кратности. Предлагаемая конструкция основана на применении внутренней адресной распараллеливаемой сортировки по ключу, производимой слиянием и сохраняющей порядок равных элементов. При этом сортировка используется в качестве конструктивной части алгоритма, лежащего в основе метода. Процесс обработки использует лишь операции сравнения, что исключает накопление погрешности, влечет высокую точность локализации нулей. Функция может задаваться значениями на равномерной сетке. В частности, идентифицируются нули разностных решений обыкновенных дифференциальных уравнений. Метод распространяется на вычисление полюсов функций с учетом их порядка. Предложенные схемы отличаются от известных по построению на основе сортировки, свойством автоматической локализации области всех нулей и каждого нуля в отдельности, параллелизмом, а также вычислительной устойчивостью. Проведен численный эксперимент, который подтверждает надежность и устойчивость метода, выполнен сравнительный анализ с известными методами, включая системы компьютерной математики Maple и MathCAD, иллюстрирующий положительные отличия предложенного метода

Ключевые слова: сортировка, нули функций

The article describes the method of calculation of zeroes of polynomials and functions according to multiplicity. The proposed design is based on the use of internal address parallelizable sort by key, made merger and preserves the order of equal elements. When this sorting is used as a structural part of algorithm underlying the method. The processing uses a comparison operation, which eliminates the accumulation of errors involves high precision localization of zeros. The function values can be set on a uniform grid. In particular, it identified the zeros of the difference solutions of ordinary differential equations. The method applies to the calculation of the poles of the functions with regard to their order. The proposed schemes differ from known by construction on the basis of sorting feature automatic localization of the field all zeros and each zero individually, parallelism and computational stability. The numerical experiment, which confirms the reliability and stability of the method, the comparative analysis with known methods, including a system of computer mathematics Maple and MathCAD, illustrating the positive differences of the proposed method

Key words: sort, functions zeroes

Заика Ирина Викторовна

Кандидат технических наук, доцент
Таганрогский институт имени А.П. Чехова
(филиал РГЭУ РИНХ)
г. Таганрог, ул. Инициативная, 48

Zaika Irina Victorovna

Candidate of Engineering Sciences, assistant
Professor
Taganrog Institute named after A.P. Chekhov
Taganrog, Iniziativnaya st., 48

Тюшнякова Ирина Анатольевна

Кандидат технических наук, доцент
Таганрогский институт имени А.П. Чехова
(филиал РГЭУ РИНХ)
г. Таганрог, ул. Инициативная, 48

Tjushnyakova Irina Anatolievna

Candidate of Engineering Sciences, assistant
Professor
Taganrog Institute named after A.P. Chekhov
Taganrog, Iniziativnaya st., 48

Введение (Introduction)

В работе описывается конструкция метода нахождения действительных и комплексных нулей и полюсов функций с учетом их кратности, а также нулей разностных решений систем обыкновенных дифференциальных уравнений (ОДУ).

Актуальность тематики обусловлена важностью практических приложений численных методов определения нулей функций. Данная задача актуальна сама по себе, на ее основе решается проблема приближенного вычисления полюсов функций [1]. Аналитические функции с кратными полюсами имеют приложения в гидродинамике при рассмотрении обтекания тел возмущенным потоком [2]. Расчет нулей и полюсов функций является одной из важных задач цифровой обработки сигналов [3, 4]. Например, в зависимости от расположения нулей и полюсов на комплексной плоскости осуществляется анализ временных функций линейной динамической системы [5]. Значения нулей и особенностей функций требуется знать при цифровой обработке в таких областях, как акустика, звуковая локация, радиолокация, сейсмология, связь, системы передачи данных, ядерная техника и др.

Современное состояние исследований в области локализации нулей функций одной и нескольких переменных характеризуется следующим образом [6-9]. Существуют компьютерные методы вычисления всех нулей полиномов произвольной степени при условии явного задания их коэффициентов. Такие методы реализованы на основе символьной обработки в системах компьютерной математики Maple, MathCAD. Кроме того, нули полинома вычисляются как собственные числа матрицы Фробениуса, строка которой задается коэффициентами полинома. В более общем случае, когда требуется вычислить нули функций произвольного вида, включая трансцендентные, известные методы строятся как методы вычисления экстремумов, в своем большинстве базирующиеся на градиентных схемах. При этом схемы не включают автоматическую идентификацию области каждого нуля (экстремума). В своем большинстве известные методы [10] ориентированы на нахождение единственного нуля, расположенного внутри некоторого интервала. Помимо того, присутствует необходимость поиска первого приближения, после чего возможно использование расчетных формул конкретного метода. Известные подходы включают больше средств теоретической математики, чем специфики компьютерных вычислений. В этом можно видеть одну из причин, по которой компьютерная реализация методов

нахождения нулей функций общего вида в существенных аспектах не гарантирует завершенности. С другой стороны, при наличии возможности программной реализации некоторых схем, в общем случае, не исключается проблема локализации, накопления погрешности и вычислительной неустойчивости.

