

## ПРОГРАММНАЯ ИДЕНТИФИКАЦИЯ НУЛЕЙ И ОСОБЕННОСТЕЙ ФУНКЦИЙ НА ОСНОВЕ СОРТИРОВКИ С ПРИЛОЖЕНИЕМ К ЦИФРОВОЙ ФИЛЬТРАЦИИ

*Я.Е. Ромм, И.А. Тюшнякова*

Таганрогский государственный педагогический институт,  
347936, Таганрог, ул. Инициативная, 48,  
тел. (863)(44) 60 1753, факс (863)(44) 60 5397, Romm@list.ru

На основе сортировки синтезируются схемы устойчивой локализации и вычисления нулей полиномов и функций с учетом их кратности. 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. С помощью сортировки строится схема приближенного вычисления полюсов комплексных функций с учетом порядка, алгоритмы приближенного вычисления вычетов функций в полюсах и, на этой основе, схема приближенного вычисления криволинейных интегралов по замкнутому контуру. Исследуется применимость метода для анализа цифровых фильтров с помощью обратного z-преобразования, для анализа временной функции линейной динамической системы и для анализа устойчивости дискретной цепи.

On the basis of sorting schemes of steady localization and calculation of zeroes of polynomials and functions with consideration of multiplicity are synthesized. Schemes differ from known ones by construction on the basis of sorting, by property of automatic localization of area all zeroes and each zero separately, by parallelism and by computing stability. By means of sorting the scheme of the approached calculation of residues of functions in poles is constructed and on this basis the scheme of the approached calculation of curvilinear integrals on the closed contour is constructed. Applicability of a method for the analysis of digital filters by means of inverse z-transformation and for the analysis of time function of linear dynamic system and for the analysis of stability of a discrete chain is investigated.

### Введение

Алгоритмы сортировки традиционно применяются в методах поиска, для моделирования операторных схем, находят разнообразное применение в системах программирования, операционных системах, системах обработки информации, управления базами данных, используются в базовых элементах компьютеров. Применение сортировки опирается на упорядочение информации и содержит только операции сравнения, исключая накопление погрешности. Отсюда возникает цель применить сортировку для построения схем вычислений, отличающихся снижением роста погрешности и максимальным упорядочением обрабатываемой информации.

Традиционный поиск нулей функций не основан на их одновременной локализации, – требуется указать область каждого нуля в отдельности. Кроме того, надежность вычислений зависит от выбора начального приближения и характера функции. По этим причинам не разработаны универсальные компьютерные способы нахождения нулей различных функций, в том числе полиномов с переменными коэффициентами.

В данном аспекте ставится вопрос о применении сортировки для локализации и устойчивого вычисления нулей полиномов и функций. Такая задача актуальна сама по себе, кроме того, на основе ее решения конструируется приближенное вычисление полюсов функций. Расчет нулей и полюсов функций – одна из важных задач цифровой обработки сигналов (ЦОС). Значения нулей и особенностей функций требуется знать при выполнении ЦОС в таких областях, как акустика, звуковая локация, радиолокация, сейсмология, связь, системы передачи данных, ядерная техника и др. Области применения ЦОС постоянно расширяются.

В работе предложен распараллеливаемый метод вычисления нулей полиномов и функций общего вида, инвариантный относительно области поиска нулей, устойчивость которого обусловлена применением сортировки. Нули автоматически локализуются с учетом их кратности. Приведены оценки временной сложности максимально параллельного варианта метода. Метод модифицирован для приближенного вычисления полюсов, вычетов функций и на этой основе распространяется на вычисление криволинейных интегралов по замкнутому контуру. Представлено применение метода для схем ЦОС. В качестве языка программирования используется объектно-ориентированный Pascal в среде Borland Delphi.

### Применение сортировки для нахождения экстремальных элементов последовательности

Под экстремальными элементами последовательности понимаются локально минимальные и максимальные элементы. Локально минимальный элемент последовательности в точности меньше предыдущего и не больше последующего элементов, локально максимальный элемент строго больше предшествующего, но не меньше последующего элементов.

Для нахождения экстремальных элементов последовательности используется адресная сортировка ее элементов. В качестве сортировки возможно использовать одну из двух параллельных внутренних сортировок по возрастанию ключей – подсчетом [1, 2] или слиянием [2, 3], или любую другую с прямой и обратной адресностью, смысл которых сводится к программной реализации взаимно однозначного соответствия между входными и выходными индексами сортируемого массива.

© Я.Е. Ромм, И.А. Тюшнякова, 2010

К выходу сортировки подсоединяется условный оператор, локализирующий все экстремальные элементы (далее экстремумы) среди этих значений [8, 4].

Оператор локализации минимума имеет вид  
 {Процедура сортировки} k:=1; while k<= nn0 do begin for l:=1 to k-1 do  
 if abs ( e [k]- e [k-1] ) <= eps0 then goto 22; writeln (c[k], e[k]);  
 22: k := k+1 end;

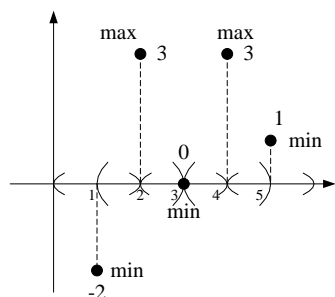
Оператор локализации максимума  
 {Процедура сортировки} k:=1; while k<= nn0 do begin for l := 1 to n - k do  
 if abs(e[k]-e[k+1]) <= eps0 then goto 222; writeln (c[k], e[k]);  
 222: k:=k+1; end.

