Введение

В работе рассматривается **глобальный бисекционный алгоритм** вычисления всех нулей полинома в комплексной плоскости [1].

В отличие от локальных итеративных методов, требующих задания начальных приближений, данный подход не зависит от выбора стартовой точки. Метод гарантирует обнаружение всех корней — как вещественных, так и комплексных, включая кратные — с заданной точностью.

Основная идея алгоритма

Метод опирается на классический **принцип аргумента** [7, 11, 12] и применение **последовательностей Штурма** [5, 8]. Эти инструменты позволяют вычислять число нулей полинома внутри произвольного прямоугольника комплексной плоскости.

Принципиальная особенность метода — его **глобальный характер**. Алгоритм стартует с области, гарантированно содержащей все корни полинома, и затем систематически делит её, локализуя каждый корень с требуемой точностью [1].

Область применения метода Вилфа

Метод Вилфа предназначен для работы с полиномами общего вида:

$$P(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \tag{1}$$

где коэффициенты $a_k \in \mathbb{C}$. В частности:

- метод применим к полиномам с вещественными и комплексными коэффициентами [1];
- при вещественных коэффициентах комплексные корни естественным образом учитываются в комплексной постановке задачи;
- не требуется симметрия коэффициентов или расположения корней.

Тип корней

Алгоритм находит все n корней полинома (1) (с учётом кратности), которые могут быть:

- **вещественными:** $z \in \mathbb{R}$;
- **комплексными:** $z = x + iy$, $y \neq 0$;
- **простыми:** корни кратности 1;
- **кратными:** корни кратности $m > 1$ (включая комплексно-сопряжённые кратные пары для вещественных полиномов) [1, 11];
- **кластеризованными:** корни, находящиеся на малом расстоянии друг от друга.

Ограничения метода

1. **Степень:** применяется к полиномам степени $n \geq 1$ [1].
2. **Корректность постановки:**
   - при $a_n = 0$ требуется предварительное приведение к стандартному виду [2];
   - нулевой полином ($P \equiv 0$) не рассматривается.
3. **Численная реализация:** точность локализации корней ограничена используемой арифметикой с плавающей точкой [1, 3].

Структура алгоритма

Алгоритм реализует итерационный процесс локализации нулей, основанный на повторном подсчёте их числа в прямоугольных областях комплексной плоскости [1].

Шаг 1: Построение вычислительного ядра. Формируется процедура подсчёта числа нулей полинома в произвольном прямоугольнике (формула (63)) [1].

Шаг 2: Инициализация области. Выбирается начальный прямоугольник R_0, содержащий все нули полинома [1, 6].

Шаг 3: Бисекция. Если область содержит более одного нуля или её размеры превышают требуемую точность, она делится на подпрямоугольники. Для каждой подобласти повторно вычисляется число нулей.

Шаг 4: Критерий завершения. Деление прекращается, когда каждая область либо не содержит нулей, либо содержит ровно один нуль и её размеры не превышают заданный порог.

Шаг 5: Формирование результата. Центры финальных прямоугольников принимаются в качестве приближений корней с оценкой погрешности [1, 3].

Таким образом, алгоритм представляет собой цикл

$$\text{подсчёт нулей} \;\rightarrow\; \text{деление области} \;\rightarrow\; \text{повторный подсчёт},$$

который продолжается до достижения заданной точности.

Преимущества метода

- **Глобальность:** не требует начальных приближений [1].
- **Корректный учёт кратности:** число нулей в малой окрестности совпадает с кратностью корня (см. (2)) [11].
- **Численный контроль:** потеря точности выявляется через анализ суммы перемен знака в формуле (63) [1].
- **Отсутствие дефляции:** вычисления выполняются с исходным полиномом (1).

Принцип аргумента не только обеспечивает математическую корректность алгоритма [7, 11], но и определяет его вычислительную структуру, делая метод особенно эффективным для полиномов с кратными и близко расположенными корнями [1].

Математическая основа — принцип аргумента и последовательности Штурма

Цель этапа

На данном этапе строится процедура точного подсчёта числа нулей полинома $P(z)$ внутри прямоугольника $R \subset \mathbb{C}$.

Основой служит принцип аргумента, который далее приводится к вычислительной форме с использованием последовательностей Штурма [1].

Принцип аргумента для полиномов

Теорема 1 (Принцип аргумента). Пусть $P(z)$ — полином, $R \subset \mathbb{C}$ — область с кусочно-гладкой границей ∂R, и $P(z) \neq 0$ для всех $z \in \partial R$.

Тогда число нулей полинома внутри R с учётом кратности равно

$$N(P, R) = \frac{1}{2\pi} \Delta_{\partial R} \arg P(z), \tag{2}$$

где $\Delta_{\partial R} \arg P(z)$ — приращение аргумента $P(z)$ при обходе ∂R в положительном направлении.

Доказательство. Так как $P(z) \neq 0$ на ∂R, логарифмическая производная

$$\frac{P'(z)}{P(z)} \tag{3}$$

аналитична в окрестности границы.

По формуле Коши [7, 12] для логарифмической производной имеем

$$\frac{1}{2\pi i} \oint_{\partial R} \frac{P'(z)}{P(z)}\, dz = \sum_{z_j \in R} m_j, \tag{4}$$

где m_j — кратность нуля z_j. Правая часть равна $N(P, R)$.

Заметим, что

$$\frac{d}{dz} \log P(z) = \frac{P'(z)}{P(z)}. \tag{5}$$

Поскольку $P(z) \neq 0$ на ∂R, вдоль контура можно выбрать непрерывную ветвь логарифма. Тогда

$$\oint_{\partial R} \frac{P'(z)}{P(z)}\, dz = \oint_{\partial R} d(\log P(z)) = \Delta_{\partial R} \log P(z). \tag{6}$$

Разложив логарифм через модуль и аргумент,

$$\log P(z) = \ln |P(z)| + i \arg P(z), \tag{7}$$

получаем, что при обходе замкнутого контура приращение $\ln |P(z)|$ равно нулю. Следовательно,

$$\Delta_{\partial R} \log P(z) = i\, \Delta_{\partial R} \arg P(z). \tag{8}$$

Подставляя в (4), получаем

$$N(P,R) = \frac{1}{2\pi}\Delta_{\partial R}\arg P(z). \tag{9}$$

$\square$

Переход к прямоугольным областям

В алгоритме область интегрирования выбирается в виде прямоугольника $R \subset \mathbb{C}$ [1]. Это позволяет явно параметризовать границу ∂R и свести вычисление изменения аргумента к анализу функций одной вещественной переменной.

Пусть вершины прямоугольника обозначены Q_1, Q_2, Q_3, Q_4 и занумерованы против часовой стрелки. Тогда граница состоит из четырёх отрезков

$$Q_1Q_2, \quad Q_2Q_3, \quad Q_3Q_4, \quad Q_4Q_1. \tag{10}$$

Параметризация стороны

Рассмотрим произвольную сторону Q_kQ_{k+1} ($Q_5 = Q_1$ для замыкания контура). Введём параметр $t \in [0,1]$ и зададим линейную параметризацию:

$$z^{(k)}(t) = Q_k + (Q_{k+1} - Q_k)t. \tag{11}$$

Проверим корректность параметризации:

$$z^{(k)}(0) = Q_k, \qquad z^{(k)}(1) = Q_{k+1}. \tag{12}$$

Следовательно, при изменении t от 0 до 1 точка $z^{(k)}(t)$ непрерывно проходит всю сторону от начальной вершины к конечной.

Преобразование полинома на стороне

Пусть

$$P(z) = \sum_{j=0}^{n} a_j z^j. \tag{13}$$

Подставим (11):

$$P(z^{(k)}(t)) = \sum_{j=0}^{n} a_j \big(Q_k + (Q_{k+1} - Q_k)t\big)^j. \tag{14}$$

Для каждой степени j применим биномиальную формулу:

$$(Q_k + (Q_{k+1} - Q_k)t)^j = \sum_{r=0}^{j} \binom{j}{r} Q_k^{j-r}(Q_{k+1} - Q_k)^r t^r. \tag{15}$$

Подставляя это в выражение для P, получаем

$$P(z^{(k)}(t)) = \sum_{j=0}^{n} a_j \sum_{r=0}^{j} \binom{j}{r} Q_k^{j-r}(Q_{k+1} - Q_k)^r t^r. \tag{16}$$

Поменяем порядок суммирования по j и r:

$$P(z^{(k)}(t)) = \sum_{r=0}^{n} \left(\sum_{j=r}^{n} a_j \binom{j}{r} Q_k^{j-r} (Q_{k+1} - Q_k)^r \right) t^r. \tag{17}$$

Обозначим внутреннюю сумму через комплексные коэффициенты:

$$c_r^{(k)} = \sum_{j=r}^{n} a_j \binom{j}{r} Q_k^{j-r} (Q_{k+1} - Q_k)^r. \tag{18}$$

Тогда

$$P(z^{(k)}(t)) = \sum_{r=0}^{n} c_r^{(k)} t^r. \tag{19}$$

Разделяя вещественную и мнимую части,

$$c_r^{(k)} = \alpha_r^{(k)} + i\beta_r^{(k)}, \tag{20}$$

получаем

$$P(z^{(k)}(t)) = \underbrace{\sum_{r=0}^{n} \alpha_r^{(k)} t^r}_{P_R^{(k)}(t)} + i \underbrace{\sum_{r=0}^{n} \beta_r^{(k)} t^r}_{P_I^{(k)}(t)}. \tag{21}$$

Таким образом, вдоль каждой стороны прямоугольника полином $P(z)$ превращается в пару вещественных полиномов $P_R^{(k)}(t)$ (вещественная часть) и $P_I^{(k)}(t)$ (мнимая часть).

Интерпретация полученного преобразования

Рассмотрим движение точки z вдоль стороны прямоугольника, заданное параметризацией $z^{(k)}(t)$, $t \in [0, 1]$. Тогда значение полинома

$$w(t) = P(z^{(k)}(t)) \tag{22}$$

задаёт непрерывную кривую в комплексной плоскости.

Таким образом, изменение аргумента $P(z)$ вдоль стороны эквивалентно изменению аргумента функции $w(t)$ при изменении параметра t от 0 до 1.

Если рассматривать всю границу прямоугольника, то суммарное изменение аргумента

$$\Delta_{\partial R} \arg P(z) \tag{23}$$

равно числу оборотов кривой $w(t)$ вокруг начала координат, умноженному на 2π. Согласно формуле принципа аргумента (2), это число совпадает с количеством нулей полинома внутри R.

Полученное ранее представление

$$P(z^{(k)}(t)) = P_R^{(k)}(t) + iP_I^{(k)}(t) \tag{24}$$

имеет принципиальное вычислительное значение. Изменение аргумента определяется знаком и нулями вещественных функций $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$, то есть задача сводится к анализу двух вещественных полиномов одной переменной.

Тем самым непрерывная комплексная задача подсчёта изменения аргумента приводится к дискретному анализу знаков вещественных функций, что делает её алгоритмически реализуемой.

Вычисление коэффициентов $\alpha_r^{(k)}$ и $\beta_r^{(k)}$

Для полинома $P(z)$ в стандартном виде (1), коэффициенты $\alpha_r^{(k)}$ и $\beta_r^{(k)}$ вычисляются по формуле [1]:

$$\alpha_r^{(k)} = \Re\left(\sum_{j=r}^{n} a_j \binom{j}{r} Q_k^{j-r}(Q_{k+1}-Q_k)^r\right), \quad \beta_r^{(k)} = \Im\left(\sum_{j=r}^{n} a_j \binom{j}{r} Q_k^{j-r}(Q_{k+1}-Q_k)^r\right). \quad (25)$$

На практике эти вычисления выполняются алгоритмически с помощью методов численного анализа [2, 3].

Пример параметризации сторон

Для применения метода Уилфа необходимо параметризовать каждую сторону прямоугольника [1]. Рассмотрим прямоугольник R с вершинами $Q_1 = 0$, $Q_2 = 3$, $Q_3 = 3+2i$, $Q_4 = 2i$, обходимыми против часовой стрелки.

Для каждой стороны Q_kQ_{k+1} ($k = 1, 2, 3, 4$, причём $Q_5 = Q_1$) используем линейную параметризацию (11).