Описанный в статье метод отличается от известных построением на основе алгоритмов сортировки, включает автоматическую программную идентификацию с наперед заданной точностью окрестности каждого нуля исходной функции и позволяет вычислять нули с учетом их кратности. Предлагаемый метод, в отличие от известных, инвариантен относительно вида функции, обладает устойчивой компьютерной реализацией, строится на основе параллельной сортировки и, на этой основе, эффективно распараллеливается.

Материалы и методы (Materials and Methods)

Базовая часть метода вычисления нулей многочленов и функций, а также листинги программ представлены в [11-14], ниже излагается видоизменение метода в контексте постановки вопроса. Первоначально будем рассматривать многочлен $P_n(z) = \sum_{\ell=0}^n a_{\ell} z^{\ell}$, у которого могут быть комплексные нули и комплексные коэффициенты. Умножив $P_n(z)$ на комплексно-сопряженное значение $\overline{P_n(z)}$, получим функцию $f(x, y) = P_n(z) \times \overline{P_n(z)} = |P_n(z)|^2$. Очевидно, нахождение нулей $P_n(z)$ сводится к нахождению нулей функции $f(x, y)$ от двух действительных переменных. Вводится равномерная прямоугольная сетка, заключающая внутри себя область всех нулей функции. Значения функции берутся во всех узлах сетки и среди них ищутся все возможные минимумы, которые принимаются за приближения нулей функции $f(x, y)$. Искомые минимумы локализуются с помощью сортировки. Используется быстрая сортировка, основанная на адресном слиянии (рис.1) двух упорядоченных массивов по матрице сравнений (МС) [15]:

	$-\infty$	2	3	3	∞
$-\infty$	0	+	+	+	+
3	-	-	0	0	+
4	-	-	-	-	+
4	-	-	-	-	+
∞	-	-	-	-	0

Рис. 1. Матрица сравнений упорядоченных массивов

Последовательный алгоритм слияния выполняется поэлементным обходом слева направо, сверху вниз неотрицательной области МС. Адрес вставки элемента в выходной массив задается соотношениями

$$c_{i+j} = \begin{cases} b_i, & \text{если } \alpha_{ij} = -1, \alpha_{i(j+1)} \geq 0, \\ a_j, & \text{если } \alpha_{ij} \geq 0, \alpha_{(i+1)j} = -1, \end{cases} \quad \alpha_{ij} = \begin{cases} 1, & a_j > a_i, \\ 0, & a_j = a_i, \\ -1, & a_j < a_i, \end{cases}$$

где i, j – произвольно взятые индексы при ограничениях $n+1 \geq i \geq 0 \geq j \geq m+1$; n, m , соответственно, – число элементов упорядоченных массивов b и a ; c – массив, получаемый в результате их слияния; α_{ij} – элемент i -й строки и j -го столбца МС; при этом a располагается на входе таблицы, задающей МС, горизонтально сверху, b – вертикально слева. По входным индексам i, j на основе данных соотношений однозначно указывается выходной индекс ℓ и обратно, – с учетом принадлежности элемента конкретно к одному из двух входных массивов. Такая обратная адресация сохраняется для сортировки слиянием, которая организуется вначале «слиянием» каждой пары элементов с запоминанием обратных адресов, затем, аналогично, последовательным слиянием упорядоченных пар, упорядоченных четверок и т. д. Программа сортировки представлена в [12].

Для нахождения нулей функции $f(x, y)$ выполняется проход в направлении оси ОУ вдоль столбцов прямоугольной сетки, во время которого находится минимальное по строкам значение $c(j) = \min_{\substack{j=\text{const} \\ 1 \leq i \leq M}} f(x_i, y_j)$ при фиксировании номера

столбца и этот минимум заносится на вход сортировки как элемент $c(j)$ сортируемого массива. К выходу процедуры подсоединяется условный оператор, локализирующий после выполнения сортировки все минимумы среди этих значений, – минимумы совпадают с искомыми приближениями нулей многочлена. Данный оператор (ниже оператор локализации минимума) имеет вид

```
k:=1; while k<= n do begin FOR l := 1 TO k-1 do
if abs(e3[k]-e3[k-l]) <= eps0/h then goto 22; xk:= x0+e3[k]*h;
22: k:=k+1; end;
```

В этом программном фрагменте n – число узлов, совпадающее с числом сортируемых элементов, h – шаг равномерной сетки, $eps0$ – константа, равная радиусу $eps0$ -окрестности текущего узла (вдоль оси ОХ). Эта константа выбирается априори таким образом, чтобы не превысить половины числа узлов, отсчитываемых вдоль этой оси между ближайшими друг к другу нулями многочлена.

