

УДК 519.6

ПРОГРАММНОЕ ОПРЕДЕЛЕНИЕ ВЫЧЕТОВ ФУНКЦИЙ В ДЕЙСТВИТЕЛЬНЫХ ПОЛЮСАХ НА ОСНОВЕ СОРТИРОВКИ

Тюшнякова И.А.

Таганрогский институт имени А.П. Чехова (филиал) РГЭУ (РИНХ), Таганрог,
e-mail: Solovyova_Irina@mail.ru

В статье изложен компьютерный метод применения сортировки для вычисления вычетов функций в действительных полюсах. Задача нахождения вычетов функций в особых точках актуальна сама по себе, кроме того, её решение ведет к вычислению криволинейных интегралов по замкнутому контуру. Рассматриваются функции комплексного переменного, которые имеют в качестве особенностей только конечные полюсы. Полюсы функции и их порядок программно определяются с использованием схем сортировки. В качестве сортировки можно использовать любую устойчивую сортировку со свойствами адресности и обратной адресности. Вычеты функции в полюсах программно определяются последовательным нахождением коэффициентов ряда Лорана. Предложенные алгоритмы проиллюстрированы структурными схемами и примерами программной реализации. В работе приводятся фрагменты листинга программной реализации в среде Delphi на языке Pascal. Описанный метод отличается от известных методов своим построением на базе схем сортировки, а также возможностью распараллеливания. Описанный в статье метод позволяет получить точный результат, помимо этого, существенна возможность программной реализации метода. Метод может использоваться как для непосредственного определения вычетов, так и для вычисления криволинейных интегралов по замкнутому контуру.

Ключевые слова: сортировка, вычеты функций, полюса функций

SOFTWARE DEFINITION RESIDUES OF FUNCTIONS IN THE REAL POLES ON THE BASIS OF THE SORT

Tyushnyakova I.A.

Taganrog Institute named after A.P. Chekhov (branch) Rostov State University of Economics,
Taganrog, e-mail: Solovyova_Irina@mail.ru

The article describes a computer method of applying sorting to calculate the residues of functions in real poles. The problem of finding the residues of functions at singular points is actual in itself, in addition, its solution leads to the calculation of curvilinear integrals over a closed contour. We consider functions of a complex variable that have only finite poles as singularities. The poles of a function and their order are programmed using the sorting schemes. As sorting, you can use any stable sorting with the properties of targeting and reverse targeting. The deductions of the function at the poles are programmed by the sequential finding of the coefficients of the Laurent series. The proposed algorithms are illustrated by structural diagrams and examples of software implementation. In the work fragments of listing of software implementation in the environment of Delphi in Pascal language are given. The described method differs from known methods by its construction on the basis of sorting schemes, as well as the possibility of parallelization. The method described in the article makes it possible to obtain an accurate result, in addition, the possibility of software implementation of the method is essential. The method can be used both for direct determination of residues and for calculating curvilinear integrals over a closed contour.

Keywords: sorting, residues of function, poles of functions

Задача определения вычетов функций в особых точках актуальна сама по себе, кроме того, её решение ведет к вычислению криволинейных интегралов по замкнутому контуру. Будем рассматривать функции комплексного переменного, у которых в качестве особенностей выступают только конечные полюсы, и искать вычеты функций в этих полюсах.

Вычет функции в любой конечной изолированной особой точке равен коэффициенту C_{-1} при первой отрицательной степени в разложении функции в ряд Лорана в окрестности этой точки [1], т.е. при $\frac{1}{z - z_0}$

для z_0 , принадлежащей области комплексных чисел: $res_{z_0} f(z) = C_{-1}$. По теореме Лора-

на [2, 3] о разложении аналитической функции в кольце в ряд по целым степеням имеет

$$\text{место равенство } f(z) = \sum_{n=-\infty}^{\infty} C_n (z - z_0)^n .$$

Поскольку рассматриваются функции, имеющие в качестве особенностей только полюсы z_0 , то в выколотой окрестности точки z_0 ряд Лорана функции $f(z)$ содержит конечное число членов с отрицательными степенями [4]. То есть, если z_0 – полюс n -го порядка, то

$$f(z) = \frac{C_{-n}}{(z - z_0)^n} + \frac{C_{-n+1}}{(z - z_0)^{n-1}} + \dots + \frac{C_{-1}}{z - z_0} + C_0 + C_1(z - z_0) + \dots$$