В программных фрагментах (1), (2)  $nn0$  – число сортируемых элементов,  $eps0$  – константа, обозначающая окрестность локализации экстремумов, равная целому числу номеров влево и вправо от входного номера  $e[k]$ . Экстремум, локализуемый данными операторами, является локальным экстремумом среди ближайших значений последовательности, отсчитанных на  $eps0$  влево и вправо от  $e[k]$ . В частности, при  $eps0 = 1$  локализуется минимальный (максимальный) элемент среди двух ближайших слева и справа. При этом локализация означает конкретное значение как точки (индекса) экстремума  $e[k]$ , так и экстремального значения  $a[e[k]]$ .

Оператор локализации находит минимальные (максимальные) элементы последовательности по смыслу своего построения. Работая после сортировки, он для текущего элемента, определяемого индексом, записанным в  $e[k]$ , находит каждый элемент, в  $eps0$ -окрестности которого нет элементов, доставляющих значения элементов входной последовательности, предшествующие (последующие)  $a[e[k]]$  в отсортированном массиве. Среди элементов последовательности с равными значениями оператор локализации минимума (максимума) фиксирует элемент, соответствующий первому в порядке нумерации элементу отсортированного массива. Этот факт опирается на устойчивость используемой сортировки. Локализовать экстремальные элементы в рассматриваемом смысле предлагаемым способом можно только при сохранении обратных адресов на выходе сортировки, причем строго в том порядке, в каком располагаются отсортированные элементы.

**Пример 1.** Найти экстремальные элементы последовательности  $(-2, 3, 0, 3, 1)$ ,  $eps0 = 1$ .

Входной массив $a[k]$	-2	3	0	3	1
Адреса входных элементов $e[k]$	1	2	3	4	5
Отсортированный массив $a[e[k]]$	-2	0	1	3	3
Обратные адреса элементов	1	3	5	2	4



Результаты работы программы, листинг которой приведен в [12], отображены на рис. 1.

После сортировки массива  $(-2, 3, 0, 3, 1)$  с помощью циклического сравнения входных индексов, принадлежащих окрестности локализации экстремумов, идентифицируются локализуемые минимальные  $(-2, 0, 1)$  и максимальные  $(3, 3)$  элементы входной последовательности.

Рис. 1. Экстремальные элементы последовательности  $(-2, 3, 0, 3, 1)$

### Подход к локализации и вычислению нулей полиномов с учетом кратности на основе сортировки

Рассмотрим полином  $P_n(z) = \sum_{\ell=0}^n a_{\ell} z^{\ell}$ , у которого могут быть комплексные нули и коэффициенты ( $z = x + y \cdot I$ ,  $I$  – мнимая единица). Для формирования функции  $f(x, y)$ , которая подается на вход метода, необходимо выделить действительную и мнимую части рассматриваемого полинома. Для этого осуществляется построение функции двух действительных переменных  $f(x, y) = |P_n(z)|^2$  с применением схемы Горнера. Ее программная реализация [5] для комплексных коэффициентов и операций имеет вид:  
 FUNCTION func (VAR x,y: extended; var bdv, bmv: vect3): extended;  
 VAR i1: 0..n1; p1,p2,pp1,pp2,d1,d2,d3,d4:extended;  
 BEGIN p1:=bdv[n1];p2:=bmv[n1]; FOR i1:=1 TO n1 DO begin  
 um(p1,p2,x,y,d1,d2); pp1:=bdv[n1-i1]; pp2:=bmv[n1-i1]; sum(pp1,pp2,d1,d2,d3,d4); p1:=d3; p2:=d4; end;  
 func:=sqrt(p1)+sqrt(p2); END;

Вышеиспользуемые процедуры *um* и *sum* реализуют соответственно комплексное умножение и сложение двух комплексных чисел. На выходе схемы Горнера получаются действительная и мнимая часть комплексного значения полинома. Функция  $func(x, y, bdv, bmv)$ , – представляет сумму квадратов действительной и мнимой части этого значения (в математической записи  $f(x, y) = (\text{Re}(P_n(z)))^2 + (\text{Im}(P_n(z)))^2$ ).

**Замечание 1.** Для формирования входной функции возможно использовать бином Ньютона для возведения в степень  $z^l = (x + y \cdot I)^l$ , коэффициенты бинома определяются по треугольнику Паскаля, затем приводятся подобные при действительной и мнимой части. После умножения  $P_n(z)$  на комплексно-сопряженное значение нахождение нулей  $P_n(z)$  сводится к нахождению нулей функции двух действительных переменных  $f(x, y) = |P_n(z)|^2$ .

Вводится равномерная прямоугольная сетка, заключающая внутри себя область всех нулей  $func(x, y, bdv, bmv)$ . Значения функции берутся во всех узлах сетки, образуя двумерный массив, и среди его элементов ищутся все возможные минимумы. Схема поиска нулей полиномов подробно описана в [1, 6] и включает следующие процедуры: построение сетки, процедуру минимизации по строкам в текущем столбце, процедуру сортировки найденных минимумов, локализацию абсциссы и ординаты точки минимума и в итоге спуск к наименьшему значению в окрестности локализованной точки.

Следует отметить, что алгоритм локализации всех комплексных нулей полиномов в  $eps0$ -окрестности без применения спуска к локализованному нулю осуществляет их одновременное вычисление с границей абсолютной погрешности  $eps0$  в евклидовой метрике пространства  $R_2$  [5]. При этом значение  $eps0$  может быть задано априори. В этом случае рассматриваемый алгоритм, локализуя значение нуля, всегда вычисляет его с априори заданной границей абсолютной погрешности  $eps0$  в метрике евклидова пространства  $R_2$ , что существенно отличает данный алгоритм от известных алгоритмов локализации.

**Замечание 2.** Особенности перехода от одного фрагмента области поиска нулей (и экстремумов) к другому рассмотрены в [7], там же предложены инвариантные относительно вида исследуемой функции, размеров области и числа фрагментов ее разбиения способы идентификации нулей (и экстремумов).

