

Содержание

1	Введение	2
2	Обзор и доказательная база метода	3
2.1	Основные определения и обозначения	3
2.2	Критерий сходимости и оценка ошибки	3
2.3	Точки старта	4
2.4	SPA - Sums of Powers Algorithm	7
3	Обзор некоторых составных частей алгоритма	9
3.1	Метод Бернулли - классическое описание	9
3.2	Вывод метода Бернулли через суммы Ньютона	9
3.3	Сходимость метода Бернулли	10
3.4	Метод Бернулли для нескольких/комплексных корней	10
3.5	Метод Бернулли для сопряжённых комплексных корней	11
3.6	Метод Бернулли для нескольких кратных корней	12
3.7	Расширенный метод Бернулли	14
4	Верхняя и нижняя оценки точности для LaSPA	15
5	Заключение	16
5.1	О методе Бернулли	16
5.2	О методе LaSPA	16
6	Список литературы	17

1 Введение

Метод Лаггера часто используется как практически безошибочный вариант во многих случаях, когда нужно быстро и сравнительно надёжно вычислить корни многочлена. Однако есть случаи, когда метод сойтись не может, либо встречает на пути к достаточному приближению препятствия. Алгоритм сумм степеней (Sums of Powers Algorithm) позволяет подстраховать метод Лаггера в этих случаях, вычисляя корни с достаточной точностью, и самое главное, с полной надёжностью. Алгоритм LaSPA (Laguerre SPA) является быстрым, и, в то же время, чрезвычайно надёжным методом для вычисления корней многочленов.

2 Обзор и доказательная база метода

2.1 Основные определения и обозначения

Пусть есть многочлен

$$p(z) = \sum_{i=0}^m a_i z^i, a_i \in \mathbb{C}, i \in \{1, \dots, m\} \quad (1)$$

Здесь a_i - коэффициенты, $m + 1$ - степень многочлена.

Для определённости сразу скажем, что в общем случае, нас интересуют многочлены степени больше 3. Если какие-то части алгоритма встречаются с многочленами степеней ниже, они пользуются аналитической формулой для квадратного или линейного многочлена.

2.2 Критерий сходимости и оценка ошибки

Для любого простого (не кратного) корня ρ многочлена $p(z)$ с помощью разложения в ряд Тейлора можно доказать, что метод Лаггера сходится кубически к пределу ρ вне зависимости от близости начального приближения. Однако не была известна окрестность этого корня (насколько малое расстояние необходимо для этого условия). В следующей теореме мы устанавливаем, что для каждого простого корня ρ существует диск с центром в точке ρ , в котором любое начальное приближение обеспечивает гарантированную сходимость метода.

Теорема 1. Пусть $p(z)$ - нормализованный многочлен степени больше 3, $P = \rho_1, \dots, \rho_l$ - множество корней. Тогда для схемы Лаггера действует следующее:

Если есть простой корень $\rho \in P$ такой, что выполняется условие

$$|u_0 - \rho| \leq \frac{1}{2m-1} \min\{|\sigma - \rho|, \sigma \in P \setminus \{\rho\}\} \quad (2)$$

Тогда последовательность Лаггера $L_p(u_0)$ сходится к пределу ρ , выполняется

$$|u_n - \rho| < \lambda^n |u_0 - \rho|, \quad (3)$$

где $\lambda = \frac{15}{16} \forall n \in \mathbb{N}$

Таким образом, обеспечивается, по крайней мере, линейная сходимость на всех дисках простых корней.

Теорема 2. Пусть $p(z)$ - нормализованный многочлен степени больше 3. Тогда

$$k(0) = \min\{k \in J_m | a_k 0\} \quad (4)$$

и

$$q_0 = m(k(0) \cdot a_{k(0)})^{-1} \quad (5)$$

Пусть $k(s)$ и q_s известны для $s = \max\{0, n - m\}, \dots, n - 1$. Тогда $k(n)$ и q_n можно вычислить следующим образом:

Для $s = \max\{0, n - m\}, \dots, n - 1$ и для $t \in J_m$, возьмём

$$\begin{cases} q_s, s > 0, q_s 0, \\ 1, s > 0, q_s = 0, \\ t(k(0)a_{k(0)}^{-1}, s = 0, \end{cases} \cdot \begin{cases} q_s, s > 0, q_s 0, \\ 1, s > 0, q_s = 0, \\ t(k(0)a_{k(0)}^{-1}, s = 0, \end{cases} . \quad (6)$$

2.3 Точки старта

Требуется показать области выполнения неравенства:

$$|u_0 - \rho| \leq \frac{1}{2m-1} \min\{|\sigma - \rho| \mid \sigma \in P \setminus \{\rho\}\} \quad (7)$$