Задача состоит в программном вычислении коэффициента C_{-1} . Очевидны соотношения

$$\left. \begin{aligned} C_{-n} &= \lim_{z \rightarrow z_0} f(z)(z - z_0)^n, \\ C_{-n+1} &= \lim_{z \rightarrow z_0} \frac{f(z)(z - z_0)^n - C_{-n}}{z - z_0}, \\ C_{-n+2} &= \lim_{z \rightarrow z_0} \frac{f(z)(z - z_0)^n - C_{-n} - C_{-n+1}(z - z_0)}{(z - z_0)^2}, \\ &\dots \end{aligned} \right\} (*)$$

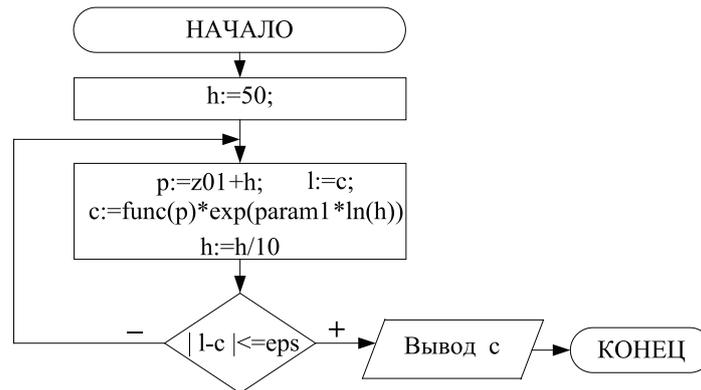


Рис. 1. Структурная схема алгоритма нахождения коэффициентов ряда Лорана с использованием предельного перехода

Последовательное вычисление всех отрицательных коэффициентов ряда Лорана рекуррентно приведет к нахождению C_{-1} . Реализацию предельного перехода алгоритмически предполагалось [5] осуществить следующим образом (рис. 1).

В данном алгоритме используются следующие обозначения: $z01$ – полюс исходной функции $func(z)$, переменная $param1$ определяет порядок полюса $z01$, eps – наперед заданная погрешность вычислений.

Однако таким образом можно точно вычислить лишь коэффициент ряда Лорана с наименьшей отрицательной степенью $C_{-param1}$. Вычисление следующего коэффициента таким же образом (заменяя лишь выражение для переменной C на выражение, определяющее $C_{-param1+1}$) дает шесть верных цифр после десятичной точки. Каждый последующий коэффициент вычисляется с большей погрешностью, чем предыдущий; ошибка накапливается, в результате оказывается невозможным точно вычислить вычет функций в полюсах, порядок которых выше второго.

С целью преодоления возникшей трудности предлагается алгоритм приближенного нахождения коэффициентов ряда

Лорана, базирующийся на использовании сортировки:

1. На основе сортировки [6, 7], по изложенной в [8, 9] схеме вычисления полюсов комплексных функций находятся все полюсы z_0 функции $f(z)$, а также их порядок n [10, 11].

2. Для определения вычета функции в первом из найденных полюсов z_k порядка $k1$ вычисляются необходимые коэффициенты $C_{-k1}, C_{-k1+1}, \dots, C_{-2}, C_{-1}$ из (*) при повторном применении схемы вычисления полюсов исходной функции (уже без учета порядка) в месте спуска к наименьшему значению в окрестности локализованной точки при достаточном сужении области поиска вокруг уже определенного полюса. Иными словами, искомые коэффициенты вычисляются по данной схеме непосредственно по виду выражений, стоящих под знаком предела из (*), по ходу циклического приближения значения z к заново вычисляемому значению z_0 .

3. Пункт 2 повторяется для каждого из найденных полюсов, пока не будет определен вычет во всех полюсах.

Программная реализация пункта 2 опирается на алгоритм, представленный на рис. 2.

Как отмечалось выше, область поиска полюсов нужно ограничить небольшой окрестностью вокруг уже известного полюса, что и отражено в блоке описания констант:

$$\begin{aligned} \text{eps}0 &= \text{zz} * \text{g}; \text{x}00 = \text{z}0 - \text{eps}0 / \text{g}; \\ \text{x}11 &= \text{z}0 + \text{eps}0 / \text{g}. \end{aligned}$$

В ходе численного эксперимента было установлено, что численные значения константы *eps0* и граничных точек окрестности поиска *x00*, *x11* зависят друг от друга. Эту зависимость определяет коэффициент *g*, введенный в блоке описания констант. Значение коэффициента *g* может быть варьировано в рамках значений от 10^{-424} до 10^{-9} , что не влияет на точность результата и позволяет произвольно сужать или расширять окрестность поиска.

Значение множителя *zz* константы *eps0* подобрано эвристически в ходе численного эксперимента, оно варьируется в зависимости от порядка полюсов. Однако выбранные значения *zz*, приведенные в каждом случае в примере 1, остаются постоянными для различных функций, которые имеют одинаковый порядок полюсов.