Оператор локализации находит минимальные элементы дискретной последовательности по смыслу своего построения. Работая после сортировки, он для текущего узла, определяемого индексом, записанным в $e3[k]$, находит каждый узел $x_0 + e3[k] * h$, в проекции $eps0$ -окрестности которого на ось ОХ нет узлов, доставляющих значения элементов входной последовательности (модуля многочлена на сетке), предшествующие $c[e3[k]]$ в отсортированном массиве. Это означает, в частности, что модуль многочлена в узле $x_0 + e3[k] * h$ не больше значений в остальных узлах его проекции $eps0$ -окрестности на ось ОХ, точнее, – строго меньше, или это значение является первым слева направо среди равных

элементов на выходе отсортированного массива. Значение локализованной абсциссы точки минимума $xk := x0 + e3[k] * h$ дает привязку к локализуемой точке двумерного минимума. Оно фиксируется и аналогичным образом локализуется ордината $yk := y0 + e3[k1] * h$, в которой достигается приближение к минимальному значению $f(x, y)$. Для нахождения нулей $P_n(z)$ осталось выполнить спуск к наименьшему значению в окрестности локализованной точки.

Очевидно, метод с упрощениями переносится на случай нахождения действительных нулей, как частный случай сказанного выше.

Вычисление кратности нулей многочленов и функций базируется на циклическом возобновлении схемы нахождения нулей функции после деления исходной функции на выражение $pp := pp * (sqr(x - kor\ x) + sqr(y - kor\ y))$, где число шагов цикла равно текущему счетчику кратности, а $kor\ x + i * kor\ y$ – текущий корень. В случае если $pp \rightarrow 0$, деление исходной функции производится на другое, отличное от близкого к нулю, циклически подсчитываемое выражение $ppp := ppp * eps$, где число шагов цикла совпадает с текущим значением порядка найденных нулей, а eps – наперед заданная погрешность вычислений. Сравнение найденных ранее нулей с вновь полученным выполняется по метрике Евклида, при этом ищется минимальное значение выражений как для некрatных

$$abs(sqr(korx - aax[l]) + sqr(kor\ y - aay[l])),$$

так и для кратных

$$abs(sqr(korx - bbx[l]) + sqr(kor\ y - bby[l]))$$

нулей. Необходимо отметить, что массивы aax , aay представляют собой априори заполненные действительные и мнимые части некрatных нулей; bbx , bby – соответственно заполняемые по ходу выполнения программы массивы кратных нулей. Если минимум метрики по всем таким сравнениям достаточно мал, то нуль повторился, его кратность увеличивается на единицу и производится деление функции на выражение pp или ppp с рекуррентным возвращением программы к началу, иначе запоминается кратность и выполняется переход к следующему фиксированному нулю до тех пор, пока не будет установлена кратность каждого.

Полюса комплексной функции $f(x, y) = g(x, y) + I \cdot p(x, y)$, квадрат модуля обратной к которой возможно представить в виде $|1/f(x, y)|^2 = u^2(x, y) + v^2(x, y)$ определяются как нули квадрата модуля обратной функций. В случае невозможности такого представления, на вход метода подается непосредственно функция $1/(g^2 + p^2)$, обратная к квадрату модуля, однако необходимо ослабить условия сужения окрестности при спуске [16]. Тем самым, осуществляется перенос изложенной схемы нахождения нулей на вычисление полюсов комплексной функции с учетом их порядка.

К недостаткам метода можно отнести длительность работы программы при определении кратных нулей и полюсов, который компенсируется параллелизмом изложенного метода. Так в его основе лежит сортировка, которая в па-

параллельном варианте имеет временную сложность $O(\log_2 n)$, ее можно заменить на сортировку с прямой и обратной адресностью сложности $O(1)$. Кроме того, на основе взаимной независимости рассматриваемого алгоритма по всем фрагментам в произвольно выбранном прямоугольном разбиении области, может быть организовано параллельное по данным фрагментам выполнение предложенного алгоритма.