В большинстве случаев выгодно взять арифметическое среднее корней:

$\alpha = -\frac{1}{m}a_{m-1}a_m^{-1}$. Если для α верно $q(\alpha) + s(\alpha)r(\alpha) \neq 0$ и оно лежит в квадрате, то это оптимальная точка старта. Возможные точки старта алгоритма могут быть разделены на 4 класса:

1. внутренние точки - в них обычный алгоритм Лагерра сойдётся гарантированно за определённо взятое количество шагов;
2. граничные точки, или точки шага - возможно, потребуется больше итераций, однако, алгоритм сойдётся;
3. циклические точки - генерируют циклические последовательности Лаггера;
4. сингулярные точки - нули производной многочлена, как начальное приближение не используется из-за мгновенного выхода алгоритма в бесконечность.

Приведём свойства этих точек и областей, которые они образуют на комплексной плоскости, рассматриваемой на предмет выполнения неравенства (7):

- если у многочлена нет точек сингулярности, существует такое число S_p , что стартовые точки одного типа с модулем больше S_p формируют неограниченные просто связные области;
- если ζ - точка сингулярности или циклическая точка, то для любого номера $N \in \mathbb{N}$ существует такой диск δ_N с центром в ζ , что все количества шагов для траекторий с начальной точкой $u_0 \in \delta_N \setminus \zeta$ больше N . Иначе говоря, выбраться алгоритмом Лагерра из точек цикла и сингулярности невозможно, каждая итерация приводит к увеличению количества необходимых шагов;
- только циклические и сингулярные точки являются изолированными, то есть, их окрестность пересекается с множеством выполнения неравенства только в них;
- Граничная кривая каждой области ступеньки полностью или частично принадлежит один из соседних районов

Теперь рассмотрим многочлен с корнями

$$[1.6 - 0.55j, -0.39 + 0.03j, -2.32 + 2.17j, 0.2 - 1.06j, -0.02 - 0.27j]$$

Это многочлен 5 степени

$$(0.306917 - 0.531751i) - (1.61433 + 2.57707i)x - \\ -(6.2503 + 1.40811i)x^2 - (0.5818 - 5.8351i)x^3 + (0.93 - 0.32i)x^4 + x^5$$

График корней на комплексной плоскости: Визуализируем все типы точек на графике:

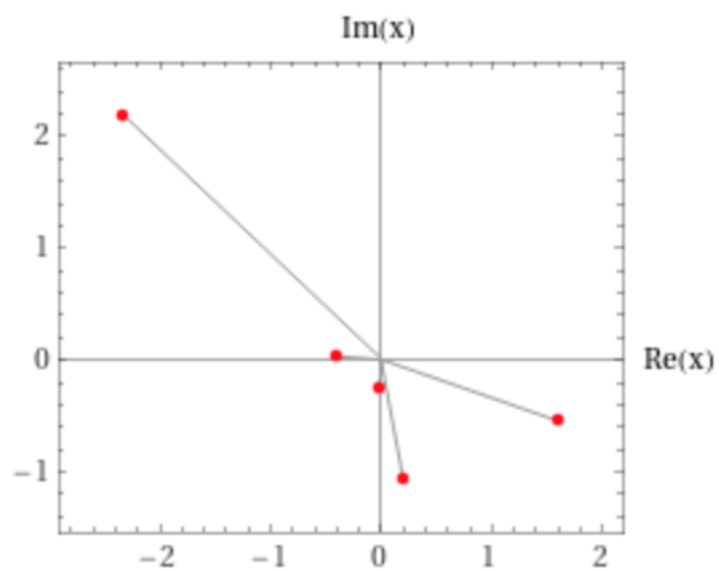


Рис. 1: Корни многочлена на комплексной плоскости

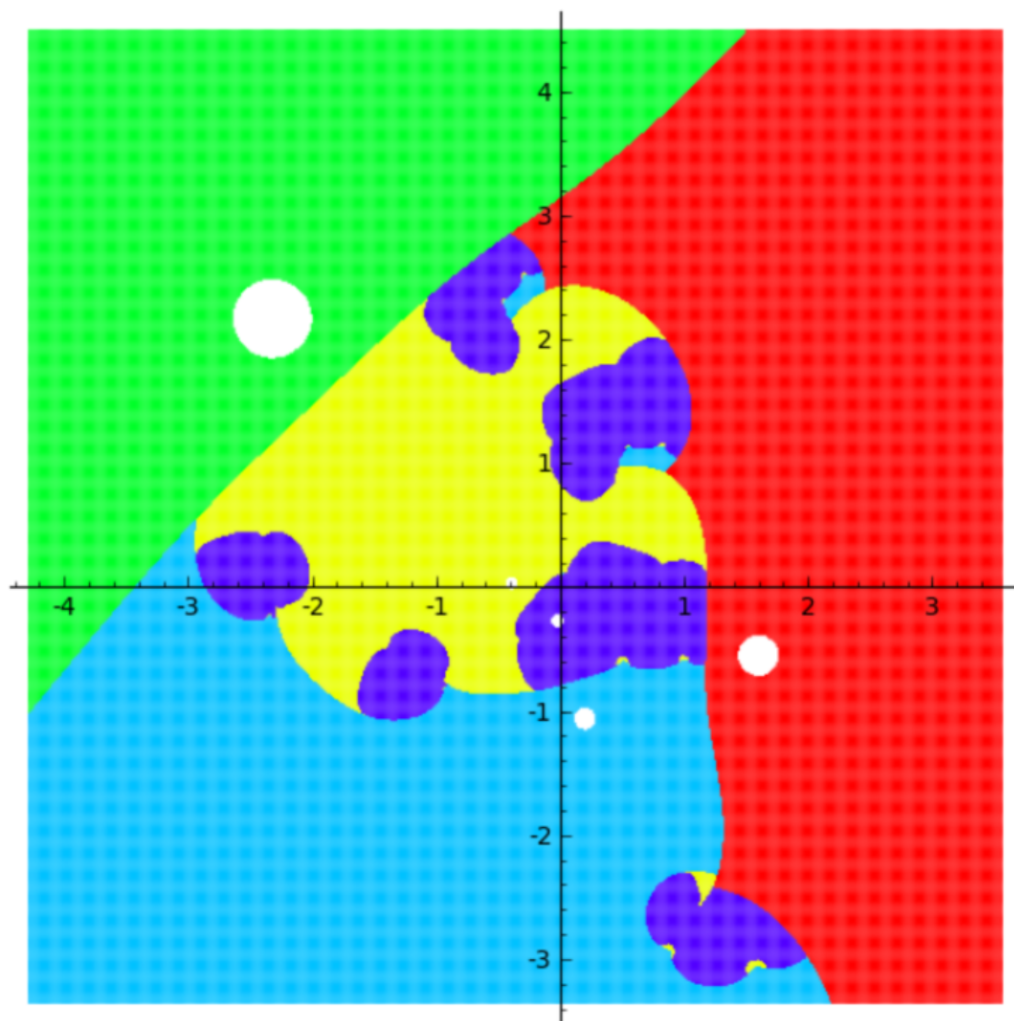


Рис. 2: Безопасные (белым) и предельные зоны

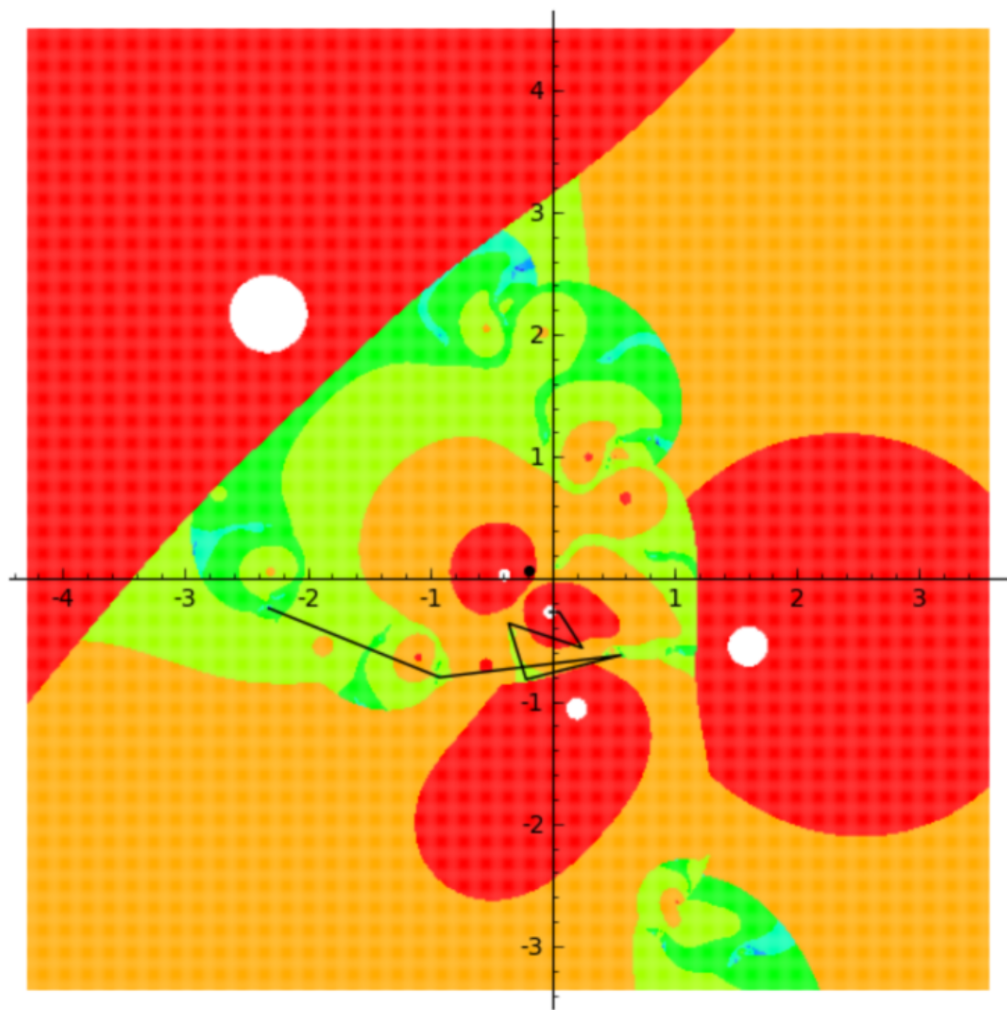


Рис. 3: Безопасные зоны, области шага, траектория наибольшего количества шагов, среднее арифметическое корней (чёрным цветом)

2.4 SPA - Sums of Powers Algorithm

Краткое описание алгоритма:

1. Производится попытка выполнения обычного метода Лаггера. Для улучшения скорости сходимости стоит выбрать наиболее оптимальное начальное приближение без учёта остальных свойств многочлена - среднее арифметическое корней. Его можно найти по формуле $-\frac{1}{m}a_{m+1}a_m^{-1}$.
2. Если количество итераций, необходимое для получения результата, достаточного по точности, даёт неудовлетворительную точность, применяем алгоритм SPA непосредственно.
3. Из пункта 1 получаем значения приближений корней. Как правило, они бывают уже достаточно близкие к настоящим значениям. Расставляем их по возрастанию их модуля: $0 < |z_1| < |z_2| < \dots < |z_k|$. Это необходимо, чтобы упорядочить их по радиусу от точки отсчёта.
4. Используем для полученных приближений метод наименьших кругов. Это должно получить достаточные по заданной точности ϵ приближения корней для подавляющего числа многочленов (по крайней мере, на практике не встречалось иного)
5. В так называемом *экстремальном случае* используется метод колец Турана. Он гарантированно находит кольцо, в котором находится искомым корень, причём кольцо достаточно тонкое, чтобы удовлетворять заданной точности.

Теперь остановимся на каждом шаге поподробнее:

1. Вначале используется обычный метод Лаггера. Для большинства многочленов ожидается, что он сойдётся - нулями около точек сингулярности и циклов обладают не так много многочленов, как показывает практика. Для улучшения скорости сходимости стоит выбрать наиболее оптимальное начальное приближение без учёта остальных свойств многочлена - среднее арифметическое корней. Его можно найти по формуле $-\frac{1}{m}a_{m+1}a_m^{-1}$.