Приближения к коэффициенту ряда Лорана, который вычисляется по соответствующим формулам из (*), запоминаются в массиве *vv*, после чего искомым коэффициент представляется как среднее арифметическое значений первых десяти элементов этого массива.

Замечание. Заметим, что весовой одночлен является необходимой компонентой для вычисления порядка полюсов функций в случае отсутствия информации о количестве полюсов, поскольку возможен случай единственности полюса. Отсутствие весовой функции может повлиять на правильность определения порядка полюса, но никак не на само значение полюса. Поэтому использование весового одночлена является излишним при вычислении вычетов функций в полюсах.

Описанный метод ниже иллюстрируется на примере.

Пример. Вычислить вычет функции $f(z) = \frac{5}{(z+1)^3 \left(z + \frac{2}{3}\right)}$ в полюсе, имеющем порядок, отличный от 1.

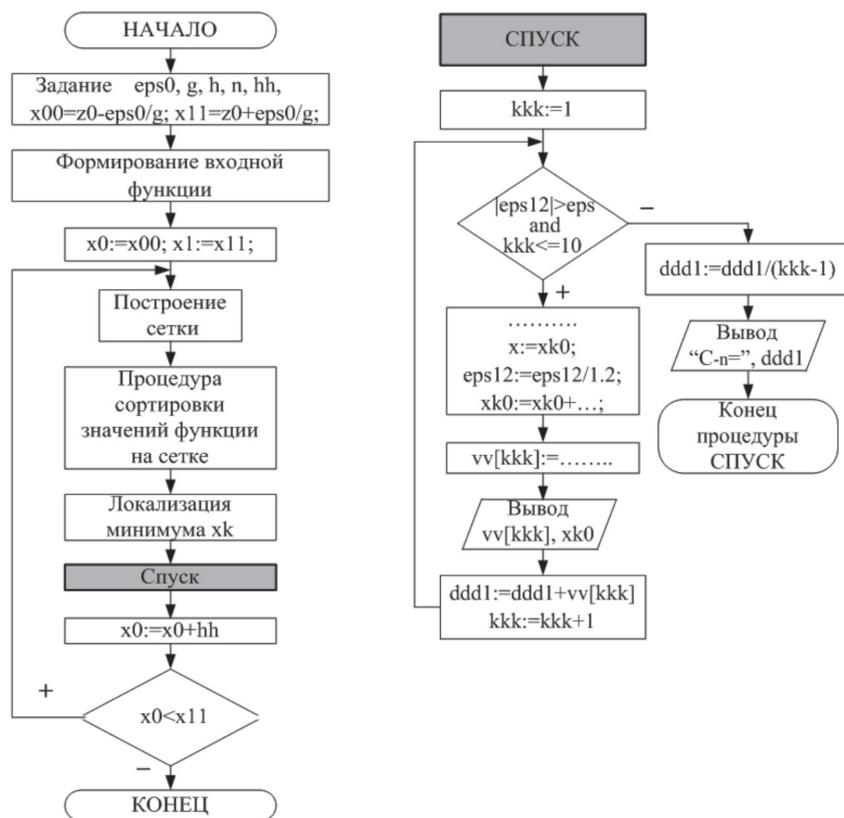


Рис. 2. Упрощенная структурная схема алгоритма приближенного вычисления коэффициента ряда Лорана в окрестности точки $z = z_0$

Отметим, что представление исходной функции в таком виде, когда обратная к ней задана разложением на множители, используется с целью иллюстрации, для сравнения результатов работы программы с очевидными значениями полюсов. Исходная функция может быть представлена в виде любой дробно-рациональной функции.

Определим все полюсы исходной функции и их порядок. Как и прежде, изменения произойдут лишь в блоке описания констант и функций:

На вход программы, полный текст которой представлен в [5], подается функция, нули которой совпадают с полюсами исходной функции $f(z)$, она получена при помощи простейших алгебраических преобразований обратной функции к $f(z)$ к виду

$$\left| \frac{1}{f(x, y)} \right|^2 = u^2(x, y) + v^2(x, y),$$

где

$$u(x, y) = 3x + x^4 - 5y^2 - 11xy^2 + y^4 + \frac{2}{3} + 5x^2 + \frac{11}{3}x^3 - 6x^2y^2,$$

$$v(x, y) = 10xy + 4x^3y - \frac{11}{3}y^3 + 11x^2y - 4xy^3 + 3y.$$

В качестве весовой используется произвольно выбранная функция $\rho(x, y) = (x - 1,052)^2 + y^2$. Результаты работы программы приведены в таблице.