Сторона 1: $Q_1 \to Q_2$

$$z^{(1)}(t) = 0 + (3-0)\cdot t = 3t, \quad t \in [0,1] \quad (26)$$

Длина стороны: $L_1 = |Q_2 - Q_1| = 3$.

Сторона 2: $Q_2 \to Q_3$

$$z^{(2)}(t) = 3 + (3+2i-3)\cdot t = 3 + 2it, \quad t \in [0,1] \quad (27)$$

Длина стороны: $L_2 = |Q_3 - Q_2| = 2$.

Сторона 3: $Q_3 \to Q_4$

$$z^{(3)}(t) = (3+2i) + (2i-(3+2i))\cdot t = (3-3t) + 2i, \quad t \in [0,1] \quad (28)$$

Длина стороны: $L_3 = |Q_4 - Q_3| = 3$.

Сторона 4: $Q_4 \to Q_1$

$$z^{(4)}(t) = 2i + (0-2i)\cdot t = 2i(1-t), \quad t \in [0,1] \quad (29)$$

Длина стороны: $L_4 = |Q_1 - Q_4| = 2$.

Полный периметр прямоугольника: $P = 10$.

Пример получения полиномов $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$

Рассмотрим полином 3-й степени:

$$P(z) = z^3 - 2z^2 + z - 2. \quad (30)$$

Для прямоугольника с заданными вершинами:

$$Q_1 = 0 + 0i, Q_2 = 3 + 0i, Q_3 = 3 + 2i, Q_4 = 0 + 2i. \tag{31}$$

вычисляем полиномы $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$ для каждой стороны:

1. **Сторона 1:** $Q_1 \to Q_2$**.** Параметризация: $z(t) = 3t$

$$P(z(t)) = (3t)^3 - 2(3t)^2 + (3t) - 2 = 27t^3 - 18t^2 + 3t - 2$$

$$\boxed{P_R^{(1)}(t) = 27t^3 - 18t^2 + 3t - 2, \quad P_I^{(1)}(t) = 0} \tag{32}$$

2. **Сторона 2:** $Q_2 \to Q_3$**.** Параметризация: $z(t) = 3 + 2i \cdot t$

$$\begin{aligned} z(t) &= 3 + 2i \cdot t \\ z(t)^2 &= (3 + 2it)^2 = 9 - 4t^2 + 12it \\ z(t)^3 &= (3 + 2it)^3 = 27 - 36t^2 + i(54t - 8t^3) \\ P(z(t)) &= [27 - 36t^2 - 2(9 - 4t^2) + 3 - 2] + i[54t - 8t^3 - 2(12t) + 2t] = \\ &= (27 - 36t^2 - 18 + 8t^2 + 3 - 2) + i(54t - 8t^3 - 24t + 2t) = \\ &= (10 - 28t^2) + i(32t - 8t^3) \end{aligned}$$

$$\boxed{P_R^{(2)}(t) = -28t^2 + 10, \quad P_I^{(2)}(t) = -8t^3 + 32t} \tag{33}$$

3. **Сторона 3:** $Q_3 \to Q_4$**.** Параметризация: $z(t) = 3(1 - t) + 2i$

$$\begin{aligned} z(t) &= 3(1 - t) + 2i = (3 - 3t) + 2i \\ z(t)^2 &= [(3 - 3t) + 2i]^2 = (3 - 3t)^2 + 4i(3 - 3t) - 4 = \\ &= (9 - 18t + 9t^2 - 4) + i(12 - 12t) = \\ &= (5 - 18t + 9t^2) + i(12 - 12t) \\ z(t)^3 &= z(t) \cdot z(t)^2 = [(3 - 3t) + 2i] \cdot [(5 - 18t + 9t^2) + i(12 - 12t)] = \\ &= (-9 - 45t + 81t^2 - 27t^3) + i(46 - 108t + 54t^2) \\ P(z(t)) &= [z^3 - 2z^2 + z - 2] \end{aligned}$$

Вещественная часть:

$$\begin{aligned} &= (-9 - 45t + 81t^2 - 27t^3) - 2(5 - 18t + 9t^2) + (3 - 3t) - 2 \\ &= -18 - 12t + 63t^2 - 27t^3 \end{aligned}$$

Мнимая часть:

$$\begin{aligned} &= (46 - 108t + 54t^2) - 2(12 - 12t) + 2t \\ &= 22 - 82t + 54t^2 \end{aligned}$$

$$\boxed{\begin{aligned} &P_R^{(3)}(t) = -27t^3 + 63t^2 - 12t - 18, \\ &P_I^{(3)}(t) = 54t^2 - 82t + 22 \end{aligned}} \tag{34}$$

4. **Сторона 4: $Q_4 \to Q_1$.** Параметризация: $z(t) = 2i(1-t)$

$$\begin{aligned}
z(t) &= 2i(1-t)\\
z(t)^2 &= [2i(1-t)]^2 = -4(1-t)^2\\
z(t)^3 &= [2i(1-t)]^3 = -8i(1-t)^3\\
P(z(t)) &= -8i(1-t)^3 + 8(1-t)^2 + 2i(1-t) - 2 =\\
&= 6 - 16t + 8t^2 + i(-6 + 22t - 24t^2 + 8t^3)
\end{aligned}$$

$$\boxed{\begin{aligned}
&P_R^{(4)}(t) = 8t^2 - 16t + 6,\\
&P_I^{(4)}(t) = 8t^3 - 24t^2 + 18t - 6
\end{aligned}} \tag{35}$$

Пример вычисления рациональной функции $R^{(k)}(t)$

Для каждой стороны $k = 1, 2, 3, 4$ прямоугольника, после получения полиномов $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$, формируем рациональную функцию [1]:

$$R^{(k)}(t) = \frac{P_I^{(k)}(t)}{P_R^{(k)}(t)}, \quad t \in [0, 1] \tag{36}$$

Конкретно для нашего примера с полиномом $P(z) = z^3 - 2z^2 + z - 2$:

1. **Сторона 1 ($Q_1 \to Q_2$)**:
$$R^{(1)}(t) = \frac{0}{27t^3 - 18t^2 + 3t - 2} = 0. \tag{37}$$

2. **Сторона 2 ($Q_2 \to Q_3$)**:
$$R^{(2)}(t) = \frac{-8t^3 + 32t}{-28t^2 + 10}. \tag{38}$$

3. **Сторона 3 ($Q_3 \to Q_4$)**:
$$R^{(3)}(t) = \frac{54t^2 - 82t + 22}{-27t^3 + 63t^2 - 12t - 18}. \tag{39}$$

4. **Сторона 4 ($Q_4 \to Q_1$)**:
$$R^{(4)}(t) = \frac{8t^3 - 24t^2 + 18t - 6}{8t^2 - 16t + 6}. \tag{40}$$

Проверка формулы для коэффициентов

Для нашего примера с полиномом $P(z) = z^3 - 2z^2 + z - 2$ коэффициенты в стандартном виде:

$$a_0 = -2, \quad a_1 = 1, \quad a_2 = -2, \quad a_3 = 1. \tag{41}$$

Покажем, как полученные выражения для $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$ соответствуют формуле (25) [1]. Например, для стороны 1 ($Q_1 = 0$, $Q_2 = 3$, $Q_2 - Q_1 = 3$):

$$\begin{aligned}\alpha_0^{(1)} &= \Re\left(a_0 \cdot 1 \cdot Q_1^0 \cdot 3^0 + a_1 \cdot 1 \cdot Q_1^1 \cdot 3^0 + a_2 \cdot 1 \cdot Q_1^2 \cdot 3^0 + a_3 \cdot 1 \cdot Q_1^3 \cdot 3^0\right) \\ &= \Re\big((-2) \cdot 1 \cdot 0^0 \cdot 1 + 1 \cdot 1 \cdot 0^1 \cdot 1 + (-2) \cdot 1 \cdot 0^2 \cdot 1 + 1 \cdot 1 \cdot 0^3 \cdot 1\big) = -2\end{aligned} \tag{42}$$

$$\begin{aligned}\alpha_1^{(1)} &= \Re\left(a_1 \cdot \binom{1}{1} Q_1^0 \cdot 3^1 + a_2 \cdot \binom{2}{1} Q_1^1 \cdot 3^1 + a_3 \cdot \binom{3}{1} Q_1^2 \cdot 3^1\right) \\ &= \Re\left(1 \cdot 1 \cdot 1 \cdot 3 + (-2) \cdot 2 \cdot 0 \cdot 3 + 1 \cdot 3 \cdot 0 \cdot 3\right) = 3\end{aligned} \tag{43}$$

$$\begin{aligned}\alpha_2^{(1)} &= \Re\left(a_2 \cdot \binom{2}{2} Q_1^0 \cdot 3^2 + a_3 \cdot \binom{3}{2} Q_1^1 \cdot 3^2\right) \\ &= \Re\left((-2) \cdot 1 \cdot 1 \cdot 9 + 1 \cdot 3 \cdot 0 \cdot 9\right) = -18\end{aligned} \tag{44}$$

$$\alpha_3^{(1)} = \Re\left(a_3 \cdot \binom{3}{3} Q_1^0 \cdot 3^3\right) = \Re(1 \cdot 1 \cdot 1 \cdot 27) = 27 \tag{45}$$

Таким образом, действительно:

$$P_R^{(1)}(t) = \alpha_0^{(1)} + \alpha_1^{(1)} t + \alpha_2^{(1)} t^2 + \alpha_3^{(1)} t^3 = -2 + 3t - 18t^2 + 27t^3 \tag{46}$$

что совпадает с ранее вычисленным $P_R^{(1)}(t) = 27t^3 - 18t^2 + 3t - 2$.

Аналогично, для всех остальных сторон полученные коэффициенты $\alpha_r^{(k)}$ и $\beta_r^{(k)}$ можно вывести по формуле (25), что подтверждает корректность проведённых вычислений рациональных функций $R^{(k)}(t)$.

Математический смысл параметра t и функции $R^{(k)}(t)$

Для понимания того, как изменение аргумента полинома вдоль стороны прямоугольника сводится к анализу вещественной функции, необходимо детально рассмотреть параметризацию и её последствия.

Геометрический смысл параметра t

Рассмотрим сторону прямоугольника, соединяющую вершины Q_k и Q_{k+1}. Введём параметр $t \in [0, 1]$, который линейно интерполирует положение точки между этими вершинами:

$$z(t) = Q_k + (Q_{k+1} - Q_k)t, \quad t \in [0, 1]. \tag{47}$$

При $t = 0$ точка находится в Q_k, при $t = 1$ — в Q_{k+1}, а при промежуточных значениях t равномерно проходит всю сторону. Таким образом, непрерывное двумерное движение вдоль стороны заменяется изменением одного скалярного параметра.

Преобразование полинома в функции от t

Подставим параметризацию (47) в полином $P(z)$. В результате получаем комплекснозначную функцию вещественного переменного:

$$P(z(t)) = P_R^{(k)}(t) + iP_I^{(k)}(t), \tag{48}$$

где $P_R^{(k)}(t) = \Re(P(z(t)))$ и $P_I^{(k)}(t) = \Im(P(z(t)))$ — вещественные полиномы от t, степени не выше n.

Связь с аргументом комплексного числа

Для любого комплексного числа $w = x + iy$, не лежащего на мнимой оси ($x \neq 0$), его аргумент связан с компонентами соотношением:

$$\tan(\arg w) = \frac{y}{x}. \tag{49}$$

Применяя это к $w = P(z(t))$ и используя разделение (48), получаем:

$$\tan(\arg P(z(t))) = \frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} =: R^{(k)}(t). \tag{50}$$

Функция $R^{(k)}(t)$ представляет собой отношение двух вещественных полиномов, то есть рациональную функцию. Важно отметить, что это отношение определено всюду, кроме точек, где $P_R^{(k)}(t) = 0$.

Поведение в особых точках

Рассмотрим точку $t_0 \in [0, 1]$, в которой знаменатель обращается в ноль:

$$P_R^{(k)}(t_0) = 0, \quad P_I^{(k)}(t_0) \neq 0. \tag{51}$$

В этой ситуации $P(z(t_0)) = iP_I^{(k)}(t_0)$ лежит на мнимой оси. Функция $R^{(k)}(t)$ терпит разрыв, имея вертикальную асимптоту. Характер этого разрыва несёт важную информацию об изменении аргумента.