Распространение схемы вычисления нулей полиномов на случай определения кратности найденных нулей достигается рекуррентным повторением приводимого далее алгоритма [5], для простоты описания приведенного для действительного случая.

**Алгоритм идентификации кратности нулей полинома:**

1. Найти все нули исходного полинома  $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  с помощью изложенного метода. Если количество найденных нулей полинома  $P_n(x)$  меньше чем его степень  $n$ , то среди нулей имеются кратные, иначе все нули – не кратные. При наличии кратных нулей перейти к пункту 2, иначе конец алгоритма.

2. Выполнить деление исходного полинома на линейный двучлен  $x - x_1$ , где  $x_1$  – первый из нулей полинома  $P_n(x)$ . После деления получаем новый полином  $G_{n-1}(x) = b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \dots + b_1 x + b_0$  степени  $n - 1$ , коэффициенты которого вычисляются путем сравнения коэффициентов при равных степенях полиномов  $P_n(x)$  и  $G_{n-1}(x) \times (x - x_1)$  по следующим формулам (схема Горнера):

$$b_{n-1} = a_n, \quad b_{n-2} = a_{n-1} + b_{n-1} x_1, \quad \dots, \quad b_0 = a_1 + b_1 x_1. \quad (3)$$

Перейти к пункту 3.

3. Заново по программе локализации и вычисления нулей на основе сортировки выполнить поиск всех нулей полинома  $G_{n-1}(x)$ . Если среди них не оказалось значения  $x_1$ , то этот нуль не является кратным. В этом случае все операции, включая схему (3), повторяются для нуля  $x_2$ . Если же среди найденных нулей оказалось значение  $x_1$ , то перейти к пункту 4.

4. Выполнить деление по схеме (3) полинома  $G_{n-1}(x)$  на линейный двучлен  $x - x_1$  с тем, чтобы выполнить проверку на наличие в нем нуля  $x_1$ . Процесс продолжается до исчерпания кратности нуля  $x_1$ . Перейти к пункту 5.

5. Выполняется аналогичная проверка на кратность каждого из оставшихся нулей  $x_2, x_3, \dots, x_n$  исходного полинома  $P_n(x)$ .

6. Конец алгоритма.

**Замечание 3.** Для нахождения нулей трансцендентных функций с учетом кратности необходимо использовать метод, подробно описанный в [3, 4]. В основе его идеи лежит деление функции, нули которой уже найдены, на циклически подсчитываемое выражение  $pp := pp * ((x - korx)^2 + (y - kory)^2)$ , где число шагов цикла равно текущему счетчику кратности, а  $korx + i * kory$  – текущий корень. В случае если  $pp \rightarrow 0$ , деление 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,  $eps$  – наперед заданная погрешность вычислений. Алгоритм и программная реализация данного метода приведены в [8].

Изложенное использование сортировки позволяет обойти математическую трудность решения классической задачи вычисления нулей, связанную с отысканием областей, в которых располагаются единственные нули; с поиском первого приближения к нулю; с необходимостью в дифференцировании полинома, кратность нулей которого требуется вычислить.

При решении различных практических задач часто отсутствует априорная информация о границах области, охватывающей нули полинома. Поэтому возникает задача программной локализации области нахождения всех нулей полинома с их одновременным вычислением.

**Замечание 4.** Возможно распространение изложенной схемы нахождения нулей на случай области на комплексной плоскости, границы которой не являются фиксированными. Переход к области с произвольно заданными границами, которые, в частности, могут не примыкать к области всех нулей полинома, напротив, они могут быть отделены от этой области произвольно большим расстоянием, основывается на следующих соображениях. Вне области нулей полинома  $P_n(z)$ , квадрат его модуля, т.е. функция  $f(x, y)$ , фактически оказывается монотонной по каждой переменной в отдельности при фиксировании другой переменной в качестве локализованного приближения. На основе достигаемой таким способом монотонности, при фиксировании одной локализованной координаты, из любых двух значений модуля полинома, взятых слева (справа), сверху (снизу) от области всех нулей полинома, меньшее значение всегда ближе к этой области. Можно взять произвольно большой квадрат и сужать его до момента обнаружения хотя бы одного приближения нулевого минимума. При этом локализацию и вычисление каждого минимума модуля полинома можно осуществлять на основе вышеописанной схемы для ограниченной области комплексной плоскости. Работоспособность схемы можно гарантировать для полиномов произвольной степени с комплексными коэффициентами, имеющих как кратные, так и не кратные нули, включая случай их плохой отделенности. Значение полинома не должно выходить из диапазона представления чисел в конкретном языке программирования.

Длительность работы программы, реализующей алгоритм нахождения комплексных нулей полиномов с учетом кратности, компенсируется параллелизмом изложенного метода. На основе взаимной независимости рассматриваемого алгоритма по всем фрагментам в произвольно выбранном прямоугольном разбиении области произведена оценка параллельного по данным фрагментам выполнения предложенного алгоритма [8]. Для оценки временной сложности (времени выполнения) параллельного алгоритма использована модель неветвящейся параллельной программы без учета обмена. В результате, полное время параллельного вычисления всех нулей полинома  $P_n$  оценивается из соотношения:

$$T(R) = O\left(\log_{\rho} \frac{\text{eps}0}{\text{eps}}\right) \times O(\log_2 n) \times O(np), \text{ число процессоров } R = O\left(\frac{(x_{11} - x_{00})(y_{11} - y_{00})}{hh^2} \times \frac{n^5}{4}\right) \times O(np).$$

Здесь  $[x_{00}, x_{11}] \times [y_{00}, y_{11}]$  – область поиска нулей;  $hh$  – длина текущего промежутка, на котором формируется равномерная сетка с шагом  $h$ ;  $n$  – число узлов равномерной сетки;  $\text{eps}0$  – радиус окрестности локализации нуля, включающий не больше половины числа узлов, отсчитываемых вдоль оси  $OX$  между ближайшими друг к другу нулями полинома;  $\text{eps}$  – наперед заданная граница абсолютной погрешности вычислений;  $np$  – степень полинома;  $\rho$  – коэффициент пропорции, в которой сокращается диаметр окрестности текущего приближения нуля при спуске.