Результаты вычисления полюсов функции $f(z) = \frac{5}{(z+1)^3 \left(z + \frac{2}{3}\right)}$ с учетом их порядка

Действительная часть полюса	Мнимая часть полюса	Порядок полюса	Значение обратной функции
-1.00E+0000	1.05E-1332	3	0.00E+0000
-6.66E-0001	1.05E-1332	1	0.00E+0000

Мнимая часть полученных полюсов настолько мала (порядка 10^{-1332}), что будем считать ее нулем и далее возможно использовать программу вычисления действительных нулей функций из [12, 13].

По условию задачи, необходимо вычислить вычет функции $f(z)$ в первом полюсе $z = -1$. Данный полюс имеет порядок 3, поэтому коэффициенты ряда Лорана имеют следующий вид:

$$C_{-3} = \lim_{z \rightarrow z_0} f(z)(z - z_0)^n, \quad C_{-2} = \lim_{z \rightarrow z_0} \frac{f(z)(z - z_0)^3 - C_{-3}}{z - z_0},$$

$$C_{-1} = \lim_{z \rightarrow z_0} \frac{f(z)(z - z_0)^3 - C_{-3} - C_{-2}(z - z_0)}{(z - z_0)^2}.$$

Фрагмент листинга программы на языке Pascal [14, 15] для вычисления коэффициента C_{-3} – раздела описания констант и функций и спуска к наименьшему значению в окрестности локализованной точки:

```

.....
const eps=1.1e-444; g=1e-19; eps0=1e-18*g;
h=eps0/(2*1024/20); n=1024; nn=trunc(n/2); hh=n*h; tt=64; np=4;
{область поиска полюсов}
x00=-1-eps0/g; x11=-1+eps0/g; .....
kkk:=1;
{Вычисление коэффициентов Лорана одновременно при спуске}
{к наименьшему значению в окрестности локализованной точки}
while (abs(eps12) > eps) and (kkk <= 10) do begin x:=xk0;
minx(x,min,ee); eps12:=eps12/1.2; xk0:=xk0+ee*hx-eps12; vv[kkk]:=sqr(xk0+1)*(xk0+1)*f(xk0);
Writeln ('C(-n)=' ,vv[kkk], ' ,xk0,' ,func(xk0,korx,r,param));
ddd1:=ddd1+vv[kkk]; kkk:=kkk+1; end; ddd1:=ddd1/(kkk-1); writeln ('!!!!C(-n)=' ,ddd1);
eps13:=eps0/1.2; x:=xk0+ee*hx+eps12; xk1:=xk1+eps12;end; .....
Результат работы программы:
C<-n>=-1.500000000000000E+0000.

```

Вычисленные коэффициенты C_{-3} заносятся в массив vv , на выходе программы представлено среднее арифметическое значений первых десяти элементов этого массива. Значение C_{-3} для функции $f(z)$ вычислено «абсолютно» точно (с точностью до формата представления выводимых значений). Как показал численный эксперимент, коэффициент с наименьшей отрицательной степенью всегда удается вычислить с такой точностью.

Для вычисления коэффициента C_{-2} ряда Лорана функции $f(z)$ необходимо в блоке описания констант положить $eps0=1e-10*g$ и заменить фрагмент программы, в котором осуществляется вычисление коэффициентов Лорана одновременно при спуске к наименьшему значению в окрестности локализованной точки:

```
while (abs(eps12) > eps) and (kkk<=10) do begin x:=xk0; minx(x,min,ee);
eps12:=eps12/1.2; xk0:=xk0+ee*hx-eps12; vvv[kkk]:=(sqr(xk0+1)*(xk0+1)*f(xk0)+15)/(xk0+1);
Writeln ('C(-n+1)=' ,vvv[kkk], ' ', xk0, ' ',func(xk0,korx,r,param));
ddd2:=ddd2+vvv[kkk]; kkk:=kkk+1; end; ddd2:=ddd2/(kkk-1); writeln('!!!!C(-n+1)=' ,ddd2);
```

Результат работы программы:

$C<-n+1>=-4.49999999783160E+0001.$

Этот результат можно считать вычисленным сравнительно точным (по сравнению с $C_{-2} = -45$).