2. Если количество итераций, необходимое для получения результата, достаточного по точности, даёт неудовлетворительную точность, применяем алгоритм SPA непосредственно.
3. Из пункта 1 получаем значения приближений корней. Как правило, они бывают уже достаточно близкие к настоящим значениям. Расставляем их по возрастанию их модуля: $0 < |z_1| < |z_2| < \dots < |z_k|$. Это необходимо, чтобы упорядочить их по радиусу от точки отсчёта.
4. Начинаем алгоритм наименьших кругов. Для начала скажем, что все корни многочлена $p(z)$ образуют на аналитическом пространстве $\{(x, y, w) \in \mathbb{R}^3 | w = |p(x + iy)|^2\}$. Следовательно, пользуясь методом наименьших кругов, будем получать сдвиги многочленов близкие к кругу минимальных расстояний - минимальному кругу без точек минимума. Берём функцию

$$\pi_r := (t \rightarrow |p(rcos t + irsin t)|^2, t \in [0, 2\pi]) \quad (8)$$

получаем разложение в ряд Фурье:

$$\pi_r(t) - f_0 = \sum_{k=1}^m (f_k 2\cos(kt) + g_k 2\cos(\frac{3\pi}{2} + kt))$$

затем получаем значение производной

$$\pi_r(t)' = \sum_{k=1}^m (kg_k 2\cos(kt) - kf_k 2\cos(\frac{3\pi}{2} + kt))$$

в равноудалённых точках t по таблице значений $2\cos(\pi n/(3N))$, где $n = 0, \dots, 6N - 1$, получаемыми рекурсивно:

$$2\cos((n+2)\Delta) = (2\cos\Delta)(2\cos((n+1)\Delta)) - 2\cos(n\Delta)$$

где $n = 0, \dots, 3N - 1$, $\Delta := \frac{\pi}{3N}$. Далее вычисляются точки $x_j, j \in \{1, \dots, e\}$, в которых $\pi'_r(x_j) \leq 0$, но $\pi'_r(x_j + \Delta) > 0$. В этих точках вычисляется точка минимума: $r\cos x_j + i r\cos(3\pi/2 + x_j), j = \{1, \dots, e\}$. Эти точки аппроксимируют локальный минимум. Назовём их точками минимума. Их можно использовать как смещение для многочлена.