Данные оценки соответствуют максимально параллельному варианту метода, показывая потенциал быстродействия рассмотренной схемы. При распараллеливании временная сложность схемы не будет критичной в сравнительно больших областях поиска нулей, включая случай их плохой отделенности.

### Локализация и приближенное вычисление нулей в окрестности единичной окружности

На основе описанной схемы возможно производить поиск нулей в произвольной области комплексной плоскости, границами которой является не только некоторый прямоугольник или квадрат, но и произвольная кривая. Рассмотрим в качестве области поиска окрестность окружности единичного радиуса.

Для удобства передвижения по контуру в данном случае целесообразно перейти к полярной системе координат  $\begin{cases} x = \rho \cos \varphi \\ y = \rho \sin \varphi \end{cases}$ , где  $\rho = 1$ . Модификация схемы заключается в следующем. После определения точки,

находящейся на контуре, она окружается квадратом со стороной  $2 \cdot \text{epsilon}$ , где  $\text{epsilon}$  – произвольно заданное значение, определяющее близость границы области поиска по отношению к контуру. В построенном квадрате поиск нулей осуществляется по вышеописанной схеме без каких-либо изменений, причем равномерная сетка строится слева направо сверху вниз, начиная с нижней левой вершины квадрата. После этого выполняется переход к следующей точке контура таким образом, чтобы полученные квадраты пересеклись. Данный процесс осуществляется до тех пор, пока весь исходный контур не будет покрыт цепочкой квадратов.

Для случая окружности в качестве первоначальной точки контура выбирается точка с нулевым аргументом  $\text{phi} = 0$ . Она окружается квадратом  $x_0 := \cos(\text{phi}) - \text{epsilon}$ ;  $y_0 := \sin(\text{phi}) - \text{epsilon}$ ; сторона которого  $2 \cdot \text{epsilon}$ , где  $\text{epsilon} := \text{eps}0 * 20$ , причем основные константы сохраняют вышеописанный смысл:  $\text{eps}0$  – константа, не превышающая половины числа узлов, отсчитываемых вдоль оси  $OX$  между ближайшими друг к

другу нулями полинома;  $n=1024$  – число узлов равномерной сетки;  $h:=2*\epsilon/n$ ; – шаг равномерной сетки. После нахождения нулей в текущем квадрате переход к следующему осуществляется с помощью операторов:  
 $\phi:=\phi+\pi/8$ ; if  $\phi < 2*\pi$  then begin  $x0:=\cos(\phi)-\epsilon$ ;  $y0:=\sin(\phi)-\epsilon$ ; goto 14;...  
 Описанным образом удастся отыскать все значения нулей, находящиеся в заданной области, включающей единичную окружность.

**Пример 2.** В окрестности единичной окружности найти все нули полинома, квадрат модуля которого задается выражением:

$$|P_n(z)|^2 = \prod_{i=1}^5 (\text{sqr}(x - b0[i]) + \text{sqr}(y - b01[i])).$$

В  $b0[i]$  и  $b01[i]$  хранятся значения действительной и мнимой частей  $i$ -го нуля многочлена, заданные в разделе описания констант ( $b0: \text{vek} = (1, -0.99, 0.5, -0.3, 1)$ ;  $b01: \text{vek} = (0.01, 0.01, 0.85, -0.95, 1)$ );).

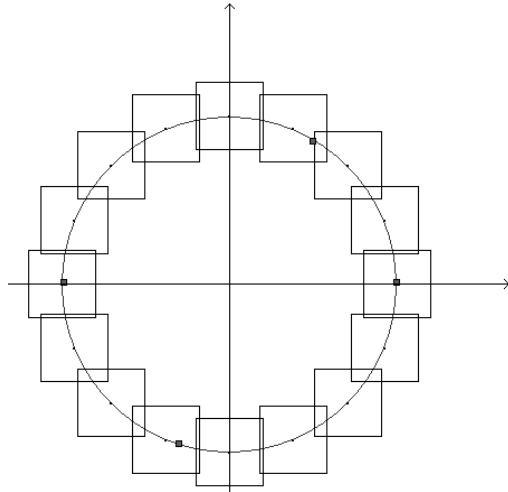


Рис. 2. Иллюстрация схемы поиска нулей в окрестности заданной окружности

Результаты работы программы приведены в табл. 1. Рис. 2 иллюстрирует программный поиск нулей в окрестности окружности по описанной схеме. Найденные нули отображены темными маркерами. Очевидно, что один из нулей полинома – (1, 1) не удовлетворяет заданной области поиска и поэтому не был найден.

Таблица 1. Результаты работы программы вычисления нулей полинома в окрестности единичной окружности

Действительная часть нуля	Мнимая часть нуля	Значение функции
1	0,01	0.00E+0000
0,5	0,85	0.00E+0000
-0,99	0,01	0.00E+0000
-0,3	-0,95	0.00E+0000

Описанный метод позволяет искать нули функций вблизи практически любых кривых на плоскости, если дополнительно использовать известные алгоритмы выделения контура (отслеживающие, сканирующие).

### Вычисление полюсов комплексной функции с учетом порядка на основе сортировки

Изложенный метод вычисления нулей ниже модифицируется для автоматической программной идентификации полюсов комплексных функций с учетом их порядка. Полюсы исходной 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 функции. 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 [8]. Тем самым осуществляется перенос схемы нахождения нулей, изложенной выше, на вычисление полюсов комплексной функции с учетом их порядка.