Если при возрастании t функция $R^{(k)}(t)$ переходит от $-\infty$ к $+\infty$ (скачок вверх), то соответствующее комплексное число $P(z(t))$ пересекает мнимую ось, переходя из левой полуплоскости в правую. При этом его аргумент изменяется скачкообразно:

$$\lim_{\varepsilon \to 0+} \arg P(z(t_0 - \varepsilon)) = \frac{\pi}{2}, \quad \lim_{\varepsilon \to 0+} \arg P(z(t_0 + \varepsilon)) = -\frac{\pi}{2}, \tag{52}$$

что даёт суммарное уменьшение аргумента на π.

Напротив, если $R^{(k)}(t)$ переходит от $+\infty$ к $-\infty$ (скачок вниз), то комплексное число пересекает мнимую ось в обратном направлении, и аргумент увеличивается на π.

Формула для изменения аргумента на стороне

Обобщая сказанное, полное изменение аргумента $P(z)$ при движении вдоль стороны $Q_k Q_{k+1}$ выражается через подсчёт направленных разрывов функции $R^{(k)}(t)$:

$$\Delta_{\text{сторона } k} \arg P(z) = \pi \left(N_{- \to +} - N_{+ \to -} \right), \tag{53}$$

где:

- $N_{- \to +}$ — количество скачков функции $R^{(k)}(t)$ от $-\infty$ к $+\infty$ (переход через мнимую ось слева направо);
- $N_{+ \to -}$ — количество скачков функции $R^{(k)}(t)$ от $+\infty$ к $-\infty$ (переход через мнимую ось справа налево).

Таким образом, исходная задача вычисления приращения аргумента комплекснозначной функции вдоль замкнутого контура сводится к существенно более простой задаче:

1. Для каждой стороны k построить вещественные полиномы $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$.
2. Найти все корни $t_0 \in (0, 1)$ знаменателя $P_R^{(k)}(t) = 0$, в которых числитель $P_I^{(k)}(t_0) \neq 0$.
3. Для каждого такого корня определить направление разрыва функции $R^{(k)}(t) = P_I^{(k)}(t)/P_R^{(k)}(t)$.
4. Подсчитать суммарное изменение аргумента по формуле (53).

Полученное значение используется в принципе аргумента (2) для определения числа нулей полинома внутри прямоугольника R. Направленные разрывы функции $R^{(k)}(t)$ эффективно обнаруживаются с помощью последовательностей Штурма, что будет подробно рассмотрено в следующем разделе.

Индекс Коши и его вычисление

Определение 1 (Индекс Коши). Для вещественной рациональной функции $R(t)$ на отрезке $[a, b]$ индекс Коши определяется как [5, 8]:

$$I_a^b(R(t)) = N^+ - N^-, \tag{54}$$

где:

- N^+ — количество точек, где $R(t)$ переходит от $-\infty$ к $+\infty$ (скачок вверх через $+\infty$)
- N^- — количество точек, где $R(t)$ переходит от $+\infty$ к $-\infty$ (скачок вниз через $-\infty$)

Индекс Коши имеет простую геометрическую интерпретацию. Рассмотрим график рациональной функции $R(t)$. Каждый раз, когда график пересекает вертикальную асимптоту:

- Если график "перескакивает" снизу вверх (от $-\infty$ к $+\infty$), это даёт вклад $+1$ в индекс;
- Если график "перескакивает" сверху вниз (от $+\infty$ к $-\infty$), это даёт вклад -1 в индекс.

Суммарный индекс равен разности между количеством "восходящих" и "нисходящих" перескоков через бесконечность [5].

Связь индекса Коши с изменением аргумента

Для стороны прямоугольника Q_kQ_{k+1} мы имеем полином $P(z(t)) = P_R^{(k)}(t) + iP_I^{(k)}(t)$. Аргумент этого комплексного числа равен [7]:

$$\arg P(z(t)) = \arctan\left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)}\right) + C_k, \tag{55}$$

где C_k — некоторая константа.

Изменение аргумента на стороне Q_kQ_{k+1} связано с индекс Коши следующим образом [1]:

$$\Delta_{Q_kQ_{k+1}} \arg P(z) = \pi \cdot I_0^1\left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)}\right). \tag{56}$$

Доказательство связи (56)

Рассмотрим момент t_0, когда $P_R^{(k)}(t_0) = 0$, а $P_I^{(k)}(t_0) \neq 0$. В окрестности этой точки:

$$\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} \sim \frac{C}{t - t_0}, \tag{57}$$

где C — некоторая константа.

Существует два случая:

1. Если $C > 0$, то при $t \to t_0^-$ отношение стремится к $-\infty$, а при $t \to t_0^+$ — к $+\infty$. Это соответствует переходу от $-\infty$ к $+\infty$, то есть $N^+ = 1$, $N^- = 0$. При этом аргумент $P(z(t))$ увеличивается на π при прохождении через t_0.
2. Если $C < 0$, то при $t \to t_0^-$ отношение стремится к $+\infty$, а при $t \to t_0^+$ — к $-\infty$. Это соответствует переходу от $+\infty$ к $-\infty$, то есть $N^+ = 0$, $N^- = 1$. При этом аргумент $P(z(t))$ уменьшается на π при прохождении через t_0.

Таким образом, каждый вклад в индекс Коши соответствует изменению аргумента на π с тем же знаком, что доказывает формулу (56) [1].

Суммирование по всем сторонам

Из аддитивности приращения аргумента вдоль замкнутого контура следует

$$\Delta_{\partial R} \arg P(z) = \sum_{k=1}^{4} \Delta_{Q_k Q_{k+1}} \arg P(z). \tag{58}$$

Согласно соотношению (56), изменение аргумента на каждой стороне выражается через индекс Коши [6]:

$$\Delta_{Q_k Q_{k+1}} \arg P(z) = \pi \cdot I_0^{L_k} \left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} \right), \quad L_k = |Q_{k+1} - Q_k|. \tag{59}$$

Подстановка (59) в (58) даёт

$$\Delta_{\partial R} \arg P(z) = \pi \sum_{k=1}^{4} I_0^{L_k} \left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} \right). \tag{60}$$

Применяя принцип аргумента (2) к (60), получаем выражение числа нулей через индексы Коши:

$$N(P, R) = \frac{1}{2} \sum_{k=1}^{4} I_0^{L_k} \left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} \right). \tag{61}$$

Для практического вычисления индексов Коши используется их связь с последовательностями Штурма [5, 8]. Пусть $V_k(t)$ — число перемен знака в последовательности Штурма для стороны k в точке t. Тогда

$$I_0^{L_k} \left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)} \right) = V_k(0) - V_k(L_k). \tag{62}$$

Подстановка (62) в (61) с учётом положительной ориентации обхода прямоугольника приводит к окончательной формуле [1]:

$$\boxed{N(P,R) = \frac{1}{2}\sum_{k=1}^{4}\left[V_k(L_k) - V_k(0)\right]}. \tag{63}$$

В оригинальной работе Вилфа [1] используется эквивалентная форма:

$$N(P,R) = -\frac{1}{2}\sum_{k=1}^{4} I_0^{L_k}\left(\frac{P_I^{(k)}(t)}{P_R^{(k)}(t)}\right), \tag{64}$$

что совпадает с (63) в силу соотношения $I_0^{L_k} = V_k(0) - V_k(L_k)$.

Таким образом, задача определения числа нулей полинома $P(z)$ в прямоугольнике R сводится к вычислению разности чисел перемен знака в последовательностях Штурма на концах каждой стороны.

Проблема прямого вычисления индекса Коши и её решение

Непосредственное вычисление индекса Коши согласно определению

$$I_a^b\left(\frac{f_2(x)}{f_1(x)}\right) = \frac{1}{\pi}\int_a^b d\left(\arctan\frac{f_2(x)}{f_1(x)}\right) \tag{65}$$

сопряжено с существенными вычислительными трудностями [5]:

1. требует определения всех вещественных корней полинома $P_R^{(k)}(t)$ на интервале $(0,1)$;
2. для каждого корня необходим анализ поведения рациональной функции $P_I^{(k)}(t)/P_R^{(k)}(t)$ в его окрестности;
3. при наличии кратных корней или корней, близких к границам интервала, возникают проблемы численной устойчивости.

Для преодоления указанных трудностей Вилф [1] применяет аппарат последовательностей Штурма, позволяющий вычислить индекс Коши без локализации корней и исследования локального поведения функции.

Основное соотношение. Пусть $\{f_1(x), f_2(x), \ldots, f_m(x)\}$ — последовательность Штурма для отрезка $[a,b]$, и $V(x)$ — число перемен знака в этой последовательности в точке x. Тогда индекс Коши выражается через разность чисел перемен знака на концах отрезка [5, 8]:

$$I_a^b\left(\frac{f_2(x)}{f_1(x)}\right) = V(a) - V(b). \tag{66}$$

Применяя (66) к каждой стороне прямоугольника и подставляя результат в (63), получаем эффективный вычислительный алгоритм, не требующий нахождения корней знаменателя. Построение соответствующих последовательностей Штурма для полиномов $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$ рассматривается далее.

Последовательности Штурма

Определение 2 (Последовательность Штурма). Система вещественных полиномов $f_1(x), f_2(x), \ldots, f_m(x)$ называется *последовательностью Штурма* на отрезке $[a, b]$, если выполнены следующие условия [5, 8]:

1. $f_m(x) \neq 0$ для всех $x \in [a, b]$;
2. если $f_i(\xi) = 0$ при $1 < i < m$, то $f_{i-1}(\xi) f_{i+1}(\xi) < 0$;
3. $f_1(x)$ и $f_2(x)$ не имеют общих корней на $[a, b]$.

Теорема 2 (Штурма). Пусть $\{f_1, f_2, \ldots, f_m\}$ — последовательность Штурма на $[a, b]$. Тогда индекс Коши рациональной функции $f_2(x)/f_1(x)$ на этом отрезке выражается через разность чисел перемен знака в последовательности на концах [5, 8]:

$$I_a^b \left(\frac{f_2(x)}{f_1(x)} \right) = V(a) - V(b), \tag{67}$$

где $V(x)$ — количество перемен знака в упорядоченном наборе $(f_1(x), f_2(x), \ldots, f_m(x))$; нулевые значения при подсчёте не учитываются.

Схема доказательства. Изменение $V(x)$ возможно лишь в точках, где обращается в нуль хотя бы один из полиномов последовательности. Условия определения гарантируют, что:

1. при прохождении через корень f_1 функция f_2/f_1 имеет простой полюс, что соответствует единичному вкладу в индекс Коши;
2. каждый такой полюс сопровождается изменением $V(x)$ ровно на единицу;
3. корни промежуточных полиномов не влияют на $V(x)$ в силу знакоопределённости соседних членов.

Следовательно, $V(a) - V(b)$ точно учитывает баланс переходов через бесконечность [5]. □

Построение последовательности Штурма для полинома

Для стороны прямоугольника $Q_k Q_{k+1}$ с параметризацией $z(t) = Q_k + i^{k-1} t$, $t \in [0, L_k]$, где $L_k = |Q_{k+1} - Q_k|$, вычисляем полином [1]:

$$P(z(t)) = P_R^{(k)}(t) + i P_I^{(k)}(t). \tag{68}$$

Берём $f_1(t) = P_R^{(k)}(t)$ и $f_2(t) = P_I^{(k)}(t)$ в качестве начальных полиномов и строим последовательность Штурма алгоритмом Евклида с отрицательными остатками [5, 8]:

$$\begin{aligned} f_1(t) &= q_1(t) f_2(t) - f_3(t), \\ f_2(t) &= q_2(t) f_3(t) - f_4(t), \\ &\vdots \\ f_{p-2}(t) &= q_{p-2}(t) f_{p-1}(t) - f_p(t), \end{aligned} \tag{69}$$

где каждый остаток f_{i+2} берётся с противоположным знаком по сравнению с обычным алгоритмом Евклида. Процесс останавливается, когда достигается полином $f_p(t)$, не имеющий корней на $[0, L_k]$.

Теорема 3 (Корректность построения). Для полиномов $f_1(t) = P_R^{(k)}(t)$ и $f_2(t) = P_I^{(k)}(t)$ без общих корней на $[0, L_k]$ последовательность, построенная по алгоритму 1, удовлетворяет определению 2 и является последовательностью Штурма [5].