5. Вычислим для многочлена диаметр Лаггера. Если для всех корней он превышает заданную точность, то выходим на пункт 6. Иначе берём каждый, для которого диаметр Лаггера меньше ϵ , то есть, для которого выполняется условие сходимости, как достаточное приближение корня. Будем называть минимальным кругом множество точек круга с минимальным расстоянием в совокупности с минимальными точками x_j . Так как корни близки к кругу, стоит снизить шаг алгоритма, приближающего корни (Бернулли) до 2.0.
6. Пусть найдена точка с минимальным диаметром Лаггера (то есть, максимально ограничивающая приближение корня). Смещаем многочлен на неё, берём радиус минимального круга r_1 для смещённого многочлена. Если $r_1 < 0.3r$, то идём дальше, строим минимальный круг для полученного. Это называется цепочка минимальных кругов.
7. Если для двоих кругов подряд не выполнится $r_1 < 0.3r$, возникает особая ситуация, требующая заменить минимальный круг кругом Турана. Это множество следующего вида:

$$\mathcal{T}(\xi) := \{z \in \mathbb{C} \mid |z - \xi| = \sigma_{16m}\}$$

Его радиус равен $r' = \sigma_{16m}$, и для него верно $0.9r' < 0.2^{\frac{1}{16}}r' < |z_1| < r'$. Получается кольцо Турана, покрываемое 12 дисками следующего вида:

$$\{z \in \mathbb{C} \mid |z - \xi_j| \leq \nu r'\} \tag{9}$$

где

$$\xi_j := (19r'/20)\cos(j\pi/6) + i\sin(j\pi/6), j = 0, \dots, 11, \tag{10}$$

$$\nu = \sqrt{1/400 + (19/5)\sin^2(\pi/24)} \tag{11}$$

z_1 находится в одном из этих дисков.

3 Обзор некоторых составных частей алгоритма

3.1 Метод Бернулли - классическое описание

Метод Бернулли основан на том, чтобы, решая линейное дифференциальное уравнение с коэффициентами как у многочлена, получить аппроксимацию его нулей.

Пусть имеется многочлен

$$p(z) = c_n z^n + c_{n-1} z^{n-1} + c_1 z + c_0 \quad (12)$$

с нулями $\xi_1 \dots \xi_n$. Тогда, если рассматривать дифференциальное уравнение

$$c_m x^m + c_{m-1} x^{m-1} + c_0 x^{n-m} = 0 (c_n \neq 0; m = n, n+1, \dots) \quad (13)$$

Общее решение выглядит следующим образом:

$$x_m = c_1 \xi_1^m + c_2 \xi_2^m + \dots + c_n \xi_n^m \quad (14)$$

3.2 Вывод метода Бернулли через суммы Ньютона

Пусть теперь имеется многочлен

$$p(z) = c_n z^n + c_{n-1} z^{n-1} + c_1 z + c_0 \quad (15)$$

где $c_n = 1$. Пусть также имеется сумма степеней, выражающаяся следующим образом:

$$S_k = \sum_{v=1}^n \xi_v^k \quad (16)$$

то есть

$$S_0 = n \quad (17)$$

$$S_1 = \xi_1 + \xi_2 + \dots + \xi_n \quad (18)$$

$$S_2 = \xi_1^2 + \xi_2^2 + \dots + \xi_n^2 \quad (19)$$

$$\dots \quad (20)$$

Тогда мы можем найти суммы степеней S_i , не находя сами решения многочлена ξ_k путём отношений:

$$S_m + c_{n-1} S_{m-1} + \dots + c_{n-m+1} S_1 + m c_{n-m} = 0 (m = 1, \dots, n) \quad (21)$$

Это же для номеров $n > m$:

$$S_m + c_{n-1} S_{m-1} + \dots + c_0 S_{m-n} = 0 \quad (22)$$

Это верно, так как:

$$f(z) = \prod_{j=1}^n (z - \xi_j) \quad (23)$$

Прологарифмировав и продифференцировав впоследствии обе части уравнения, получаем:

$$f'(z) = f(z) \sum_{j=1}^n \frac{1}{z - \xi_j} = f(z) \sum_{j=1}^n \frac{1}{z} \frac{1}{1 - \frac{\xi_j}{z}} = f(z) \sum_{j=1}^n \frac{1}{z} \sum_{v=0}^{\infty} \frac{\xi_j^v}{z^v} \quad (24)$$

$$= \frac{f(z)}{z} \sum_{v=0}^{\infty} z^{-v} \sum_{j=1}^n \xi_j^v = \frac{f(z)}{z} \sum_{v=0}^{\infty} \frac{S_v}{z^v} = \frac{f(z)}{z} \sum_{v=0}^{\infty} \frac{S_v}{z^v} \quad (25)$$