**Пример 3.** В прямоугольнике  $[-0.5, 0.5] \times [-1.5, 1.5]$  найти полюсы функции  $f(z) = \frac{z}{(z^2 + 1)^2}$  и определить их порядок.

Результаты работы программы [8] приведены в табл. 2.

Таблица 2. Результаты работы программы вычисления полюсов функции

Действительная часть полюса	Мнимая часть полюса	Порядок полюса	Значение обратной функции
-6.72E-0011	-1.00E+0000	2	0.00E+0000
-6.72E-0011	1.00E+0000	2	0.00E+0000

Схема распространяется на вычисление вычетов комплексных функций, имеющих в качестве особенностей полюсы, с последующим вычислением криволинейных интегралов по замкнутому контуру.

Вычет  $res f(z) = C_{-1}$  исходной функции  $f(z)$  в полюсе  $z_0$  определяется при последовательном вычислении всех отрицательных коэффициентов ряда Лорана  $C_{-n} = \lim_{z \rightarrow z_0} f(z)(z - z_0)^n$ ,

$(C_{-n+1} = \lim_{z \rightarrow z_0} \frac{f(z)(z - z_0)^n - C_{-n}}{z - z_0}, \dots)$  априори заданной границей абсолютной погрешности [8]. Эти

коэффициенты вычисляются при повторном применении схемы вычисления нулей в месте спуска к наименьшему значению в окрестности локализованной точки по ходу циклического приближения значения  $z$  к заново вычисляемому значению  $z_0$  при существенном сужении области поиска вокруг найденного значения полюса. За количественную меру сужения окрестности области поиска отвечает специально введенный коэффициент  $g \in (10^{-424}, 10^{-9})$ , определяющий зависимость между константой  $eps0$  и граничными значениями области поиска:  $eps0 = zz * g$ ;  $x00 = x0 - eps0/g$ ;  $x11 = x0 + eps0/g$ ;  $y00 = y0 - eps0/g$ ;  $y11 = y0 + eps0/g$ ;  $z_0 = x_0 + i * y_0$  – полюс, в котором вычисляется вычет.

Значение множителя  $zz$  константы  $eps0$  подобрано эвристически в ходе численного эксперимента, оно варьируется в зависимости от порядка найденных полюсов, однако выбранные значения  $zz$  [8] остаются постоянными для различных функций, имеющих одинаковый порядок полюсов. Коэффициент  $g$  принимает произвольные значения из указанного интервала, остальные же константы программы [8] находятся в прямой или обратной пропорции к  $g$  и  $eps0$ .

На основе сортировки, с использованием метода нахождения вычетов функций в полюсах возможно вычисление криволинейных интегралов по замкнутому контуру. В силу основной теорема о вычетах –

$$\oint_{\partial D} f(z) dz = 2\pi i \sum_{k=1}^n res f(z), \quad z_k \in D, \text{ где } D \text{ – односвязная область в комплексной плоскости.}$$

Определив вычеты исходной функции, непосредственно можно перейти к вычислению интеграла по замкнутому контуру. Этот переход возможен в случае полюсов, находящихся внутри области  $D$ , поэтому необходимо программное представление этой области. Представляется возможным также нахождение всех полюсов исходной функции и последующий выбор тех из них, которые входят в область, ограниченную криволинейным контуром.

### Схемы анализа цифровых фильтров с использованием сортировки

При моделировании линейных динамических систем, а также при фильтрации или обработке сигналов проектируются и применяются цифровые фильтры. Термин цифровой фильтр [9] относится к процессу вычислений по некоторому алгоритму, с помощью которого дискретный сигнал преобразуется в последовательность чисел, выражающих выходной сигнал.

Основным математическим аппаратом для анализа проектирования цифровых фильтров, является  $z$ -преобразование [9], которое отображает дискретизированные сигналы из временной области в их алгебраическое представление в области  $z$ , что значительно упрощает анализ систем с дискретизированными сигналами.

Двустороннее  $z$ -преобразование дискретной функции  $f(nT)$  определяется как  $F(z) = \sum_{n=-\infty}^{\infty} f(nT)z^{-n}$  для

всех  $z$ , для которых ряд сходится. Одностороннее  $z$ -преобразование дискретной функции  $f(nT)$

представляется в виде  $F_1(z) = \sum_{n=0}^{\infty} f(nT)z^{-n}$ . По отношению к  $F(z)$  функция  $f(nT)$  называется обратным

$z$ -преобразованием функции  $F(z)$ ,  $f(nT)$  и используется для получения соответственной временной функции и определяет коэффициенты дискретного сигнала, отвечающего этой функции.

В силу сходимости ряда функция  $f(nT)$  однозначно определяется соотношением

$$f(nT) = \frac{1}{2\pi i} \oint_{\Gamma} F(z) z^{n-1} dz, \text{ где } \Gamma \text{ – контур, охватывающий все полюсы функции } F(z) \text{ [9]. Поэтому}$$

$$f(nT) = \sum_{i=1}^k res[F_0(z)], \text{ где } F_0(z) = F(z)z^{n-1}. \quad (4)$$

На этой основе описанный метод нахождения вычетов с помощью сортировки распространяется на программное вычисление обратного  $z$ -преобразования.

Предложенный подход иллюстрирует следующий пример.

**Пример 4** [1]. Задано  $z$ -преобразование вида  $X(z) = \frac{z+1}{z}$ . Найти коэффициенты  $\{x_n\}$  дискретного сигнала, отвечающего этой функции.