Свойства построенной последовательности

- Последний полином $f_p(t) = \gcd(P_R^{(k)}(t), P_I^{(k)}(t))$ — константа.
- Условия последовательности Штурма выполняются автоматически [5].
- Длина последовательности $p \leq \deg(f_1) + 1$

Вычисление количества нулей. Для каждой стороны прямоугольника $k = 1, 2, 3, 4$ строится последовательность Штурма согласно описанной процедуре [1]. Затем вычисляются величины $V_k(0)$ и $V_k(L_k)$ — числа перемен знака в построенной последовательности в начале и в конце отрезка соответственно. Индекс Коши для данной стороны определяется как разность этих величин: $I_0^{L_k}(P_I^{(k)}/P_R^{(k)}) = V_k(0) - V_k(L_k)$ [5].

Согласно формуле Вилфа (63), общее количество нулей полинома $P(z)$ внутри прямоугольника [1]:

$$N(P, R) = \frac{1}{2} \sum_{k=1}^{4} (V_k(0) - V_k(L_k)) \, . \tag{70}$$

Замечание 1. Алгоритм требует $O(n^2)$ операций для каждой стороны, где $n = \deg P$ [1]. Результирующая последовательность содержит не более $n + 1$ полиномов [5].

Критерии применимости последовательностей Штурма

Для корректного применения последовательностей Штурма в методе Вилфа необходимо обеспечить выполнение ряда условий, гарантирующих возможность построения последовательности и достоверность получаемых результатов. Нарушение любого из этих критериев указывает на наличие корней полинома на границе прямоугольника и требует модификации исходной области [1].

S1. *Нетривиальность мнимой части.* Для каждой стороны $k = 1, 2, 3, 4$ должно выполняться условие $P_I^{(k)}(t) \not\equiv 0$.

Если $P_I^{(k)}(t) \equiv 0$, то вдоль всей стороны полином принимает только вещественные значения. Такая ситуация возникает, когда сторона прямоугольника лежит на вещественной оси. В этом случае построение последовательности Штурма невозможно, поскольку алгоритм требует деления на $P_I^{(k)}(t)$ [5]. Физически это означает, что корни полинома могут находиться непосредственно на рассматриваемой стороне.

S2. *Сходимость алгоритма Евклида.* Построение последовательности Штурма осуществляется по модифицированному алгоритму Евклида с отрицательными остатками [1]:

$$f_i^{(k)}(t) = q_i^{(k)}(t) f_{i+1}^{(k)}(t) - f_{i+2}^{(k)}(t), \quad i = 1, 2, \ldots, p-2, \tag{71}$$

где $f_1^{(k)} = P_R^{(k)}$, $f_2^{(k)} = P_I^{(k)}$.

Алгоритм должен завершиться за конечное число шагов, не превышающее $\deg P_R^{(k)} + 1$, причём на каждом шаге деление на $f_{i+1}^{(k)}$ должно быть выполнимо. Если на некотором шаге $f_{i+1}^{(k)} \equiv 0$, это свидетельствует о наличии общего корня у $P_R^{(k)}$ и $P_I^{(k)}$, что соответствует наличию кратного корня полинома $P(z)$ на рассматриваемой стороне.

S3. *Свойства последовательности Штурма.* Построенная система полиномов $\{f_1^{(k)}, f_2^{(k)}, \ldots, f_p^{(k)}$ должна удовлетворять определению последовательности Штурма на отрезке $[0, L_k]$ [5]:

(i) $f_p^{(k)}(t) \neq 0$ для всех $t \in [0, L_k]$;

(ii) если $f_i^{(k)}(t_0) = 0$ при $1 < i < p$, то $f_{i-1}^{(k)}(t_0) f_{i+1}^{(k)}(t_0) < 0$;

(iii) полиномы $f_1^{(k)}(t)$ и $f_2^{(k)}(t)$ не имеют общих корней на $[0, L_k]$.

Нарушение любого из этих условий означает, что построенная последовательность не является последовательностью Штурма, и применение формулы (67) некорректно. Такая ситуация возникает, в частности, когда граница прямоугольника проходит через корень полинома $P(z)$ или через точку, где $P_R^{(k)}$ и $P_I^{(k)}$ имеют общий корень.

Замечание 1

Все три критерия фактически проверяют отсутствие корней полинома $P(z)$ на границе прямоугольника R. Если какой-либо из критериев нарушен, необходимо уменьшить размер прямоугольника или сместить его так, чтобы граница не проходила через корни или особые точки [1]. В противном случае принцип аргумента в форме (2) неприменим, поскольку требует отсутствия нулей на контуре интегрирования.

Замечание 2

На практике критерий S3 проверяется косвенно: если алгоритм Евклида успешно завершился и последний полином $f_p^{(k)}$ является ненулевой константой, то все три условия выполняются автоматически [5]. Проблемы возникают лишь при наличии общих корней у $P_R^{(k)}$ и $P_I^{(k)}$, что проявляется на этапе S2.

Пример: анализ полинома $P(z) = z^3 - 2z^2 + z - 2$

Рассмотрим применение указанных критериев к полиному $P(z) = z^3 - 2z^2 + z - 2$ и прямоугольнику с вершинами $Q_1 = 0$, $Q_2 = 3$, $Q_3 = 3 + 2i$, $Q_4 = 2i$.

Рис. 1: Прямоугольник R с вершинами $Q_1 = 0$, $Q_2 = 3$, $Q_3 = 3 + 2i$, $Q_4 = 2i$ и корни полинома $P(z) = z^3 - 2z^2 + z - 2$.

1. **Сторона 1 ($Q_1 \to Q_2$):**

$$R^{(1)}(t) = \frac{0}{27t^3 - 18t^2 + 3t - 2} = 0. \tag{72}$$

Нарушение критерия S1: $P_I^{(1)}(t) \equiv 0$, что делает невозможным построение последовательности Штурма [5]. Формула (67) $I_a^b(f_2/f_1) = V(a) - V(b)$ теряет смысл, так как $f_2(t) \equiv 0$.

2. **Сторона 4 ($Q_4 \to Q_1$):**

$$R^{(4)}(t) = \frac{8t^3 - 24t^2 + 18t - 6}{8t^2 - 16t + 6}. \tag{73}$$

Проверяем наличие общих корней числителя и знаменателя. Решая систему:

$$\begin{cases} 8t^3 - 24t^2 + 18t - 6 = 0 \\ 8t^2 - 16t + 6 = 0 \end{cases}. \tag{74}$$

находим общий корень $t = 0.5$, что соответствует $z = 2i(1 - 0.5) = i$. Нарушение критерия S3: $P_R^{(4)}(t)$ и $P_I^{(4)}(t)$ имеют общий корень [5].

3. **Стороны 2 и 3:** Критерии S1-S3 формально выполняются, однако наличие корней на смежных сторонах делает невозможным применение метода в целом [1].

Признаки наличия корней на границе и выводы

Нарушение критериев S1-S3 служит индикатором проблем, связанных с наличием корней полинома на границе прямоугольника [1]:

1. **Нарушение критерия S1:** $P_I^{(k)}(t) \equiv 0$ возникает, когда сторона прямоугольника лежит на вещественной оси. Если при этом $P_R^{(k)}(t)$ имеет корень на $[0, L_k]$, то в этой точке $P(z(t)) = 0$, что соответствует корню полинома на границе. Для стороны 1: $P_R^{(1)}(t)$ имеет корень при $t = 2/3$, что даёт $z = 2$.

2. **Нарушение критерия S3 (условие 3):** Наличие общих корней у $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$ на отрезке $[0, L_k]$ означает существование $t_0 \in [0, L_k]$ такого, что $P(z(t_0)) = 0$. Для стороны 4: общий корень при $t = 0.5$ соответствует $z = i$.

3. **Непосредственный вывод о неприменимости метода:** Уже на этапе анализа рациональных функций можно сделать окончательный вывод [1]:
   - Для стороны 1: $R^{(1)}(t) \equiv 0$ означает, что индекс Коши $I_0^{L_1}(R^{(1)})$ не определён по формуле (67) [5].
   - Для стороны 4: наличие особенности типа $0/0$ в $R^{(4)}(t)$ при $t = 0.5$ нарушает условия теоремы Штурма [5].
   - Оба случая соответствуют наличию корней на границе ($z = 2$ и $z = i$), что противоречит условию отсутствия нулей $P(z)$ на ∂R [11].

Математические следствия

Нарушение критериев S1-S3 имеет следующие следствия [1, 11, 5]:

1. **Неприменимость теоремы Штурма:**

   Формула $I_a^b(f_2/f_1) = V(a) - V(b)$ требует корректной последовательности Штурма [5].

2. **Нарушение условий принципа аргумента:**

   Наличие корней на границе противоречит условию теоремы 1 (формула (2)) [11].

3. **Невозможность вычисления $N(P,R)$:**
   Формула $N(P,R) = \frac{1}{2}\sum_{k=1}^{4} I_k$ становится неприменимой [1].

Таким образом, проверка критериев S1-S3 служит эффективным средством ранней диагностики проблем и позволяет своевременно выявить неприменимость метода Вилфа [1].

Пример

Для демонстрации корректной работы метода Вилфа теперь будем рассматривать полином с заранее заданными корнями, не имеющий корней на границе прямоугольника. Возьмём полином:

$$P(z) = (z-(1+i))(z-(2+i))(z-(2+2i)) = z^3 - (5+4i)z^2 + (9+11i)z - (2+8i) \quad (75)$$

и прямоугольник с вершинами $Q_1 = 0$, $Q_2 = 3$, $Q_3 = 3+3i$, $Q_4 = 3i$. Корни этого полинома: $z_1 = 1+i$, $z_2 = 2+i$, $z_3 = 2+2i$, все они лежат строго внутри прямоугольника.

Проверим выполнение условий метода Вилфа [1]:

- $z_1 = 1+i$: $1 \in (0,3)$, $1 \in (0,3)$ — внутри,
- $z_2 = 2+i$: $2 \in (0,3)$, $1 \in (0,3)$ — внутри,
- $z_3 = 2+2i$: $2 \in (0,3)$, $2 \in (0,3)$ — внутри.

Ни один корень не лежит на прямых границах прямоугольника.

С этим полиномом и прямоугольником будут проделаны те же самые действия, что и в предыдущем примере.

Шаг 1: Параметризация сторон прямоугольника

Для каждой стороны используем параметризацию по Вилфу [1]:

1. **Сторона 1 ($Q_1 \to Q_2$)**: $z(t) = t$, $t \in [0,3]$,
2. **Сторона 2 ($Q_2 \to Q_3$)**: $z(t) = 3 + i \cdot t$, $t \in [0,3]$,
3. **Сторона 3 ($Q_3 \to Q_4$)**: $z(t) = 3 + 3i - t$, $t \in [0,3]$,
4. **Сторона 4 ($Q_4 \to Q_1$)**: $z(t) = 3i - i \cdot t$, $t \in [0,3]$.

Шаг 2: Вычисление $P_R^{(k)}(t)$ и $P_I^{(k)}(t)$

Для каждой стороны вычисляем коэффициенты полиномов $P_R^{(k)}(t)$ (действительная часть) и $P_I^{(k)}(t)$ (мнимая часть) [1].

Сторона 1:

$$P_R^{(1)}(t) = t^3 - 5t^2 + 3t + 4, P_I^{(1)}(t) = -4t^2 + 13t - 8. \quad (76)$$

Сторона 2:

$$P_R^{(2)}(t) = -4t^2 + 11t - 5, P_I^{(2)}(t) = -1t^3 + 4t^2 - 5. \tag{77}$$

Сторона 3:

$$P_R^{(3)}(t) = -t^3 + 4t^2 + 3t - 8, P_I^{(3)}(t) = 5t^2 - 13t + 4. \tag{78}$$

Сторона 4:

$$P_R^{(4)}(t) = 5t^2 - 17t + 10, P_I^{(4)}(t) = t^3 - 5t^2 + 10. \tag{79}$$

Шаг 3: Построение последовательностей Штурма

Для каждой пары $(P_R^{(k)}(t), P_I^{(k)}(t))$ строим последовательность Штурма S_k с помощью алгоритма Евклида с отрицательными остатками [5, 8].