Сумма в конце сходится к $|z| > \text{Max}_j |\xi_j|$. Таким образом:

$$zf'(z) = nz^n + (n-1)z^{n-1}c_{n-1} + \dots + c_1z = (z^n + c_{n-1}z^{n-1} + \dots + c_0) \sum_{v=0}^{\infty} \frac{S_v}{z^v} \quad (26)$$

Для $1 \leq m \leq n$ коэффициент для z^{n-m} это $(n-m)c_{n-m}$ в левой части уравнения, в правой:

$$nc_{n-m} + c_{n-m+1}S_1 + c_{n-m+2}S_2 + \dots + c_{n-1}S_{m-1} + S_m \quad (27)$$

Приравняв коэффициенты, получим (21). Аналогично, для $m > n$ получим:

$$S_m + c_{n-1}S_{m-1} + \dots + c_0S_{m-n} \quad (28)$$

Здесь уже получаем (22).

3.3 Сходимость метода Бернулли

Метод сходится линейно:

$$\left| \frac{x_{j+1}}{x_j} - \xi_j \right| \leq 2n|\xi_1| \left| \frac{\xi_2}{\xi_1} \right|^{j+1} \quad (29)$$

3.4 Метод Бернулли для нескольких/комплексных корней

Краткое описание алгоритма:

1. задаются начальные значения $f(-2), f(-1), f(0), f(1)$. лучше взять $f(-2) = 0, f(-1) = 0, f(0) = 0, f(1) = 1$ для простоты начальных вычислений
2. Вычисляется столбец f , используя коэффициенты многочлена
3. Вычисляется столбец f_2 как определитель, используя значения из столбца f
4. Далее уже по другой формуле, с делением на определитель из предыдущего столбца, вычисляются f_r

Algorithm 1: Метод Бернулли для нескольких/комплексных корней

Пусть m - степень многочлена, a - массив его коэффициентов

Задаются "краевые условия" $f(-m), f(-m+1), \dots, f(0)$

for $t = 1$ to $iter$ **do**

$f(t) = (a[1] * f(t-1) + a[2] * f(t-2) + \dots + a[m-1] * f(t-m+1)) / a[0]$

end for

for $t = 1$ to $iter - 1$ **do**

$f_2(t) = f(t)^2 - f(t-1) * f(t+1)$

end for

for $r = 3$ to m **do**

for $t = 1$ to $iter - r - 1$ **do**

$f_{r+1}(t) = (f_r(t)^2 - f_r(t-1) * f_r(t+1)) / f_{r-1}(t)$

end for

end for

for $r = 1$ to m **do**

$x_r = f_r(iter-1) / f_r(iter-2)$

end for

3.5 Метод Бернулли для сопряжённых комплексных корней

Различие с предыдущим алгоритмом заключается лишь в том, что последовательность Z_r не сходится, когда алгоритм встречается с парой сопряжённых комплексных корней. В таком случае считается отдельный столбец $c(t)$ и $(c(t+1)c(t)+1)/2c(t)$, из которого уже получается значение для двух корней.

Algorithm 2: Метод Бернулли для сопряженных комплексных корней

```
Пусть  $m$  - степень многочлена,  $a$  - массив его коэффициентов
Задаются "краевые условия"  $f(-m), f(-m+1), \dots, f(0)$ 
for  $t = 1$  to  $iter$  do
     $f(t) = (a[0] * f(t-1) + a[1] * f(t-2) + \dots + a[m-1] * f(t-m+1)) / a[0]$ 
end fortry:
for  $t = 1$  to  $iter - 1$  do
     $f_2(t) = f(t)^2 - f(t-1) * f(t+1)$ 
end for
if  $|f_2(iter-1) - f_2(iter-2)| > |f_2(iter-2) - f_2(iter-3)|$  then
    не сошлись по шагу 2. нужно рассматривать на сопряжённые корни
    for  $t = 1$  to  $iter - 4$  do
         $c(t) = f_2(t) / (r * Z_1)$ 
    end for
    for  $t = 1$  to  $iter - 4$  do
         $ct(t) = (c(t) * c(t+1) + 1) / (2 * c(t))$ 
    end for
    данная итерация гарантированно сходится

else
    идём дальше
end if
for  $r = 3$  to  $m$  do
    for  $t = 1$  to  $iter - r - 1$  do
         $f_{r+1}(t) = (f_r(t)^2 - f_r(t-1) * f_r(t+1)) / f_{r-1}(t)$ 
    end for
end for
for  $r = 1$  to  $m$  do
     $x_r = f_r(iter-1) / f_r(iter-2)$ 
end for
```