По описанной схеме можно идентифицировать все нули разностного решения системы обыкновенных дифференциальных уравнений (ОДУ). Пусть вначале рассматривается задача Коши для одного уравнения

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0.$$

На промежутке $[x_0, x_n]$ вычисляется разностное приближение, для определенности выбран метод Эйлера:

$$y_{i+1} = y_i + f(x_i, y_i)h, \quad x_i = x_0 + ih, \quad i = 0, 1, \dots, n, \quad h = \frac{x_n - x_0}{n}.$$

Разностные значения принимаются за элементы массива

$$c[i] = y_{i-1}, \quad i = 1, 2, \dots, n,$$

который сортируется. Присоединение условия локализации минимумов к программе сортировки влечет устойчивую идентификацию нулей как минимумов модуля приближенного решения рассматриваемой задачи. Инвариантность схемы относительно длины промежутка достигается с использованием сортировки и повторной локализации на границах смещаемого промежутка.

Для случая системы дифференциальных уравнений схема применяется к каждому уравнению в отдельности. Для других разностных методов, в частности, для методов Эйлера-Коши и Рунге-Кутты, схема нахождения экстремумов аналогична. При этом удастся достигать значительной точности идентификации экстремальных значений при достаточно малом шаге.

В экспериментах [17] удастся достигать точности идентификации в рамках формата представления данных языка программирования, идентификация является «автоматической» – для ее осуществления достаточно ввести правую часть системы. Метод применим к системам произвольной размерности. Детальное описание и другие примеры на случай системы приводятся в [18].

Результаты (Results)

Известные методы вычисления экстремумов и нулей функций [10, 11, 19, 20] в своем большинстве базируются на градиентных схемах и не включают автоматическую идентификацию области каждого нуля (экстремума). Предложенный метод отличается от известных методов построением на основе сортировки, он включает автоматическую идентификацию области каждого нуля исходной функции и позволяет вычислять нули с учетом их кратности. На его основе возможно нахождение полюсов функций с учетом их порядка.

Для сравнения получим результаты вычисления нулей функций по описанной схеме на основе сортировки и в системах компьютерной математики MathCAD и Maple.

Пример 1. Найти нули полинома

$$P_n(z) = z^5 - 8z^4 + 25z^3 - 21z^2 - 15z + 10$$

Для вычисления нулей на основе сортировки использовалась программа описанного метода [14]. Для поиска нулей полинома в MathCAD используется схема метода секущих, реализованная функцией *polyroots(V)*, которая возвращает вектор всех нулей полинома степени n , где коэффициенты полинома находятся в векторе V , имеющем длину $n + 1$. В Maple нули полинома

$$f(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_0$$

ищутся как корни уравнения $P_n(\lambda) = 0$ следующим образом: вычисляются собственные значения матрицы Фробениуса

$$A_f = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & 0 & \dots & 0 & -a_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & -a_{n-1} \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix};$$

для этого матрица Фробениуса методом последовательных приближений переводится в диагональную матрицу, на главной диагонали которой расположены все нули исходного полинома.

В Maple этот алгоритм реализуется с помощью функции *solve*.

Результаты вычислений представлены в табл. 1.

Таблица 1. Сравнительная таблица вычисления нулей полинома на основе сортировки, в MathCAD и в Maple

Метод на основе сортировки		MathCAD		Maple	
Нули полинома	Значение полинома	Нули полинома	Значение полинома	Нули полинома	Значение полинома
4.95E-0001+i*0.00E+0000	1.88E-0037	0.495527831363	1.69E-0009	4.95E-0001	1.88E-0037
2.00E+0000+i*0.00E+0000	7.16E-0886	2.0000000000081	2.02E-0009	2.00E+0000	7.16E-0886
-6.92E-0001+i*0.00E+0000	3.00E-0036	- 0.6927497918342	1.42E-0009	-6.92E-0001	3.00E-0036
3.09E+0000+i*2.22E+0000	2.85E-0026	3.098610980194 +i*2.22802941174	2.27E-0009 +i*0.49E-0009	3.09E+0000+i*2.22E+0000	2.85E-0026
3.09E+0000-i*2.22E+0000	2.57E-0025	3.098610980194 -i*2.22802941174	2.27E-0009 -i*0.49E-0009	3.09E+0000-i*2.22E+0000	2.57E-0025

Табл. 1 показывает, что для данного примера точность вычислений нулей полинома описанным методом сравнима с точностью вычислений в Maple и превышает точность вычислений в MathCAD (практически это всегда так, если степень полинома выше пятой-шестой).