Сторона 1:

$$\begin{aligned}
f_1^{(1)}(t) &= P_R^{(1)}(t) = t^3 - 5t^2 + 3t + 4,\\
f_2^{(1)}(t) &= P_I^{(1)}(t) = -4t^2 + 13t - 8,\\
f_3^{(1)}(t) &= 4.6875t - 7.5,\\
f_4^{(1)}(t) &= -2.56 \quad \text{(константа)}.
\end{aligned} \tag{80}$$

Сторона 2:

$$\begin{aligned}
f_1^{(2)}(t) &= P_R^{(2)}(t) = -4t^2 + 11t - 5,\\
f_2^{(2)}(t) &= P_I^{(2)}(t) = -t^3 + 4t^2 - 5,\\
f_3^{(2)}(t) &= 4t^2 - 11t + 5,\\
f_4^{(2)}(t) &= -4.6875t + 6.5625,\\
f_5^{(2)}(t) &= 2.56 \quad \text{(константа)}.
\end{aligned} \tag{81}$$

Сторона 3:

$$\begin{aligned}
f_1^{(3)}(t) &= P_R^{(3)}(t) = -t^3 + 4t^2 + 3t - 8,\\
f_2^{(3)}(t) &= P_I^{(3)}(t) = 5t^2 - 13t + 4,\\
f_3^{(3)}(t) &= -7.44t + 9.12,\\
f_4^{(3)}(t) &= 4.4225 \quad \text{(константа)}.
\end{aligned} \tag{82}$$

Сторона 4:

$$\begin{aligned}
f_1^{(4)}(t) &= P_R^{(4)}(t) = 5t^2 - 17t + 10,\\
f_2^{(4)}(t) &= P_I^{(4)}(t) = t^3 - 5t^2 + 10,\\
f_3^{(4)}(t) &= -5t^2 + 17t - 10,\\
f_4^{(4)}(t) &= 7.44t - 13.2,\\
f_5^{(4)}(t) &= -4.4225 \quad \text{(константа)}.
\end{aligned} \tag{83}$$

Шаг 4: Вычисление $V_k(0)$ и $V_k(L_k)$

Для каждой последовательности S_k вычисляем количество перемен знака $V_k(t)$ в точке t [5].

Сторона 1 ($L_1 = 3$): Вычисляем значения полиномов последовательности Штурма в точках $t = 0$ и $t = 3$:

$$\begin{aligned} \text{При } t = 0: & \\ f_1^{(1)}(0) &= 1 \cdot 0^3 - 5 \cdot 0^2 + 3 \cdot 0 + 4 = 4 \quad (\text{знак } +), \\ f_2^{(1)}(0) &= -4 \cdot 0^2 + 13 \cdot 0 - 8 = -8 \quad (\text{знак } -), \\ f_3^{(1)}(0) &= 4.6875 \cdot 0 - 7.5 = -7.5 \quad (\text{знак } -), \\ f_4^{(1)}(0) &= -2.56 \quad (\text{знак } -), \\ \text{Знаки (без нулей):} \quad & [+, -, -, -], \\ V_1(0) &= 1 \end{aligned} \tag{84}$$

$$\begin{aligned} \text{При } t = 3: & \\ f_1^{(1)}(3) &= 1 \cdot 27 - 5 \cdot 9 + 3 \cdot 3 + 4 = -5 \quad (\text{знак } -), \\ f_2^{(1)}(3) &= -4 \cdot 9 + 13 \cdot 3 - 80 = -5 \quad (\text{знак } -), \\ f_3^{(1)}(3) &= 4.6875 \cdot 3 - 7.5 = 6.5625 \quad (\text{знак } +), \\ f_4^{(1)}(3) &= -2.56 \quad (\text{знак } -), \\ \text{Знаки (без нулей):} \quad & [-, -, +, -], \\ V_1(3) &= 2 \end{aligned} \tag{85}$$

$$I_1 = V_1(0) - V_1(3) = 1 - 2 = -1. \tag{86}$$

Аналогично для остальных сторон:

- Сторона 2 ($L_2 = 3$): $I_2 = V_2(0) - V_2(3) = 1 - 3 = -2$.
- Сторона 3 ($L_3 = 3$): $I_3 = V_3(0) - V_3(3) = 1 - 2 = -1$.
- Сторона 4 ($L_4 = 3$): $I_4 = V_4(0) - V_4(3) = 1 - 3 = -2$.

Шаг 5: Суммирование индексов Коши

Вычисляем сумму всех четырёх индексов Коши:

$$\sum_{k=1}^{4} I_k = I_1 + I_2 + I_3 + I_4 = -6. \tag{87}$$

Шаг 6: Применение формулы Вилфа

Применяем формулу Вилфа (63) [1]:

$$N(P, R) = -\frac{1}{2} \sum_{k=1}^{4} I_k = -\frac{1}{2} \times (-6) = 3. \tag{88}$$

Шаг 7: Проверка результата по принципу аргумента

Для проверки корректности результата вычислим изменение аргумента $P(z)$ вдоль границы прямоугольника по принципу аргумента [11, 12]. Для этого разобьём каждую сторону на достаточно большое количество точек (например, $N = 1000$) и вычислим изменение аргумента как сумму приращений [3].

Сторона 1 ($Q_1 \to Q_2$, $z(t) = t$, $t \in [0, 3]$): Вычисляем значения $P(z(t))$ в равномерно распределённых точках $t_i = \frac{3i}{N}$, $i = 0, \ldots, N$:

$$\begin{gathered} P(z(t)) = t^3 - (5+4i)t^2 + (9+11i)t - (2+8i), \\ \arg(P(z(t))) = \arctan\left(\frac{\Im(P(z(t)))}{\Re(P(z(t)))}\right). \end{gathered} \tag{89}$$

Вычисляем изменение аргумента как сумму конечных разностей с учётом возможных скачков через $\pm\pi$:

$$\Delta_{Q_1Q_2} \arg P(z) = \sum_{i=1}^{N} \mathrm{wrap}(\arg(P(z(t_i))) - \arg(P(z(t_{i-1})))). \tag{90}$$

где функция $\mathrm{wrap}(\Delta\theta)$ корректирует разность углов, чтобы она лежала в интервале $(-\pi, \pi]$ [3]:

$$\mathrm{wrap}(\Delta\theta) = \begin{cases} \Delta\theta - 2\pi, & \text{если } \Delta\theta > \pi \\ \Delta\theta + 2\pi, & \text{если } \Delta\theta < -\pi \\ \Delta\theta, & \text{иначе} \end{cases}. \tag{91}$$

Численный расчёт даёт:

$$\Delta_{Q_1Q_2} \arg P(z) \approx 5.034140 \text{ рад.} \tag{92}$$

Сторона 2 ($Q_2 \to Q_3$, $z(t) = 3 + i \cdot t$, $t \in [0, 3]$): Аналогично вычисляем:

$$\begin{aligned} P(z(t)) &= (3+it)^3 - (5+4i)(3+it)^2 + (9+11i)(3+it) - (2+8i) \\ &= (-t^3 - 4t^2 + 11t - 5) + i(4t^2 - t^3 - 5). \end{aligned} \tag{93}$$

После аналогичных вычислений получаем:

$$\Delta_{Q_2Q_3} \arg P(z) \approx 5.034140 \text{ рад} \tag{94}$$

Сторона 3 ($Q_3 \to Q_4$, $z(t) = 3 + 3i - t$, $t \in [0, 3]$): Для $z(t) = 3 + 3i - t$:

$$\begin{aligned} P(z(t)) &= (3+3i-t)^3 - (5+4i)(3+3i-t)^2 + (9+11i)(3+3i-t) - (2+8i) \\ &= (-t^3 + 4t^2 + 3t - 8) + i(5t^2 - 13t + 4). \end{aligned} \tag{95}$$

Вычисляем изменение аргумента:

$$\Delta_{Q_3Q_4} \arg P(z) \approx 4.390638 \text{ рад} \tag{96}$$

Сторона 4 ($Q_4 \to Q_1$, $z(t) = 3i - i \cdot t$, $t \in [0, 3]$): Для $z(t) = 3i - it$:

$$\begin{aligned} P(z(t)) &= (3i - it)^3 - (5 + 4i)(3i - it)^2 + (9 + 11i)(3i - it) - (2 + 8i) \\ &= (5t^2 - 17t + 10) + i(t^3 - 5t^2 + 10). \end{aligned} \tag{97}$$

Вычисляем изменение аргумента:

$$\Delta_{Q_4 Q_1} \arg P(z) \approx 4.390638 \text{ рад.} \tag{98}$$

Полное изменение аргумента

Суммируем изменения аргумента по всем четырём сторонам:

$$\begin{aligned} \Delta_{\partial R} \arg P(z) &= \sum_{k=1}^{4} \Delta_{Q_k Q_{k+1}} \arg P(z) \\ &= 5.03414 + 5.03414 + 4.390638 + 4.390638 = 18.849556 \text{ рад.} \end{aligned} \tag{99}$$

Применение принципа аргумента

Согласно принципу аргумента (2) [11, 12], число нулей полинома $P(z)$ внутри контура ∂R равно:

$$N(P, R) = \frac{1}{2\pi} \Delta_{\partial R} \arg P(z) = \frac{1}{2\pi} \cdot 18.849556. \tag{100}$$

Вычисляем:

$$N(P, R) = \frac{18.849556}{2\pi} = \frac{18.849556}{6.283185307179586} \approx 3.000000. \tag{101}$$

Полученное значение в точности равно 3, что подтверждает правильность результата, полученного методом Вилфа [1].

Визуальная интерпретация

Изменение аргумента на каждой стороне можно интерпретировать как угол поворота вектора $P(z)$ в комплексной плоскости при движении z вдоль соответствующей стороны прямоугольника [11]:

- На стороне 1: вектор $P(z)$ совершает почти полный оборот (≈ 5.03 рад $\approx 288°$).
- На стороне 2: аналогично, ≈ 5.03 рад $\approx 288°$).
- На сторонах 3 и 4: ≈ 4.39 рад $\approx 251°$).

Суммарный поворот составляет 18.85 рад $\approx 1080° = 3 \times 360°$, что соответствует трём полным оборотам вокруг начала координат, указывая на наличие трёх нулей полинома внутри прямоугольника [11].

Вывод

Оба метода дают одинаковый результат: $N(P, R) = 3$, что соответствует действительному количеству корней полинома $P(z)$ внутри прямоугольника R:

$$z_1 = 1 + i, \quad z_2 = 2 + i, \quad z_3 = 2 + 2i. \tag{102}$$

Таким образом, метод Вилфа успешно определил количество нулей полинома в заданном прямоугольнике [1], подтверждая корректность реализации алгоритма и правильность применения формулы (63).

Особенности и замечания

1. **Точность**

   Формула (63) даёт *точное* целое число нулей, если вычисления проводятся точно [1].

2. **Кратные нули на границе**

   Если $P(z)$ имеет ноль на ∂R, то $\gcd(P_R^{(k)}, P_I^{(k)})$ не будет константой, и алгоритм обнаружит это через нарушение критериев S1-S3 [1, 5].

3. **Численная устойчивость**

   При вычислениях с плавающей точкой возможна потеря точности при определении знаков полиномов. Алгоритм Вилфа включает механизм обнаружения таких ситуаций (см. раздел 4) [1, 3].

4. **Сложность**

   Построение последовательности Штурма для стороны требует $O(n^2)$ операций, где n — степень полинома [1]. Подсчёт $V_k(t)$ в двух точках требует $O(n)$ операций [5].

Инициализация и выбор начального прямоугольника

Согласно оригинальной статье Вилфа (1978, стр. 4, "Start-Up Procedure") [1], начальный квадрат S выбирается следующим образом:

1. **Центр квадрата**

   Выбирается случайная точка внутри единичного квадрата с центром в начале координат.

2. **Размер стороны**

   Выбирается случайное положительное число.

3. **Проверка охвата**

   Вычисляется число нулей полинома $P(z)$ внутри S по формуле Вилфа (63) [1].

4. **Адаптивное увеличение**

   Если $N(P, S) < n$ (степени полинома), сторона квадрата удваивается, и проверка повторяется.

Ключевое уточнение

В статье явно указано, что центр квадрата выбирается случайным образом внутри квадрата $[0, 1] \times [0, 1]$ комплексной плоскости, а **не обязательно в начале координат**.