3.6 Метод Бернулли для нескольких кратных корней

Данный алгоритм использует наработки предыдущего, при этом рассматривает даже не сходящиеся (либо сходящиеся слишком медленно) последовательности f_r . Если они сходятся медленно, то можно сделать вывод, что алгоритм столкнулся с кратными корнями. В таком случае, можно вычислить столбец $c(t)$. Если указанная выше функция $(c(t) * c(t+1) + 1) / (2 * c(t)) \rightarrow 1$, то делаем вывод, что, с точки зрения алгоритма, есть два сопряжённых комплексных корня с практически нулевой мнимой частью. Если кратных корней больше, то больше и больше столбцов f_r будут сходиться медленно. Можно использовать достаточное число итераций и затем просто оценить корень с помощью некоторого Z_r , где r - кратность корня, но лучше проверить гипотезу о кратности следующим способом если выражение:

$$(1 - C^r) \rightarrow 0 \quad (30)$$

где

$$C^r = c(t)c(t+1)c(t+2) \dots c(t+r-1) \quad (31)$$

В таком случае достаточно будет лишь вычислить

$$\frac{Z_{m+p-1}}{Z_{m+1}} \quad (32)$$

Algorithm 3: Метод Бернулли для кратных корней

Пусть m - степень многочлена, a - массив его коэффициентов, eps - некоторое условие сходимости, выраженное числом, например, 1

Задаются "краевые условия" $f(-m)$, $f(-m+1)$, \dots , $f(0)$

for $t = 1$ to $iter$ **do**

$f(t) = (a[1] * f(t-1) + a[2] * f(t-2) + \dots + a[m-1] * f(t-m+1)) / a[0]$

end fortry:

for $t = 1$ to $iter - 1$ **do**

$f_2(t) = f(t)^2 - f(t-1) * f(t+1)$

end for

if $|f_2(iter-1) - f_2(iter-2)| > |f_2(iter-2) - f_2(iter-3)|$ **then**

не сошлись по шагу 2. нужно рассматривать на сопряжённые корни

for $t = 1$ to $iter - 4$ **do**

$c(t) = f_2(t) / (r * Z_1)$

end for

for $t = 1$ to $iter - 4$ **do**

$ct(t) = (c(t) * c(t+1) + 1 / (2 * c(t)))$

end for

данная итерация гарантированно сходится

else if $|f_2(iter-1) - f_2(iter-2)| - |f_2(iter-2) - f_2(iter-3)| < eps$ **then**

if $1 - C^p \rightarrow 0$ **then**

получаем p корней по формуле Z_{m+p-1} / Z_{m+1}

end if

else

идём дальше

end if

for $r = 3$ to m **do**

for $t = 1$ to $iter - r - 1$ **do**

$f_{r+1}(t) = (f_r(t)^2 - f_r(t-1) * f_r(t+1)) / f_{r-1}(t)$

if $|f_r(iter-1) - f_r(iter-2)| - |f_r(iter-2) - f_r(iter-3)| < eps$ **then**

if $(1 - C^p) \rightarrow 0$ **then**

получаем p корней по формуле Z_{m+p-1} / Z_{m+1}

end if

end if

end for

end for

for $r = 1$ to m **do**

$x_r = f_r(iter-1) / f_r(iter-2)$

end for

3.7 Расширенный метод Бернулли

В данной главе продемонстрируем расширенный метода Бернулли для аппроксимации к численно наибольшему корню алгебраического уравнения. На основе приведенного здесь расширения теперь становится возможным сделать метод Бернулли средством оценки не только наибольшего корня, но и всех корней уравнения, будь то вещественные, комплексные или многократные, с помощью арифметического процесса, хорошо приспособленного к механическим вычислениям, без какого-либо предварительного определения природы или положения корней. Метод Бернулли заключается в формировании последовательности решений линейного разностного уравнения, связанного с данным алгебраическим уравнением.