Пример 2. Найти нули функции

$$F := p \left(\cos p \cdot \sin \sqrt{\frac{1}{p + 2 \cdot 10^{-23}}} + 1 + \sin \frac{1}{p + 2 \cdot 10^{-23}} \right) \cdot \left(\frac{1}{\cos \left(\frac{1}{p + 1.2 + \sin p} \right)} \right),$$

где $p = (x - 1.000001)(x - 1.0000011)$.

Результаты вычислений представлены в табл. 2.

Табл. 2 показывает, что в MathCAD нули данной трансцендентной функции не найдены. В Maple приближенный результат удастся получить с помощью функции *fsolve*, которая используется для численного решения уравнений в тех случаях, когда трансцендентные уравнения не имеют аналитических решений. В результате найдено достаточно грубое приближение к нулям. Использование сортировки в этом случае дает точный результат с автоматической идентификацией всех нулей исходной функции.

Таблица 2. Сравнительная таблица вычисления нулей трансцендентной функции на основе сортировки, в MathCAD и в Maple

Метод на основе сортировки		MathCAD		Maple	
Нули полинома	Значение полинома	Нули полинома	Значение Полинома	Нули полинома	Значение полинома
1.000001E+0000	1.44E-0903	--	--	1.308077894	1.96E-0009
1.0000011E+0000	1.44E-0903	--	--	--	--

Для вычисления нулей функции двух переменных и в Maple, и в MathCAD отсутствуют специализированные функции. Поэтому нули функции возможно искать как минимумы ее модуля. В MathCAD для этого используется функция *minimize* (f, x, y), причем перед решением надо задать начальные значения искомых переменных (чем они ближе к верному решению, тем быстрее будет получен правильный результат). В Maple для поиска минимума функции используется команда *minimize* ($f, x = x1..x2, y = y1..y2, location$), где $[x1, x2] \times [y1, y2]$ – область поиска минимума, *location* – опция данной команды, позволяющая получить координаты точек минимума, то есть искомые нули.

Пример 3. Найти нули функции двух переменных

$$f(x, y) = x^2 - 3x + y^2 + 3y + 3$$

Результаты вычислений представлены в табл. 3.

Исходная функция имеет бесконечное множество нулей, поэтому Maple не дал результатов, строка вывода отображается в виде:

0, {[FAIL, 0]}.

Это означает, что минимум модуля исходной функции равен нулю, а координаты точек минимума неизвестны.

Столбец решений данного примера в MathCAD получен при различных начальных значениях переменных x, y . Однако результаты решения сильно зависят от выбора начальных значений переменных и имеют высокую погрешность вычислений.

Таблица 3. Сравнительная таблица вычисления нулей функции двух переменных на основе сортировки, в MathCAD и в Maple

Метод на основе сортировки		MathCAD		Maple	
Нули функции	Значение функции	Нули функции	Значение функции	Нули функции	Значение функции
$x=8.88E-0001$ $y=-2.56E+0000$	0.00E+0000	$x=1.65950126458145$ $y=-0.285685647670115$	0.32E-0009	--	--
$x=4.46E-0001$ $y=-2.12E+0000$	0.00E+0000	$x=1.26577088735596$ $y=-2.70213839319366$	0.63E-0008	--	--
...	...	...	...	--	--
$x=4.46E-0001$ $y=-8.74E-0001$	0.00E+0000	$x=0.936172287252808$ $y=-0.412756580408887$	0.56E-0008	--	--
$x=1.49E+0000$ $y=-2.72E+0000$	0.00E+0000	$x=0.304414215462631$ $y=-1.23434113712145$	0.38E-0009	--	--

Для разностных решений ОДУ сравнение предложенного метода [21] проведено также с методами пакета Maple и MathCAD на примерах поиска всех нулей решений ОДУ. Сравнение проводилось путем численного эксперимента на ряде следующих примеров.

Пример 4. Для уравнения вида $\frac{dy}{dx} = (0.8 - y^2) * \cos(x) - 0.4 * y$, $y(0) = 0$, в области $[0, 50]$ требуется идентифицировать все нули разностного решения. Для сопоставления используются пакеты MathCAD и Maple, результаты их работы представляют собой графическое отображение (рис. 2), представленное ниже.

Рис. 2. График решения ОДУ $\frac{dy}{dx} = (0.8 - y^2) * \cos(x) - 0.4 * y$

С другой стороны, поиск экстремумов разностного решения ОДУ данного примера на основе сортировки представлен в табл. 4 (соответствующая программа дана в [12]).