Интерпретация алгоритма

Алгоритм в псевдокоде

Algorithm 1 Инициализация начального квадрата (точная версия по Вилфу) [1]

Require: Полином $P(z)$ степени n (1)
Ensure: Квадрат S, содержащий все n нулей $P(z)$

// **Шаг 1: Выбор случайного центра в единичном квадрате**
$x_{\text{center}} \leftarrow \text{random}(0, 1)$
$y_{\text{center}} \leftarrow \text{random}(0, 1)$
$z_c \leftarrow x_{\text{center}} + i \cdot y_{\text{center}}$
// **Шаг 2: Выбор случайного размера**
$L \leftarrow \text{random}(0, L_{\max})$ ▷ Обычно $L_{\max} = 1$
// **Шаг 3: Построение квадрата**
$S \leftarrow \{z \in \mathbb{C} : |\Re(z) - \Re(z_c)| \leq L/2, |\Im(z) - \Im(z_c)| \leq L/2\}$
// **Шаг 4: Проверка и адаптация**
$N \leftarrow \text{CountZeros}(P, S)$ ▷ По формуле Вилфа (??) [1]
while $N < n$ **do**
    $L \leftarrow 2 \cdot L$ ▷ Удвоение стороны
    Перестроить S с тем же центром z_c и новой стороной L
    $N \leftarrow \text{CountZeros}(P, S)$
end while
return S, N

Математическое обоснование

1. **Избегание вырожденных случаев**

   Если корни полинома симметрично расположены относительно начала координат, квадрат с центром в (0,0) мог бы быть оптимальным. Однако для произвольных полиномов это не гарантировано [1].

2. **Статистическая устойчивость**

   Случайный выбор центра делает алгоритм статистически устойчивым к специально подобранным полиномам [1].

3. **Равномерное покрытие**

   При многократном запуске алгоритма случайный центр обеспечивает равномерное исследование комплексной плоскости.

Адаптивное увеличение размера

Математическое обоснование

Пусть $z_1, z_2, \ldots, z_n$ – корни полинома, и z_c – случайный центр квадрата. Обозначим:

$$R_{\max} = \max_{k=1}^{n} |z_k - z_c|. \tag{103}$$

Тогда квадрат со стороной L содержит все корни, если [1]:

$$L \geq 2R_{\max}. \tag{104}$$

Процесс удвоения гарантирует, что за конечное число шагов k:

$$L_0 \cdot 2^k \geq 2R_{\max} \quad \Rightarrow \quad k \geq \log_2 \left(\frac{2R_{\max}}{L_0} \right). \tag{105}$$

Эффективность по сравнению с кругом Коши

Круг Коши имеет радиус [6]:

$$R_{\text{Cauchy}} = 1 + \max_{k=0}^{n-1} \left| \frac{a_k}{a_n} \right|. \tag{106}$$

Для квадрата с центром в начале координат потребовалась бы сторона:

$$L_{\text{origin}} = 2R_{\text{Cauchy}}. \tag{107}$$

В методе Вилфа фактическая необходимая сторона [1]:

$$L_{\text{Wilf}} = 2 \cdot \max_k |z_k - z_c|. \tag{108}$$

Поскольку z_c выбирается случайно вблизи корней (в единичном квадрате), обычно [1]:

$$\max_k |z_k - z_c| \ll R_{\text{Cauchy}}. \tag{109}$$

Пример из статьи Вилфа с учётом случайного центра

В разделе 4 статьи рассматривается полином 5-й степени. Для инициализации [1]:

1. случайный центр z_c в $[0, 1] \times [0, 1]$;
2. начальная сторона L_0 – случайное число.;
3. удвоение продолжается, пока $N(P, S) < 5$.

Важно: в процессе удвоения **центр не меняется**, меняется только размер стороны. Это критически важно для корректной работы алгоритма [1].

Хранение информации о начальном квадрате

После определения начального квадрата S, он помещается в стек в формате [1]:

$$\begin{aligned}
\mathrm{NZ} &= n \quad \text{(количество нулей внутри, вычисленное по 63)},\\
\mathrm{COR} &= \left(\Re(z_c) - \frac{L}{2}\right) + i\left(\Im(z_c) + \frac{L}{2}\right),\\
\mathrm{D1} &= L, \quad \mathrm{D2} = L,\\
\{S_k\} &= \text{последовательности Штурма для 4 сторон (построенные как в 69)}.
\end{aligned}$$

где COR – северо-западный (левый верхний) угол квадрата.

Связь с контролем точности

Механизм инициализации области обнаруживает концептуальное сходство с процедурой контроля точности, подробно рассматриваемой в разделе 4 [1]. В обоих случаях применяются случайные модификации для преодоления численных затруднений, однако цели этих модификаций различны. При инициализации, если исходный квадрат оказывается слишком мал для гарантированного размещения всех корней, производится удвоение его размера. В механизме контроля точности, напротив, при обнаружении потери значащих цифр в вычислениях центр текущего прямоугольника подвергается случайному смещению в малой окрестности. Несмотря на различие целевых ситуаций, оба подхода объединяет использование элементов случайности для обеспечения робастности вычислительного процесса.

Вычислительная сложность инициализации

Каждая проверка числа нулей $N(P, S)$ в текущем квадрате требует выполнения двух основных этапов [1]. Во-первых, необходимо построение четырёх последовательностей Штурма — по одной на каждую сторону квадрата. Вычислительная сложность построения одной последовательности оценивается как $O(n^2)$ операций, где $n = \deg P$ (см. замечание в разделе 1). Во-вторых, для каждой стороны вычисляются значения $V_k(0)$ и $V_k(L)$, то есть числа перемен знака в последовательности Штурма на концах отрезка; эта процедура требует $O(n)$ операций на сторону.

Если в процессе инициализации потребовалось k последовательных удвоений размера квадрата, общая вычислительная сложность составит

$$O(k \cdot n^2). \tag{110}$$

На практике число удвоений k обычно не превышает 3–5, поэтому итоговая сложность инициализации остаётся в пределах $O(n^2)$ [1].

Заключение

Процедура инициализации Вилфа [1] характеризуется следующими особенностями:

- случайный выбор центра в единичном квадрате, а не фиксированное начало координат;
- адаптивное удвоение стороны до выполнения условия (104);
- минимальный размер начальной области, достаточный для охвата всех корней;
- статистическая устойчивость за счёт случайного выбора параметров.

Этот подход обеспечивает эффективное начало работы алгоритма без необходимости вычисления теоретических границ вроде круга Коши [6], что особенно важно для полиномов высоких степеней [1].

Разделение прямоугольника и управление стеком

Инвариант алгоритма и логика деления

Пусть прямоугольник $R \subset \mathbb{C}$ содержит $N(P, R)$ нулей полинома $P(z)$, вычисленных по формуле (63).

Ключевой инвариант алгоритма состоит в следующем:

> Каждый прямоугольник, находящийся в стеке, сопровождается точным числом нулей $N(P, R)$, расположенных внутри него.

Пусть R разделён на четыре подпрямоугольника R_1, R_2, R_3, R_4 непересекающимися внутренностями. Тогда из аддитивности интеграла в формуле принципа аргумента следует

$$N(P, R) = \sum_{j=1}^{4} N(P, R_j). \tag{111}$$

Доказательство

Из формулы принципа аргумента:

$$N(P, R) = \frac{1}{2\pi i} \int_{\partial R} \frac{P'(z)}{P(z)}\, dz. \tag{112}$$

Граница ∂R распадается на объединение границ подпрямоугольников с учётом ориентации. Внутренние линии разреза проходят дважды — в противоположных направлениях — и их вклады взаимно уничтожаются.

Следовательно,

$$\int_{\partial R} \frac{P'(z)}{P(z)}\, dz = \sum_{j=1}^{4} \int_{\partial R_j} \frac{P'(z)}{P(z)}\, dz, \tag{113}$$

что и даёт формулу (111). □

Таким образом, при делении прямоугольника число нулей распределяется по подпрямоугольникам без потерь и без «создания» новых нулей. Это и обеспечивает корректность рекурсивного алгоритма.

Структура элемента стека

Каждый элемент стека представляет собой структуру данных

$$(R,\ N(P, R),\ \mathcal{S}_R), \tag{114}$$

где:

- R — прямоугольник, задаваемый координатой северо-западного угла и размерами $(D1, D2)$;

- $N(P,R)$ — точное число нулей внутри R;
- $\mathcal{S}_R$ — набор последовательностей Штурма, соответствующих сторонам прямоугольника.

Алгоритм использует стек по принципу LIFO. Это означает, что деление происходит «вглубь», что соответствует естественной рекурсивной структуре задачи.

Прямоугольник помещается в стек только если $N(P,R) \geq 1$. Если $N(P,R) = 0$, область отбрасывается, так как она не содержит корней.

Алгоритм деления прямоугольника

Геометрия деления прямоугольника

Пусть прямоугольник R имеет северо-западный угол $\mathrm{COR} = x_{\mathrm{NW}} + iy_{\mathrm{NW}}$ и размеры $(D1, D2)$.

Центр прямоугольника равен

$$z_c = \left(x_{\mathrm{NW}} + \frac{D1}{2}\right) + i\left(y_{\mathrm{NW}} - \frac{D2}{2}\right). \tag{115}$$

Через точку z_c проводятся вертикальная и горизонтальная линии, делящие R на четыре подпрямоугольника.

Каждый из них имеет размеры

$$\frac{D1}{2} \times \frac{D2}{2}. \tag{116}$$

Такое деление гарантирует:

- непересечение внутренних областей,
- сохранение объединения:
$$R = \bigcup_{j=1}^{4} R_j, \tag{117}$$
- выполнение формулы аддитивности (111).

Эффективное переиспользование последовательностей Штурма

Ключевой момент в методе Вилфа (стр. 4) – эффективное переиспользование вычислений [1]. При делении прямоугольника R:

1. Уже имеются последовательности Штурма для сторон R (построенные ранее).
2. Для горизонтальной линии через z_c: используется разложение $P(z)$ в ряд Тейлора относительно z_c и строится одна новая последовательность Штурма.
3. Для вертикальной линии через z_c: разложение $P(z_c + it)$ даёт ещё одну последовательность Штурма.

4. Всего требуется построить только 2 новые последовательности вместо 8 (по 2 на каждую из 4 новых сторон).

Алгоритм в псевдокоде

Algorithm 2 Типичный шаг деления прямоугольника (по Вилфу) [1]

Require: Прямоугольник R с данными: NZ, COR, D1, D2, $\{S_k\}_{k=1}^4$
Ensure: Подпрямоугольники с NZ ≥ 1 помещаются в стек

// **Шаг 1: Извлечение данных из стека**
$(NZ, COR, D1, D2, \{S_k\}) \leftarrow \text{PopFromStack}()$
// **Шаг 2: Вычисление центра по (115)**
$x_c \leftarrow \Re(\text{COR}) + \frac{\text{D1}}{2}$
$y_c \leftarrow \Im(\text{COR}) - \frac{\text{D2}}{2}$ ▷ Вычитаем, так как COR – северо-западный угол
$z_c \leftarrow x_c + iy_c$
// **Шаг 3: Разложение в ряд Тейлора**
Coeff $\leftarrow$ TaylorExpansion(P, z_c) ▷ Алгоритм Taylor из [9]
CoeffRotated $\leftarrow$ Rotate90(Coeff) ▷ Коэффициенты $P(z_c + it)$
// **Шаг 4: Построение новых последовательностей Штурма**
$S_H \leftarrow$ BuildSturm(Coeff) ▷ Для горизонтальной линии через z_c
$S_V \leftarrow$ BuildSturm(CoeffRotated) ▷ Для вертикальной линии через z_c
// **Шаг 5: Формирование подпрямоугольников**
for $j \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$ **do**
    Определить COR$_j$, D1$_j$, D2$_j$ как указано выше
    Построить $\{S_k^{(j)}\}$ из имеющихся последовательностей S_k, S_H, S_V
    Вычислить $NZ_j = N(P, R_j)$ по формуле Вилфа (??) [1]
    if $NZ_j \geq 1$ **then**
        PushToStack$(NZ_j, \text{COR}_j, \text{D1}_j, \text{D2}_j, \{S_k^{(j)}\})$
    end if
end for

Построение новых последовательностей Штурма

При делении прямоугольника требуется определить изменение аргумента вдоль новых внутренних линий $z = z_c + t$ и $z = z_c + it$.