Заметим, что функция  $X(z)$  аналитична на всей комплексной плоскости за исключением точки  $z = 0$  (полус  $X(z)$ ), следовательно, она может описывать  $z$ -преобразование. Для нахождения коэффициентов дискретного сигнала необходимо вычислить обратное  $z$ -преобразование  $f(nT)$  из (4). Поскольку речь идет об одностороннем  $z$ -преобразовании, то  $n = 0, 1, 2, \dots$ .

В качестве области поиска полюсов функций  $F_0(z) = F(z)z^{n-1}$  выберем квадрат  $[-1, 1] \times [-1, 1]$ . Следует отметить, что для поиска полюсов можно пользоваться программой [8], модифицированной для области на комплексной плоскости с границами, которые не являются фиксированными.

Результаты работы соответствующих программ представлены в табл. 3.

Мнимая часть полученных вычетов достаточно мала (порядка  $10^{-16}$ ), приближенно ее можно считать нулем (если полюса функций вычисляются с точностью  $10^{-15}$ , можно пользоваться программой нахождения вычетов для действительного случая).

Каждая из функций  $F_0(z) = F(z)z^{n-1}$  (где  $n = 0, 1, 2, \dots$ ) имеет не более одного полюса.

Таблица 3. Результаты работы программы вычисления обратного  $z$ -преобразования функции  $X(z) = \frac{z+1}{z}$

	Вид правой части входной функции	Полюс	Порядок полюса	Вычет в полюсе	Приближенное значение обратного $z$ -преобразования
0	$\frac{x^2 + 2x + y^2 + 1}{x^4 + 2x^2y^2 + y^4}$	8.52E-0016+ +i*8.52E-0016	2	1.00E-0000+ +i* 8.78E-0016	1
1	$\frac{x^2 + 2x + y^2 + 1}{x^2 + y^2}$	8.52E-0016+ +i*8.79E-0016	1	9.99E-0001+ +i* 8.78E-0016	1
2	$x^2 + 2x + y^2 + 1$	–	–	0	0
3	$x^4 + 2x^3 + 2x^2y^2 + x^2 + 2y^2x + y^4 + y^2$	–	–	0	0

В результате коэффициенты дискретного сигнала имеют вид  $\{1, 0, 0, \dots\}$ .

Предложенный метод также применим для анализа временных функций в зависимости от расположения нулей и полюсов на комплексной плоскости. Полюсы внутри круга единичного радиуса, имеющие  $|z| < 1$ , соответствуют затухающим сигналам и наоборот. Полюс на положительной действительной оси соответствует дискретизированной действительной экспоненте. Полюс на отрицательной действительной оси соответствует дискретизированному гармоническому сигналу с двумя отсчетами на период. Сигнал с четырьмя интервалами дискретизации на период соответствует полюсу на мнимой оси. Для гармонических сигналов моменты дискретизации связаны с нулями  $z$ -преобразования.

Таким образом, программная идентификация нулей и полюсов  $z$ -преобразования позволяет сделать вывод о характере временной функции линейной динамической системы.

Метод вычисления полюсов функции на основе сортировки применим к анализу устойчивости дискретной цепи, которая считается неустойчивой, если ограниченное по амплитуде входное воздействие вызывает на ее выходе бесконечно нарастающий отклик. Наоборот, дискретная цепь устойчива, когда отклик на ограниченное воздействие также ограничен [9].

Передаточной или системной функцией называют  $z$ -преобразование импульсной характеристики. Передаточная функция характеризует поведение системы в зависимости от частоты синусоидального воздействия. Известно, что в устойчивой аналоговой цепи полюсы передаточной функции располагаются в левой полуплоскости переменной  $p$ . При переходе от аналоговой цепи к дискретной и замене преобразования Лапласа  $z$ -преобразованием точки левой полуплоскости  $p$ -плоскости переходят в точки, лежащие внутри единичной окружности  $z$ -плоскости. Таким образом, полюсы передаточной функции устойчивой дискретной цепи располагаются внутри единичной окружности  $z$ -плоскости.

На основе метода с применением сортировки всегда оказывается возможным по виду передаточной функции дискретной цепи определить устойчива или нет данная дискретная цепь.

**Пример 5.** Цифровой фильтр Баттерворта нижних частот 3-го порядка описывается передаточной функцией

$$H_1(z) = \frac{0.1317}{1 - 1.785z^{-1} + 1.202z^{-2} - 0.2853z^{-3}}$$

На вход метода подается полином, заданный своими коэффициентами  $(-0.2853, 1.202, -1.785, 1)$ . Результаты работы программы приведены в табл. 4.

Таблица 4. Результаты работы программы вычисления полюсов функции  $H_1(z)$

Действительная часть полюса	Мнимая часть полюса	Порядок полюса	Значение обратной функции
5.35E-0001	9.57E-0445	1	2.08E-0890
6.24E-0001	-3.77E-0001	1	1.78E-0022
6.24E-0001	3.77E-0001	1	1.82E-0022

Расположение полюсов в плоскости  $z$  показано на рис. 3, а. Здесь показана структурная схема такого фильтра (рис. 3, б). Цепь устойчива, поскольку все полюсы передаточной функции лежат внутри единичной окружности.

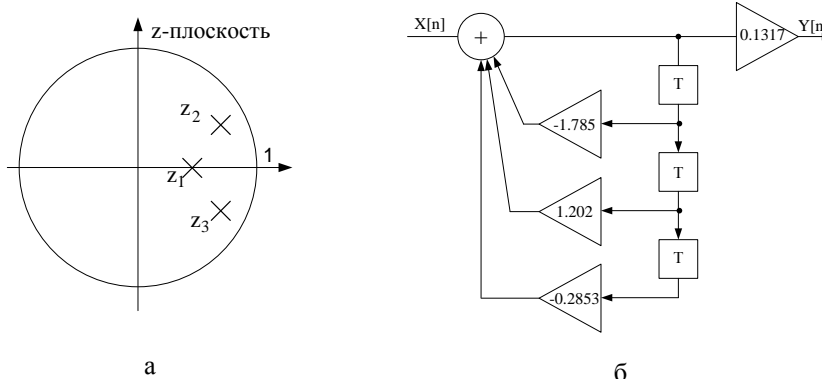


Рис. 3. Расположение полюсов передаточной функции, описывающей фильтр Баттерворта

**Замечание 5.** Для того чтобы обойти некоторые алгебраические преобразования, связанные с приведением к общему знаменателю компонент передаточной функции, возможно использовать следующее положение об устойчивости цифрового фильтра. Его называют устойчивым, когда полюсы передаточной функции расположены вне окружности  $|z^{-1}| = 1$  [9].