Таблица 4. Поиск всех нулей разностного решения в случае одного ОДУ

Метод	Шаг	точки нулей	Значения решения y
идентификация на основе сортировки	10^{-5}	2.57585	miny=0.00000
		5.81575	miny=0.00000
		8.94628	miny=0.00000
		12.08920	miny=0.00000
		15.23063	miny=0.00000
		18.37225	miny=0.00000
		21.51384	miny=0.00000
		24.65543	miny=0.00000
		27.79703	miny=0.00000
		30.93862	miny=0.00000
		34.08020	miny=0.00000
		37.22180	miny=0.00000
		40.36339	miny=0.00000
		43.50499	miny=0.00000
		46.64658	miny=0.00000
		49.78818	miny=0.00000

Таким образом, математические пакеты MathCAD, Maple не предоставляют численных значений нулей решений ОДУ, ограничиваясь лишь графическим представлением разностного решения. На графике можно затем идентифицировать нули, но без гарантии точности и с погрешностью графического редактора.

Ниже исследуется на нули каждая компонента решения системы ОДУ.

Пример 5. Для системы вида

$$\frac{dy_1}{dt} = y_2, \frac{dy_2}{dt} = -y_1 - 0.1 * y_2, y_1(0) = 0.1, y_2(0) = 1.$$

на промежутке $[0, 50]$ требуется найти все нули разностного решения. Результаты работы пакетов MathCAD и Maple формируются путем перевода системы в уравнение высшего порядка и представлены на рис. 3.

Рис. 3. График решения системы ОДУ

В табл. 5 приводятся результаты схемы идентификации нулей решения ОДУ.

Таблица 5. Поиск всех нулей разностного решения в случае системы ОДУ

Метод	Шаг	Точки нулей	Значения решения y_1	Точки Нулей	Значения решения y_2
идентификация на основе сортировки	10^{-5}	1.62237	Miny1=0.00000	3.14505	Miny2=0.00000
		4.76789	miny1=0.00000	6.29057	miny2=0.00000
		7.91342	miny1=0.00000	9.43467	miny2=0.00000
		11.05590	miny1=0.00000	12.57706	miny2=0.00000
		14.19829	miny1=0.00000	15.71945	miny2=0.00000
		17.34066	miny1=0.00000	18.86182	miny2=0.00000
		20.48305	miny1=0.00000	22.00421	miny2=0.00000
		23.62544	miny1=0.00000	25.14660	miny2=0.00000
		26.76782	miny1=0.00000	28.28898	miny2=0.00000
		29.91020	miny1=0.00000	31.43136	miny2=0.00000
		33.05259	miny1=0.00000	34.57375	miny2=0.00000
		36.19497	miny1=0.00000	37.71613	miny2=0.00000
		39.33736	miny1=0.00000	40.85852	miny2=0.00000
		42.47975	miny1=0.00000	44.00090	miny2=0.00000
		45.62213	miny1=0.00000	47.14329	miny2=0.00000
		48.76451	miny1=0.00000		

MathCAD и Maple ограничиваются лишь графическим представлением разностного решения дифференциальных уравнений, с их помощью нельзя идентифицировать нули всех компонент разностного решения. Ясно, что графическое отображение решения уравнения высшего порядка корректно только для несложных решений уравнений невысокой степени, усложнение графика влечет потерю точности идентификации нулей и экстремумов. График не дает возможности идентификации экстремумов и нулей нормы решений системы ОДУ. Необходимо принять во внимание, что для того, чтобы решение ОДУ отобразилась на графике в виде кривой в плоскости Oty , диапазон изменения переменной и масштаб координатных осей приходится подбирать вручную.

Существенно то неудобство графического представления экстремумов, в силу которого идентифицированные экстремумы недоступны для динамического программного использования, включая режим реального времени.

Обсуждение и заключение (Discussion and Conclusion)

На основе сравнения можно сделать следующий вывод. Предложенный метод дает автоматическую программную идентификацию области каждого нуля функции одной и двух переменных, включая нули комплексных полиномов и трансцендентных функций. Существующие схемы не обладают автоматической идентификацией области нулей или требуют начального приближения,

соответственно они игнорируют наличие нулей в некоторых сложных случаях, либо не достигают в этих случаях аналогичной точности вычислений [22]. Существующие схемы неустойчивы для поиска нулей трансцендентных функций, в отличие от предложенного метода. Помимо устойчивости метод обладает параллелизмом и распространяется на различные области математики и информатики.