Для этого полином раскладывается в ряд Тейлора:

$$P(z_c + w) = \sum_{k=0}^{n} b_k w^k. \tag{118}$$

Коэффициенты b_k вычисляются за $O(n^2)$ по алгоритму Нийенхёйса–Вилфа.

Подстановка $w = t$ даёт горизонтальную линию:

$$P(z_c + t) = P_R^{(H)}(t) + iP_I^{(H)}(t). \tag{119}$$

Подстановка $w = it$ даёт вертикальную линию:

$$P(z_c + it) = P_R^{(V)}(t) + iP_I^{(V)}(t). \tag{120}$$

Из каждой пары вещественных полиномов строится последовательность Штурма, позволяющая вычислить изменение аргумента и тем самым число нулей в соответствующих подпрямоугольниках.

Пример деления

Рассмотрим прямоугольник R с параметрами:

$$\begin{aligned} \mathrm{COR} &= 0 + 3i \quad \text{(северо-западный угол)}, \\ \mathrm{D1} &= 4, \quad \mathrm{D2} = 3, \\ \mathrm{NZ} &= 2 \quad \text{(содержит 2 корня)}. \end{aligned} \tag{121}$$

Шаг 1: Вычисление центра по (115)

$$z_c = \left(0 + \frac{4}{2}\right) + i\left(3 - \frac{3}{2}\right) = 2 + 1.5i. \tag{122}$$

Шаг 2: Деление на подпрямоугольники

$$\begin{aligned} R_I: &\quad \mathrm{COR} = 0 + 3i, \quad \mathrm{D1} = 2, \quad \mathrm{D2} = 1.5, \\ R_{II}: &\quad \mathrm{COR} = 2 + 3i, \quad \mathrm{D1} = 2, \quad \mathrm{D2} = 1.5, \\ R_{III}: &\quad \mathrm{COR} = 0 + 1.5i, \quad \mathrm{D1} = 2, \quad \mathrm{D2} = 1.5, \\ R_{IV}: &\quad \mathrm{COR} = 2 + 1.5i, \quad \mathrm{D1} = 2, \quad \mathrm{D2} = 1.5. \end{aligned}$$

Шаг 3: Подсчёт нулей

Допустим, вычисления по формуле Вилфа (63) дали:

$$\begin{aligned} N(P, R_I) &= 0, \\ N(P, R_{II}) &= 1, \\ N(P, R_{III}) &= 0, \\ N(P, R_{IV}) &= 1. \end{aligned} \tag{123}$$

Тогда в стек помещаются только R_{II} и R_{IV}.

Управление стеком и критерии остановки

Алгоритм использует стек прямоугольников. Каждый элемент стека имеет вид

$$(R,\ N(P, R),\ \mathcal{S}_R), \tag{124}$$

где R — прямоугольник, $N(P, R)$ — число нулей внутри него, $\mathcal{S}_R$ — набор последовательностей Штурма для его сторон.

Стек работает по принципу LIFO. Это означает, что после деления прямоугольника алгоритм сразу углубляется в один из полученных подпрямоугольников. Тем самым реализуется рекурсивный обход «в глубину».

Прямоугольник помещается в стек тогда и только тогда, когда

$$N(P, R) \geq 1. \tag{125}$$

Если $N(P, R) = 0$, область немедленно отбрасывается.

Алгоритм завершает работу, когда выполнено одно из условий:

1. Стек пуст — все области, содержащие корни, локализованы;
2. Размер прямоугольника стал меньше заданного порога точности;
3. Достигнута заранее установленная максимальная глубина деления.

Таким образом, стек хранит только те области, в которых гарантированно находятся корни, и автоматически отбрасывает пустые области.

Вычислительная сложность

Рассмотрим один шаг деления прямоугольника для полинома степени n.

1. Разложение в ряд Тейлора

Разложение

$$P(z_c + w) = \sum_{k=0}^{n} b_k w^k \tag{126}$$

вычисляется за $O(n^2)$ операций.

2. Построение новых последовательностей Штурма

Для горизонтальной и вертикальной линий строятся две последовательности Штурма. Каждая требует $O(n^2)$ операций, так как выполняется последовательность делений полиномов степени не выше n.

3. Подсчёт нулей в подпрямоугольниках

После построения последовательностей вычисление знаковых вариаций для каждого из четырёх подпрямоугольников требует $O(n)$ операций.

Следовательно, асимптотическая сложность одного шага деления равна

$$O(n^2). \tag{127}$$

Если выполнено k шагов деления, то общая сложность алгоритма составляет

$$O(k\,n^2). \tag{128}$$

На практике число шагов деления пропорционально количеству корней и логарифму требуемой точности, то есть

$$k = O\left(n \log \frac{1}{\varepsilon}\right). \tag{129}$$

Тем самым метод имеет полиномиальную сложность по степени полинома.

Особенности реализации

Компактное хранение последовательностей Штурма

При построении последовательности Штурма возникает рекуррентное соотношение

$$S_{k-1}(x) = q_k(x)S_k(x) - S_{k+1}(x). \tag{130}$$

Для хранения достаточно сохранять частные $q_k(x)$, так как остальные элементы последовательности могут быть восстановлены обратным проходом.

Таким образом, для одной последовательности достаточно $2n$ регистров, что делает метод экономным по памяти.

Граничные случаи

- Если $N(P, R) = 0$, прямоугольник не делится.
- Если $N(P, R) = 1$, деление продолжается до достижения требуемой точности локализации.
- Если внутри области находится кратный корень, его кратность автоматически учитывается в формуле подсчёта нулей.

Контроль точности и условия остановки

Встроенная проверка точности

Одной из ключевых особенностей метода Вилфа является встроенный механизм контроля точности вычислений. Алгоритм автоматически обнаруживает потерю значащих цифр и соответствующим образом реагирует.

Математическая основа контроля

Из формулы Вилфа для подсчёта нулей в прямоугольнике R (см. формулу (63)) [1]:

$$N(P, R) = \frac{1}{2} \sum_{k=1}^{4} \left[V_k(L_k) - V_k(0) \right]. \tag{131}$$

левая часть $N(P, R)$ является целым числом. Следовательно, сумма в правой части должна быть чётным числом.

Критерий контроля точности

Если при вычислении суммы

$$S = \sum_{k=1}^{4} \left[V_k(L_k) - V_k(0) \right]. \tag{132}$$

получается нечётное число, это свидетельствует о потере значащих цифр в вычислениях [1].

Алгоритм контроля точности

Основная процедура

Algorithm 3 Контроль точности при делении прямоугольника (по Вилфу) [1]

Require: Прямоугольник R, полином $P(z)$
Ensure: Корректное деление или вывод прямоугольника

```
Разделить R на 4 подпрямоугольника R_1, R_2, R_3, R_4
for j = 1 to 4 do
    Вычислить сумму S_j = Σ_{k=1}^{4} [V_k^(j)(L_k^(j)) − V_k^(j)(0)]
    if S_j нечётно then
        // Обнаружена потеря точности
        loss_detected ← true
        bad_rectangles ← bad_rectangles ∪ {j}
    end if
end for
if не loss_detected then
    return Продолжить обычное деление
else
    Выполнить процедуру восстановления точности
end if
```

Процедура восстановления точности

При обнаружении потери точности (когда один или несколько S_j нечётны), алгоритм предпринимает следующие действия (стр. 4-5) [1]:

1. **Случайное смещение центра**

   Вместо использования точного центра z_c, выбирается новая точка в небольшой окрестности истинного центра.

2. **Повторное деление**

   Прямоугольник R делится с новым центром.

3. **Многократные попытки**

   Процедура повторяется до 3 раз.

4. **Завершение при неудаче**

   Если после 3 попыток не удаётся получить все чётные суммы, прямоугольник R считается окончательным и выводится как результат.

Геометрическая интерпретация проблемы точности

"Зона тумана"вокруг кратных корней

В разделе 4 статьи Вилф объясняет концепцию "зоны тумана"(fog zone) вокруг кратных корней [1]. Для корня z^* кратности p:

$$P(z) \sim c_p(z - z^*)^p \quad (c_p \neq 0). \tag{133}$$

Если вычисления ведутся с d десятичными цифрами, и K – модуль наибольшего коэффициента $P(z)$ относительно начала координат, то "зона тумана"имеет радиус:

$$R_f \sim 10^{-d/p}|K/c_p|^{1/p}. \tag{134}$$

Пример из статьи Вилфа

В разделе 4 приведён конкретный пример полинома 5-й степени [1]:

$$P(z) = z^5-(13.999+5i)z^4+\cdots = (z-(1+i))^2(z-(4-3i))(z-(4+3i))(z-(3.999+3i)). \tag{135}$$

При вычислениях с двойной точностью (16 цифр) были получены следующие результаты:

Корень	Действительная часть (ошибка)	Мнимая часть (ошибка)
1	4.0000000000 ($\pm 3 \times 10^{-15}$)	-3.0000000000 ($\pm 3 \times 10^{-15}$)
2	3.9999999999 ($\pm 4 \times 10^{-11}$)	2.9999999999 ($\pm 2 \times 10^{-11}$)
3	3.9989999999 ($\pm 1 \times 10^{-11}$)	3.0000000000 ($\pm 1 \times 10^{-11}$)
4	0.99999998109 ($\pm 5 \times 10^{-8}$)	0.99999997949 ($\pm 5 \times 10^{-8}$)
5	1.00000002282 ($\pm 3 \times 10^{-8}$)	1.00000002142 ($\pm 3 \times 10^{-8}$)

Анализ результатов

1. **Простые корни** (1, 2, 3): точность порядка 10^{-11}–10^{-15}.
2. **Кратный корень** (двойной, корни 4 и 5): точность порядка 10^{-8}.
3. **"Зона тумана"**: для двойного корня при $d = 16$, $p = 2$, используя формулу (134) [1]:
$$R_f \sim 10^{-16/2} = 10^{-8}, \tag{136}$$
что соответствует фактической точности 5×10^{-8}.
4. **Разделение кратного корня**: Алгоритм фактически разделил двойной корень на два близких простых [1]

Условия остановки алгоритма

Алгоритм завершает обработку прямоугольника R в одном из следующих случаев:

1. $N(P, R) = 0$ — область не содержит корней;
2. $N(P, R) \geq 1$ и
$$\min(D1, D2) < \varepsilon, \tag{137}$$
где ε — заданная точность;
3. после нескольких попыток восстановления чётность не восстанавливается.

В последнем случае размер прямоугольника служит оценкой погрешности положения корней.

Алгоритм остановки

Algorithm 4 Процедура остановки (по Вилфу) [1]

Require: Прямоугольник R с данными: NZ, COR, D1, D2
Require: Параметры: ϵ (точность), max_attempts = 3
Ensure: Решение о продолжении или остановке

```
attempt ← 0
success ← false
while attempt < max_attempts и не success do
    Разделить R с (возможно смещённым) центром
    Проверить чётность сумм для 4 подпрямоугольников
    if все суммы чётные then
        success ← true
    else
        Случайно сместить центр в малой окрестности
        attempt ← attempt + 1
    end if
end while
if success then
    return Продолжить деление
else
    Вывести прямоугольник R как результат
    Вывести NZ как количество корней внутри
    // Примечание: размер R даёт оценку погрешности
end if
```

Особые случаи и обработка ошибок

Кратные корни на границе

Особые трудности возникают в ситуации, когда корень полинома лежит непосредственно на границе прямоугольника или находится к ней очень близко [1]. В этом случае значение $P(z)$ на одной из сторон прямоугольника становится чрезвычайно малым по модулю. Поскольку вычисление величин $V_k(t)$ основано на анализе изменения аргумента функции вдоль стороны, малость $P(z)$ приводит к ухудшению устойчивости вычислений и усилению влияния округления. В результате могут возникать ошибки при определении изменения аргумента, что отражается на суммарной величине

$$S = \sum_{k=1}^{4} \left[V_k(L_k) - V_k(0) \right]. \tag{138}$$

Такие численные нарушения обнаруживаются автоматически, поскольку сумма S должна быть чётной. Если в вычислениях возникает нечётное значение, это интерпретируется как признак потери точности (см. критерий (132)). В ответ алгоритм выполняет смещение центра прямоугольника в небольшой окрестности исходного положения и повторяет процедуру деления. Таким образом устраняется неблагоприятная геометрическая конфигурация, при которой корень оказывается слишком близко к границе.