Пусть решаемое уравнение имеет вид:

$$a_0 z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_n = 0, \quad (33)$$

тогда рассматривается следующее разностное уравнение в $f(t)$:

$$a_0 f(t+n) + a_1 f(t+n-1) + a_2 f(t+n-2) + \dots + a_n f(t) = 0 \quad (34)$$

Последовательным n значениям f , а именно $f(t+n-1), f(t+n-2), \dots, f(t)$, присваиваются произвольные значения и по уравнению (31) находится $f(t+n)$. Далее n последовательных значений $f(t+n), f(t+n-1), \dots, f(t+1)$ служат аналогичным образом для определения $f(t+n+1)$. И путем повторения этого процесса получается последовательность решений разностного уравнения с аргументами, возрастающими на единицу. Тогда, если $|z_1|$ модуль наибольшего корня уравнения (30) больше модуля любого другого корня, то последовательность $f(t)$ будет стремиться стать геометрической прогрессией с знаменателем z_1 :

$$\lim_{t \rightarrow \infty} \frac{f(t+1)}{f(t)} = z_1$$

4 Верхняя и нижняя оценки точности для LaSPA

Пусть $Z := \{z_1, \dots, z_k\}$, $Z_1 := \{z \in \mathbb{C} \mid p'(z) = 0\}$, $h_m(u) := u - m \frac{p(u)}{p'(u)}$

Продemonстрируем границы ошибок и два критерия останковки. Они связаны с дисками Лагерра, которые, по определению, являются замкнутыми дисками, имеющие любые $u \in \mathbb{C} \setminus (Z \cup Z_1)$ и $h_m(u)$ в качестве конечных точек диаметра. Известно, что каждый диск Лагерра содержит хотя бы один корень из $p(z)$. Более того, радиус ограничен сверху и снизу постоянными кратными расстояния между $\{u\}$ и Z , если это расстояние достаточно мало.

Теорема 1. Если $u \in \mathbb{C} \setminus (Z \cup Z_1)$, то всякая окружность, проходящая через u и $h_m(u)$, обладает тем свойством, что либо внутри и вне этой окружности есть элементы Z , либо все элементы Z лежат на этой окружности.

В частности, замкнутый диск

$$\mathcal{L} = \mathcal{L}_u(p) := \{\omega \in \mathbb{C} \mid |\omega - h_{m/2}(u)| \leq |u - h_{m/2}(u)|\},$$

называемый диском Лагерра элемента u , содержит по крайней мере один корень из p .

Теорема 2. Пусть $p(z) = a_m \prod_{j=1}^k (z - z_j)^{m_j}$ многочлен с $k \geq 2$. Если $z_j \in Z$, то

$$|u - z_j| < m \left| \frac{p(u)}{p'(u)} \right| \leq \frac{2m - 1}{2m_j - 1} |u - z_j|$$

и $\mathcal{L}_u \cap Z = \{z_j\}$ выполняется для каждого $u \in \mathbb{C}$ при $0 < |u - z_j| \leq \mu_j/(2m)$.

Таким образом, радиус \mathcal{L}_u меньше $m|u - z_j|$, если u достаточно близко к z_j .

5 Заключение

5.1 О методе Бернулли

Стоит отметить, что метод Бернулли показывает себя хорошо на предоставленных результатах экспериментов. Условленной точности достигает за сравнительно небольшое число несложных итераций, легко параллелизируем. Не требует сложных функций (вычисления или знания производной, пусть даже для многочлена).

Простота реализации более общих методов, позволяющих вычислять на основе уже имеющихся и их сравнительно небольшая вычислительная сложность позволяет сказать, что метод легко генерализуется.

Помимо вышесказанного, метод также обладает большим количеством ускорений и упрощений.

5.2 О методе LaSPA

У алгоритма сумм степеней есть некоторые минусы:

1. довольно очевидная сложность вычислений в момент попадания в неудачный случай, компенсирующаяся редкостью таких случаев;
2. необходимость строго оптимизировать критически важные участки кода - например, процедуру сокращения многочлена

Но есть и плюсы:

1. надёжность - метод сойдётся для любого многочлена;
2. точность - аппроксимации корней будут найдены для указанной точности;

6 Список литературы

- Moeller, H. The Laguerre-and-Sums-of-Powers Algorithm for the Efficient and Reliable Approximation of All Polynomial Roots [Текст] / Н. Moeller // Проблемы передачи информации - 2023. - № 51.
- Moeller, H. Convergence and visualization of Laguerre's rootfinding algorithm - [Электронный ресурс] / Н. Moeller – Электрон. текстовые дан. - 2015. - Режим доступа: https://www.researchgate.net/publication/270450403_Convergence_and_visualization_of_Laguerre's_rootfinding_algorithm свободный.
- Aitken, A. C. On Bernoulli's Numerical Solution of Algebraic Equations - [Текст] / А. С. Aitken - 1926.
- Moeller, H. Convergence and visualization of Laguerre's rootfinding algorithm - [Электронный ресурс] / Н. Moeller – Электрон. текстовые дан. - 2015. - Режим доступа: https://www.researchgate.net/publication/259333181_Visualization_of_the_First_Stage_of_the_Laguerre's_rootfinding_algorithm свободный.