На основе изложенной схемы автоматически программно идентифицируются все нули разностных решений ОДУ, причем в произвольно заданной области, входящей в область допустимых значений решения. Границы погрешности определяются свойствами устойчивости и характером сходимости используемых разностных схем [23]. Если условия сходимости и устойчивости разностной схемы обеспечены, то результаты идентификации экстремумов без дополнительного накопления погрешности представляются в численном виде, достигая точности формата представления данных.

Схема по построению отличается от известных методов исследования решений дифференциальных уравнений, в которых сортировка не используется в качестве базовой части вычислительной схемы.

Список используемых источников:

1. Маркушевич А.И. Краткий курс теории аналитических функций. М.: Наука, 1978. 416 с.
2. Гермогенова Т.А., Терентьев А.Г. Аналитические функции с кратными полюсами на бесконечности, их приложение в гидродинамике. // Труды Математического центра им. Лобачевского. Казань, 2001. Т. 8. С. 72-73.
3. Антонью А. Цифровые фильтры: анализ и проектирование. М.: Радио и связь, 1983. 320 с.
4. Ганеев Р.М. Математические модели в задачах обработки сигналов. М.: Телеком, 2002. 83 с.
5. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов: Пер. с англ. / Под ред. С.Я. Шаца. М.: Связь, 1979. 416 с.
6. Ромм Я.Е. Метод вычисления нулей и экстремумов функций на основе сортировки с приложением к поиску и распознаванию. I // Кибернетика и системный анализ. 2001. №4. С. 142-159.
7. Ромм Я.Е. Метод вычисления нулей и экстремумов функций на основе сортировки с приложением к поиску и распознаванию. II // Кибернетика и системный анализ. 2001. №5. С. 81-101.
8. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. СПб: Лань, 2002. 736 с.
9. Semerdziem Khr. Iteration methods for simultaneous finding all roots of generalized polynomial equations. Math. Balkan. 1994. № 4. P. 311-335.
10. Березин И.С., Жидков Н.П. Методы вычислений. Т. 2. М.: Государственное издательство физико-математической литературы, 1962. 640 с.
11. Тюшнякова И.А. Использование сортировки как основы метода выделения границы плоского изображения // Научные исследования в современном мире. Нефтекамск, 2015. С. 89-91.
12. Ромм Я.Е., Заика И.В. Схемы численной оптимизации на основе алгоритмов сортировки с приложением к идентификации экстремумов решений дифференциальных уравнений. Таганрог, 2010.
13. Ромм Я.Е., Соловьева И.А. Распараллеливаемый метод вычисления нулей многочленов в произвольной области комплексной плоскости / Депонированная рукопись № 210-B2005 14.02.2005.
14. Тюшнякова И.А. Разработка и исследование схем применения сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений. Диссертация на соискание ученой степени кандидата технических наук. Таганрог, 2006.
15. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. // Кибернетика и системный анализ. 2007. №1. С.165-182.
16. Ромм Я.Е., Тюшнякова И.А. Применение сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений. Таганрог, 2009.
17. Заика И.В., Ромм Я.Е. Метод нахождения экстремумов решений дифференциальных уравнений на основе адресной сортировки / Депонированная рукопись № 908-B2003 12.05.2003

18. Заика И.В. Разработка и исследование схем оптимизации на основе алгоритмов сортировки с приложением к идентификации экстремумов решений дифференциальных уравнений. Диссертация на соискание ученой степени кандидата технических наук. Таганрог, 2007.
19. Калиткин Н.Н. Численные методы. М.: Наука, 1978. 512 с.
20. Киреев В.И., Пантелеев А.В. Численные методы в примерах и задачах. М.: Высш. шк., 2004. 480 с.
21. Romm Y.E., Zaika I.V. Numerical sorting-based optimization as applied to general differential and nonlinear equations // *Cybernetics and Systems Analysis*. 2011. Т. 47. № 2. С. 316-329.
22. Ромм Я.Е., Тюшнякова И.А. Метод вычисления собственных значений матриц на основе сортировки в приложении к распознаванию изображений // *Известия высших учебных заведений. Северо-Кавказский регион. Серия: Технические науки*. 2006. № 1. С. 11-20.
23. Заика И.В., Ромм Я.Е. Численное решение нелинейных систем уравнений общего вида на основе алгоритмов сортировки // *Научные исследования в современном мире, Материалы Международной (заочной) научно-практической конференции. НИЦ «Наука и образование»; под общей редакцией Д.А. Ефремова. Нефтекамск, 2015. С. 78-80.*