Очень близкие корни

Другой проблемной ситуацией является наличие нескольких корней, расстояние между которыми меньше радиуса так называемой «зоны тумана» [1]. В условиях конечной точности вычислений такие корни становятся практически неразличимыми. Алгоритм в этом случае может не суметь разделить их при последовательных делениях прямоугольника. Тогда итоговый прямоугольник будет содержать более одного корня, и соответствующее значение NZ окажется больше единицы. Это не является ошибкой метода, а отражает объективное ограничение машинной точности: корни расположены настолько близко, что дальнейшее их разделение потребовало бы увеличения разрядности вычислений.

Вычислительные аспекты

Стоимость контроля точности

Процедура контроля чётности сумм требует вычисления величин $V_k(t)$ в двух точках для каждой из четырёх сторон прямоугольника [1]. Поскольку вычисление одного значения $V_k(t)$ имеет трудоёмкость порядка $O(n)$ для полинома степени n, общая стоимость проверки составляет $4 \times 2 \times O(n)$, что эквивалентно $O(n)$. Дополнительные операции суммирования четырёх разностей и проверки чётности требуют лишь постоянного числа арифметических действий, то есть имеют сложность $O(1)$. Следовательно, полный контроль точности для одного прямоугольника выполняется за время $O(n)$, что существенно меньше затрат на само деление, имеющее сложность порядка $O(n^2)$.

Влияние на общую сложность

Введение механизма контроля точности приводит к увеличению числа шагов деления, поскольку алгоритм может продолжать разбиение после первой обнаруженной потери точности [1]. Однако при этом каждый дополнительный шаг не требует полного перестроения всех последовательностей, используемых для вычисления индексов, что снижает относительные затраты на повторные операции. В итоге достигается компромисс: общее число операций возрастает, но обеспечивается более глубокое деление и, как следствие, существенно более высокая итоговая точность локализации корней.

Заключение

Механизм контроля точности, предложенный Вилфом [1], обеспечивает автоматическое обнаружение потери значащих цифр без введения дополнительных эмпирических порогов. Алгоритм не только фиксирует факт ухудшения вычислений, но и предпринимает попытки восстановить корректность путём адаптивного смещения центра прямоугольника. Размер итогового прямоугольника служит естественной оценкой достигнутой точности локализации корня или группы корней. Благодаря этому метод остаётся применимым как к простым, так и к кратным корням, сохраняя устойчивость и надёжность при конечной машинной точности. Именно сочетание автоматического контроля, адаптивности и информативности делает алгоритм особенно ценным в задачах, где требуется гарантированная корректность результата.

Вывод результатов и учёт кратности

Формирование конечных результатов

После завершения работы алгоритма, то есть при опустошении стека прямоугольников или выполнении одного из условий остановки, формируется окончательный набор выходных данных. Согласно статье Вилфа [1], результатом является система неперекрывающихся прямоугольников R_i, каждый из которых гарантированно содержит фиксированное число корней полинома. Это число определяется величиной NZ_i, вычисленной на основе формулы Вилфа.

Для каждого выходного прямоугольника естественным приближением к корню (или группе корней) служит его центр

$$z_{\text{approx}}^{(i)} = \text{center}(R_i), \tag{139}$$

поскольку корень гарантированно лежит внутри области R_i. Оценка погрешности определяется геометрическим размером прямоугольника и может быть принята равной половине его максимального линейного размера:

$$\epsilon_i = \frac{1}{2}\max(\mathrm{D1}_i, \mathrm{D2}_i). \tag{140}$$

Таким образом, каждый результат задаётся тройкой характеристик: приближением $z_{\text{approx}}^{(i)}$, оценкой точности ϵ_i и числом корней NZ_i, заключённых в прямоугольнике.

Особенности работы с кратными корнями

Корректность учёта кратности основана на принципе аргумента [11]. Согласно формуле

$$\frac{1}{2\pi}\Delta_{\partial R}\arg P(z) = \sum_{z_j \in R} m_j, \tag{141}$$

приращение аргумента вдоль границы прямоугольника равно сумме кратностей всех корней внутри него. Следовательно, величина NZ автоматически учитывает кратность, а не только количество различных корней. Именно поэтому, как подчёркивает Вилф во введении [1], метод находит все корни вместе с их кратностями.

Если прямоугольник содержит корень кратности p, то соответствующее значение NZ равно p. При дальнейших делениях алгоритм пытается локализовать корни более точно. Если корни действительно совпадают (истинная кратность), разделение невозможно, и прямоугольник будет уменьшаться до предела машинной точности. Если же корни лишь очень близки, но различны, то при достаточной точности вычислений алгоритм способен разделить их на отдельные прямоугольники.

Иллюстрация на примере из статьи

В примере Вилфа для полинома пятой степени [1] наблюдается показательный эффект обработки двойного корня $(1+i)$. Численно он был представлен двумя очень близкими приближениями. Расстояние между ними оказалось порядка 6×10^{-8}, что согласуется с оценкой радиуса «зоны тумана» при двойной точности ($d = 16$, $p = 2$), получаемой из формулы (134):

$$R_f \sim 10^{-16/2} = 10^{-8}. \tag{142}$$

Таким образом, разделение кратного корня на два близких значения отражает фундаментальное ограничение машинной арифметики и не противоречит теории метода.

Постобработка результатов

При необходимости повышения точности возможна дополнительная обработка найденных приближений [1]. Один из подходов заключается в использовании арифметики повышенной разрядности с повторным применением алгоритма деления. Альтернативой является применение локальных итерационных методов, например метода Ньютона [2], к начальному приближению $z^{(i)}_{\text{approx}}$. Поскольку алгоритм Вилфа обеспечивает глобальную локализацию корней, такие итерационные методы работают устойчиво и быстро сходятся.

Проверка корректности результатов может выполняться различными способами [3, 11]. В частности, можно вычислить значение $P(z^{(i)}_{\text{approx}})$ и убедиться, что оно мало по модулю. Дополнительно можно сопоставить произведение линейных множителей, построенных по найденным приближениям, с исходным полиномом (1). Также используется проверка формул Виета, например сравнение суммы найденных корней с величиной $-a_{n-1}/a_n$.

Вычислительная сложность и оптимизации

Анализ сложности

Теоретическая оценка

Пусть $P(z)$ — полином степени n.

Подсчёт числа нулей в одном прямоугольнике R включает:

- построение последовательностей Штурма для соответствующих вещественных полиномов степени $\leq n$ — $O(n^2)$ [5];
- вычисление величин $V_k(t)$ в формуле Вилфа (131) — $O(n)$.

Следовательно,

$$T_{\text{count}}(n) = O(n^2). \tag{143}$$

Один шаг деления прямоугольника требует конечного числа таких операций, поэтому

$$T_{\text{split}}(n) = O(n^2) \quad [1]. \tag{144}$$

При обработке полинома степени n общее число шагов деления имеет порядок $O(n)$–$O(n^2)$ в зависимости от распределения корней. Вилф даёт итоговую оценку общего времени работы

$$T(n) = O(n^3) \quad [1]. \tag{145}$$

Заключение

Итоги метода Вилфа

Метод глобальной бисекции в комплексной плоскости, предложенный Вилфом [1], представляет собой детерминированный алгоритм нахождения всех нулей полинома с комплексными коэффициентами. Его принципиальное отличие от итерационных схем состоит в глобальном характере: алгоритм не требует задания начальных приближений и не зависит от локальных свойств функции вблизи предполагаемых корней.

Надёжность метода обеспечивается использованием формулы подсчёта нулей в прямоугольнике (131), основанной на изменении аргумента. Эта формула даёт точное число корней (с учётом кратности) внутри области. Кратные корни корректно учитываются благодаря формуле (141), в которой изменение аргумента равно сумме кратностей всех нулей внутри контура [11].

Существенным преимуществом алгоритма является отсутствие дефляции: на всех этапах вычисления ведутся с исходным полиномом. Это повышает численную устойчивость по сравнению с методами, в которых после нахождения каждого корня выполняется деление полинома на соответствующий множитель. Дополнительным элементом надёжности служит встроенный механизм контроля точности, основанный на анализе чётности суммы (132), что позволяет автоматически обнаруживать потерю значащих цифр [1].

Области применения и ограничения

Метод Вилфа целесообразно применять в задачах, где требуется гарантированное нахождение всех корней полинома, включая кратные и комплексные. Он особенно полезен в ситуациях, когда отсутствует информация о распределении корней и невозможно задать разумные начальные приближения. Благодаря глобальному характеру алгоритм не зависит от выбора стартовой точки и не страдает от сходимости к «нежелательным» решениям.

В то же время метод имеет ограничения. Его асимптотическая трудоёмкость порядка $O(n^3)$ [1] делает его менее эффективным по сравнению с некоторыми специализированными алгоритмами при больших степенях полинома. Кроме того, точность локализации кратных корней ограничивается явлением «зоны тумана», описываемым формулой (134), что отражает фундаментальные ограничения конечной машинной арифметики. Алгоритм также требует хранения стека прямоугольников и промежуточных данных, связанных с последовательностями Штурма, что увеличивает объём используемой памяти.

Перспективы развития

Метод Вилфа сохраняет актуальность и в современных вычислительных системах. Перспективным направлением является построение гибридных алгоритмов, в которых глобальная локализация корней осуществляется методом деления прямоугольников, а окончательное уточнение — быстрыми локальными методами, например методом Ньютона. Такая комбинация позволяет объединить глобальную надёжность с высокой скоростью локальной сходимости.

Другим направлением является параллельная реализация, поскольку обработка различных прямоугольников не требует обмена информацией и может выполняться независимо. Кроме того, возможны адаптивные стратегии управления делением области, учитывающие распределение корней и численные свойства полинома.

Заключительные замечания

Метод Вилфа представляет собой строго обоснованный алгоритм, объединяющий классические результаты комплексного анализа и алгебры с вычислительной процедурой гарантированной локализации корней. Несмотря на более высокую асимптотическую сложность по сравнению с некоторыми современными методами, его глобальная сходимость, корректный учёт кратности и встроенный механизм контроля точности делают его надёжным инструментом в задачах, где приоритетом является полнота и достоверность результата [1, 3].

Список литературы

[1] **Wilf H. S.** A global bisection algorithm for computing the zeros of polynomials in the complex plane // Journal of the ACM (JACM). — 1978. — Vol. 25. — No. 3. — P. 415–420.

[2] **Березин И. С., Жидков Н. П.** Методы вычислений: В 2-х т. — М.: Физматлит, 1962. — Т. 1. — 464 с.

[3] **Калиткин Н. Н.** Численные методы. — 2-е изд., перераб. и доп. — М.: Наука, 1978. — 512 с.

[4] **Фаддеев Д. К., Фаддеева В. Н.** Вычислительные методы линейной алгебры. — М.: Физматлит, 1963. — 656 с.

[5] **Воеводин В. В.** Численные методы алгебры (теория и алгоритмы). — М.: Наука, 1966. — 248 с.

[6] **Гантмахер Ф. Р.** Теория матриц. — 5-е изд. — М.: Физматлит, 2004. — 560 с.

[7] **Корн Г., Корн Т.** Справочник по математике для научных работников и инженеров. — М.: Наука, 1973. — 832 с.

[8] **Демидович Б. П., Марон И. А.** Основы вычислительной математики. — М.: Наука, 1970. — 664 с.

[9] **Штёр Й., Бульирш Р.** Введение в численный анализ: В 2-х т. — М.: Мир, 1988.

[10] **Pan V. Ya.** (Пан В. Я.) Computation of Polynomial Zeros and Matrix Eigenvalues // Computers & Mathematics with Applications. — 2000. — Vol. 40. — P. 41–76.

[11] **Marden M.** Geometry of Polynomials. — 2nd ed. — Providence, RI: American Mathematical Society, 1966. — 243 p.

[12] **Henrici P.** Applied and Computational Complex Analysis: Volume 1: Power Series, Integration, Conformal Mapping, Location of Zeros. — New York: Wiley, 1974. — 682 p.

[13] **Alefeld G., Herzberger J.** Introduction to Interval Computations. — New York: Academic Press, 1983. — 334 p.

[14] **Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.** Numerical Recipes: The Art of Scientific Computing. — 3rd ed. — Cambridge: Cambridge University Press, 2007.