Численный эксперимент показал, что для наиболее точного вычисления последнего коэффициента C_{-1} изменения следует внести в блок описания констант: $eps0=1e-5*g$, а фрагмент, в котором осуществляется вычисление коэффициентов Лорана одновременно при спуске к наименьшему значению в окрестности локализованной точки изменить на

```
while (abs(eps12) > eps) and (kkk<=10) do begin x:=xk0; minx(x,min,ee);
eps12:=eps12/1.2; xk0:=xk0+ee*hx-eps12;
vvv[kkk]:=(sqr(xk0+1)*(xk0+1)*f(xk0)+15+44.9999999783160*(xk0+1))/sqr(xk0+1);
Writeln ('C(-n+2)=' ,vvv[kkk], ' ', xk0, ' ',func(xk0,korx,r,param));
ddd2:=ddd2+vvv[kkk]; kkk:=kkk+1; end; ddd2:=ddd2/(kkk-1);
writeln('!!!!C(-n+2)=' ,ddd2); goto 777;
```

Результат работы программы:

$C<-n+2>=-1.34993781725914E+0002.$

Приближение этого значения к истинному ($C_{-1} = -135$) достаточно грубое – с точностью до 10^{-2} . Увеличить точность вычисления вычета C_{-1} в этом варианте метода не удалось. А поскольку чем выше порядок полюса, тем большая погрешность вычислений будет накапливаться, данным способом можно вычислить вычеты функций в точках полюсов с порядком не выше третьего.

Значения коэффициентов ряда Лорана, вычисленных фактически без погрешности, можно получить, реализовав следующий алгоритм.

1. Определяются все полюсы z_k функции $f(z)$ с учетом их порядка на основе изложенного метода.
2. Для определения вычета функции в первом из найденных полюсов z_0 порядка n точно вычисляется коэффициент ряда Лорана C_{-n} с наименьшей отрицательной степенью способом, описанным выше.
3. Для вычисления коэффициентов C_{-n+k} (где $k = 1, 2, \dots, n-1$) формируется функция

$$f_k(z) = f(z) - \frac{C_{-n}}{(z - z_0)^n} - \frac{C_{-n+1}}{(z - z_0)^{n-1}} - \dots - \frac{C_{-n+k-1}}{(z - z_0)^{n-k+1}},$$

правая часть которой поступает в блок формирования входной функции только после предварительного преобразования ее к виду неприводимой рациональной дроби.

4. Пункты 2–3 повторяются для всех найденных полюсов до тех пор, пока не будет установлен вычет в каждом полюсе.

Описанный способ нахождения коэффициентов ряда Лорана и, на его основе, вычетов функции в полюсах представляется наиболее рациональным, несмотря на алгебраические преобразования, так как он позволяет получить точный результат, помимо этого, существенна возможность программной реализации этих преобразований ввиду их простоты. Рекуррентное вычисление последующих коэффициентов ряда Лорана через предыдущие описанным способом с априори заданной точностью позволяет определить необходимое количество коэффициентов при отрицательных степенях ряда Лорана.

Работа выполнена при финансовой поддержке РФФИ (проект 16-07-00100).

Список литературы

1. Маркушевич А.И. Краткий курс теории аналитических функций. – М.: Мир, 2009. – 423 с.
2. Лунц Г.Л., Эльсгольд Л.Э. Функции комплексного переменного с элементами операционного исчисления. – М.: Ленанд, 2017. – 314 с.
3. Свешников А.Г., Тихонов А.Н. Теория функций комплексной переменной. – М.: Физматлит, 2010. – 336 с.
4. Карасев И.П. Теория функций комплексного переменного. – М.: Физматлит, 2008. – 216 с.
5. Ромм Я.Е., Тюшнякова И.А. Применение сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений: учебное пособие. – Таганрог, 2009.
6. Вирт Н. Алгоритмы и структуры данных. – М.: ДМК-Пресс, 2016. – 272 с.
7. Заика И.В., Тюшнякова И.А. Обзор методов сортировки // Научный взгляд в будущее. – 2016. – Т. 2, № 1 (1). – С. 206–211.
8. Ромм Я.Е., Гуревич М.Ю., Белоконова С.С., Соловьева 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.
9. Ромм Я.Е., Заика И.В., Тюшнякова И.А. Локализация экстремумов и нулей функций на основе сортировки в приложении к анализу устойчивости // Фундаментальные исследования. – 2015. – № 12–4. – С. 718–723.
10. Ромм Я.Е., Соловьева И.А. Распараллеливаемый метод вычисления нулей многочленов в произвольной области комплексной плоскости // Депонированная рукопись № 210-B2005 14.02.2005.
11. 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.
12. 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.
13. 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. – Таганрог: ТРТУ, 2007.
14. Лукин С.Н. Turbo Pascal 7.0. – М.: Диалог-МИФИ, 2015. – 384 с.
15. Фаронов В.В. Turbo Pascal. – СПб.: Питер, 2007. – 367 с.