References:

1. Markushevich A.I. *A brief course in the theory of analytic functions*. M.: Nauka, 1978. 416 p.
2. Germogenova T.A., Terentyev A.G. *Analytic functions with multiple poles at infinity, their application in hydrodynamics. Proceedings of the Mathematical Center. Lobachevsky. Kazan, 2001. V. 8. P. 72-73.*
3. Antonov *Digital Filters: Analysis and Design*. M.: Radio and Communications, 1983. 320 p.
4. Ganev R.M. *Mathematical models in problems of signal processing*. M.: Telecom, 2002. 83 p.
5. Oppenheim A.V., Schaffer R.W. *Digital Signal Processing: Trans. from English. / Ed. SY Schatz. M.: Communications, 1979. 416 p.*
6. Romm J.E. The method of calculation of zeroes and extrema of functions on the basis of sorting with application to search and recognition. I. *Cybernetics and Systems Analysis*. 2001. № 4. P. 142-159.
7. Romm J.E. The method of calculation of zeroes and extrema of functions on the basis of sorting with application to search and recognition. II. *Cybernetics and Systems Analysis*. 2001. № 5. P. 81-101.
8. Faddeev D.K., Faddeev V.N. *Computational methods of linear algebra*. St. Petersburg: Lan, 2002. 736 p.
9. Semerdziem Khr. *Iteration methods for simultaneous finding all roots of generalized polynomial equations. Math. Balkan*. 1994. №4. P. 311-335.
10. Berezin I.S., Zhidkov N.P. *Methods vychisleniy. T.2. M.: Gosudarstvennoe izdatelstvo physical and mathematical literature, 1962. 640 p.*
11. Tyushnyakova I.A. Using as a basis for sorting method of separating the border of the flat image. *Scientific research in the world today, Materials International (correspondence) scientific conference. SIC "Science and education"; edited by DA Ephraim. Neftekamsk, 2015. P. 89-91.*
12. Romm J.E. *Numerical optimization schemes based sorting algorithms with application to the identification of extrema solutions of differential equations. Taganrog, 2010.*
13. Romm J.E., Solovieva I.A. *Parallelizing method for calculating the zeros of polynomials in an arbitrary domain of the complex plane. Deposited manuscript number 210-V2005 14.02.2005.*
14. Tyushnyakova I.A. *Development and research of circuits use sorting to find zeros of functions and features of the application to the identification of flat images. thesis for the degree of candidate of technical sciences. Taganrog, 2006.*
15. Romm J.E. *Localization and stable calculation of zeros of a polynomial-based sorting. Cybernetics and Systems Analysis*. 2007. № 1. P. 165-182.
16. Romm J.E., Tyushnyakova I.A. *Application collation to find zeros of functions and features of the application to the identification of flat images. Taganrog, 2009.*
17. Zaika I.V., Romm J.E. *A method for finding extrema of solutions of differential equations based on the destination sorting. Deposited manuscript number 908-V2003 12.05.2003.*
18. Zaika I.V. *Development and research of optimization schemes based sorting algorithms with application to the identification of extrema solutions of differential equations. thesis for the degree of candidate of technical sciences. Taganrog, 2007.*
19. Kalitkin N.N. *Numerical methods*. M.: Nauka, 1978. 512 p.
20. Kireev V.I., Panteleev A.V. *Numerical methods in examples and problems*. M.: Higher. HQ., 2004. 480 p.
21. Romm Y.E., Zaika I.V. *Numerical sorting-based optimization as applied to general differential and nonlinear equations. Cybernetics and Systems Analysis*. 2011. Vol. 47. № 2. P. 316-329.

22. Romm J.E., Tyushnyakova I.A. *The method of calculating the eigenvalues of the matrices on the basis of the sort in the annex to the image recognition. Proceedings of the higher educational institutions. North Caucasus region. Series: Engineering. 2006. № 1. P. 11-20.*

23. Zaika I.V., Romm Ya.E. *Numerical solution of nonlinear systems of equations of the form-based sorting algorithms. Scientific research in the world today, Materials International (correspondence) scientific conference. SIC "Science and education"; edited by DA Ephraim. Neftekamsk, 2015. P. 78-80.*

© 2015, Заика И.В., Тюшнякова И.А.

*Программное определение нулей функций и
разностных решений систем обыкновенных
дифференциальных уравнений на основе сортировки*

© 2015, Zaika I.V., Tyushnyakova I.A.

*Software definition of functions zeroes and difference
solutions of systems ordinary differential equations on
the basis of the sort*