### Сравнение метода вычисления нулей и полюсов функций на основе сортировки с существующими методами

Известные методы вычисления экстремумов и нулей функций [10–13] 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, включает автоматическую программную 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, вычетов функций в полюсах и, как следствие, криволинейных интегралов по замкнутому контуру. Метод практически применим для анализа цифровых фильтров.

Следует отметить, что предложенный метод использования сортировки с последующим условием локализации экстремумов затруднительно заменить другим приемом. В ходе численного эксперимента нули функций вычислялись на основе непосредственного выделения минимальных значений путем сравнения текущего значения с соседними. В результате применения данной схемы обнаружилось, что она неустойчива в случае поиска нулей сложных функций комплексной переменной, что иллюстрирует следующий пример.

**Пример 6.** Вычислить нули функции

$$F := p \left( \cos p \cdot \sin \sqrt{\frac{1}{p + eps}} + 1 + eps0 + \sin \frac{1}{p + eps} \right) \cdot \left( \frac{1}{\cos \left( \frac{1}{p + 1.2 + \sin p} \right)} \right) \cdot \left( 2 + \sin \frac{1}{p + eps} \right)^2,$$

в квадрате  $[x00, x11] \times [y00, y11]$ ,

где  $eps=1.1e-444$ ;  $eps0=0.00000000000000000029*1e-286$ ;  $x00=-0.00000000000000000089*1e-287$ ;

$x11=0.00000000000000000084*1e-287$ ;  $y00=-0.00000000000000000089*1e-287$ ;

$y11=0.00000000000000000084*1e-287$ ;

$p$  конструируется в виде  $-p = p * \left( -b[i]^2 + -b1[i]^2 \right)$ , а в массивы  $b[i]$  и  $b1[i]$  записываются значения, заданные в разделе описания констант:

b: vect3 =(0.00000000000000000101\*1e-287, 0.00000000000000000203\*1e-287, 0.00000000000000000302\*1e-287, 0.00000000000000000401\*1e-287, 0.00000000000000000505\*1e-287, 0.00000000000000000602\*1e-287, 0.00000000000000000701\*1e-287, 0.00000000000000000805\*1e-287);  
 b1: vect3 =(0.00000000000000000107\*1e-287, 0.00000000000000000203\*1e-287, 0.00000000000000000309\*1e-287, 0.00000000000000000404\*1e-287, 0.00000000000000000503\*1e-287, 0.00000000000000000603\*1e-287, 0.00000000000000000702\*1e-287, 0.00000000000000000806\*1e-287);

Табл. 5 с очевидностью показывает неустойчивость схемы вычисления нулей без использования сортировки.

Таблица 5. Результаты работы программ определения комплексных нулей функции  $F(x, y)$  с использованием сортировки и без ее использования

Схема на основе сортировки			Схема без использования сортировки и условия локализации		
Действительная часть нуля	Мнимая часть нуля	Значение функции	Действительная часть нуля	Мнимая часть нуля	Значение функции
2.03E-0306	2.03E-0306	0.00E+0000	1.01E-0306	1.07E-0306	0.00E+0000
5.05E-0306	5.03E-0306	0.00E+0000	—	—	—
4.01E-0306	4.04E-0306	0.00E+0000	—	—	—
3.02E-0306	3.09E-0306	0.00E+0000	—	—	—
6.02E-0306	6.03E-0306	0.00E+0000	—	—	—
7.01E-0306	7.02E-0306	0.00E+0000	—	—	—
8.05E-0306	8.06E-0306	0.00E+0000	—	—	—
1.01E-0306	1.07E-0306	0.00E+0000	—	—	—

Проведем сравнение предложенной схемы с математическими методами.

Поведение нулей полиномов и функций исследуется в алгебре и математическом анализе [14, 15]. Схемы численного решения задачи опираются на алгебраические преобразования [10, 15] и методы последовательных приближений [13, 16]. В своем большинстве известные методы 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 специфика компьютерных вычислений. В этом можно видеть одну из причин, по которой компьютерная реализация методов нахождения нулей полиномов и функций общего вида в существенных аспектах не гарантирует завершения решения. С другой стороны, при наличии возможности программной реализации некоторых схем, вообще говоря, не исключается проблема локализации, накопления погрешности и вычислительной неустойчивости [10, 16]. В этом контексте выделяются классические методы Штурма и Бюдана – Фурье [15], однако они относятся строго к случаю локализации действительных нулей полинома, не распространяясь на случай комплексных нулей и нулей трансцендентных функций.

В отличие от известных математических методов предложенный метод на основе сортировки дает программную идентификацию области всех нулей и каждого нуля в отдельности (с одновременным вычислением с точностью до  $\epsilon$ ), фактически не используя при этом ограничения на вид входной функции. Требуется только, чтобы функция была непрерывной и чтобы дискретизация не приводила к существенным изменениям рельефа функции.

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

**Пример 7.** Найти нули функции

$$F := p \left( \cos p \cdot \sin \sqrt{\frac{1}{p + 2 \cdot 10^{-23}}} + 1 + \sin \frac{1}{p + 2 \cdot 10^{-23}} \right) \cdot \left( \frac{1}{\cos \left( \frac{1}{p + 1.2 + \sin p} \right)} \right), \text{ где } p = (x - 1.000001)(x - 1.000001).$$

Результаты вычислений приведены в табл. 6, которые показывают, 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 6. Сравнительная таблица вычисления нулей трансцендентной функции 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
1.000001E+0000	1.44E-0903	—	—	1.308077894	1.96E-0009
1.0000011E+0000	1.44E-0903	—	—	—	—

**Пример 8.** Найти нули функции двух переменных  $f(x, y) = x^2 - 3x + y^2 + 3y + 3$

Таблица 7. 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	—	—

Результаты вычислений представлены в табл. 7. 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 не известны. Столбец решений данного примера в Mathcad получен при различных начальных значениях переменных  $x, y$ . Однако результаты решения существенно зависят от выбора начальных значений переменных и имеют высокую погрешность вычислений.

На основе сравнения можно сделать следующий вывод. Предложенный метод дает программную идентификацию области каждого нуля функции одной и двух переменных, включая случай нулей комплексных полиномов и трансцендентных функций, при этом достигается точность вычислений в максимальном формате представления данных. Такая идентификация инвариантна относительно вида функции, ее области определения, области расположения всех и каждого нуля в отдельности. Существующие схемы не обладают инвариантной идентификацией области нулей или требуют начального приближения, соответственно они могут игнорировать наличие нулей в некоторых сложных случаях, либо не достигают в этих случаях аналогичной точности вычислений. Существующие схемы, как правило, неустойчивы при поиске нулей трансцендентных функций. Помимо устойчивости предложенный метод обладает параллелизмом и свойством совмещения локализации одновременно всех нулей с их вычислением с априори заданной границей абсолютной погрешности.

### Заключение

В работе описан распараллеливаемый метод вычисления нулей полиномов и функций общего вида с учетом кратности, инвариантный относительно области поиска нулей, устойчивость которого обусловлена применением сортировки. Метод модифицирован для приближенного вычисления полюсов и вычетов функций и на этой основе распространяется на вычисление криволинейных интегралов по замкнутому контуру. Разработанная схема приближенного вычисления полюсов комплексных функций с учетом порядка использует обработку индексов отсортированных элементов, в результате не вносится вычислительная погрешность, метод достигает устойчивости, фактически не включает ограничений на вид входной функции, отличаясь этим от известных аналогов. Построенные на данной основе алгоритмы приближенного вычисления вычетов функций в полюсах отличаются от известных методов вычислительной устойчивостью и параллелизмом. Представлено применение предложенных методов для анализа цифровых фильтров с помощью обратного z-преобразования, для анализа временной функции линейной динамической системы и для анализа устойчивости дискретной цепи.

1. Ромм Я. Е., Заика И.В., Тюшнякова И.А. Идентификация экстремумов функций на основе сортировки с приложением к вычислительным схемам алгебры, анализа и распознаванию изображений. // Проблемы програмування: Матеріали 5-й міжнар. науково-практичної конф. з програмування УкрПРОГ'2006. 23–25 травня 2006 р. Україна, Київ. – 2006. – № 2–3. – С. 708 – 717.
2. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. I // Кибернетика и системный анализ, Киев. – 2007. – № 1. – С. 165 – 182.
3. Ромм Я.Е., Гуревич М.Ю., Белоконова С.С., Соловьёва И.А. Вычисление нулей и полюсов функций на основе устойчивой адресной сортировки с приложением к поиску и распознаванию. // Проблемы програмування. – 2004. – № 2 – 3. – С. 462–472.
4. Ромм Я.Е., Тюшнякова И.А. Применение сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений. – Таганрог: Изд-во Таганрог. гос. пед. ин-та, 2009. – 155 с.
5. Веселая А.А. Вычисление нулей и экстремумов функций при вариации параметров на основе сортировки с приложением к моделированию устойчивости систем линейных дифференциальных уравнений: Автореф. дис. ... канд. техн. наук. – Изд-во ТТИ ЮФУ, Таганрог, 2009. – 18 с.
6. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. II // Кибернетика и системный анализ. – Киев. – 2007. – № 2. – С. 161 – 174.
7. Заика И.В. Разработка и исследование схем оптимизации на основе алгоритмов сортировки с приложением к идентификации экстремумов решений дифференциальных уравнений: Автореф. дис. ... канд. техн. наук. Изд-во ТРТУ, Таганрог, 2007. – 19 с.
8. Тюшнякова И.А. Разработка и исследование схем применения сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений; Диссертация канд. техн. наук. – Таганрог: ТРТУ. – 2006. – 196 с.
9. Баскаков С.И. Радиотехнические цепи и сигналы. – М.: Высшая школа, 2003. – 462 с.
10. Березин И.С., Жидков Н.П. Методы вычислений. – Т. 2. – М.: Физматгиз, 1962. – 640 с.
11. Киреев В.И., Пантелеев А.В. Численные методы в примерах и задачах. – М.: Высшая школа, 2004. – 480 с.
12. Kalinina E.A., Uteshev A.Yu. Determination of the Number of Roots of a Polynomial Lying in a Given Algebraic Domain // Linear Algebra and its Applications. – 1993. – V. 185. – P. 61 – 81.
13. Semerdziev Khr. Iteration methods for simultaneous finding all roots of generalized polynomial equations // Math. Balkan. – 1994. – 8, № 4. – P. 311 – 335.
14. Маркушевич А.И. Краткий курс теории аналитических функций. – М.: Наука, 1978. – 416 с.
15. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. – СПб: Лань, 2002. – 736 с.
16. Форсайт Д.Э., Малькольм М., Моулдер К. Машинные методы математических вычислений. – М.: Мир, 1980. – 279 с.