Постановка математической задачи для Θ – алгоритма

Дано: медленно сходящаяся последовательность (S_n), где (S_n) – частичные суммы ряда

$$S_n = \sum_{k=0}^{n} a_k \tag{1}$$

Условие на сходимость:

1. Последовательность (S_n) сходится к пределу S (т.е. $\lim_{n \to \infty} s_n = s$), но делает это медленно.

Возможные формы (S_n):

1. Экспоненциальная: $S_n = S + a\lambda^n + o(\lambda^n), |\lambda| < 1, \lambda \epsilon \mathbb{C}$.
2. Рациональная: $S_n = S + an^{-d} + o(n^{-d}), d > 0$, d$\epsilon\mathbb{R}$.
3. Смешанные из первых двух.

Цель: обеспечить более быструю сходимость ряда к S по сравнению с исходной последовательностью.

Θ – алгоритм

Как показано в работе Брезински [1] классический ε-алгоритм может быть представлен в виде:

$$\varepsilon_{(k+1)}^{(n)} = \varepsilon_{(k-1)}^{(n+1)} + D_k^{(n)}, D_k^{(n)} = \left(\varepsilon_{(k)}^{(n+1)} - \varepsilon_{(k)}^{(n)}\right)^{-1} \tag{2}$$

Нам важно то, что данная формула подчеркивает двухшаговую природу алгоритма, где каждый новый элемент зависит от элементов на двух предыдущих уровнях.

Применяя оператор конечной разности Δ, можно получить соотношение:

$$\Delta\varepsilon_{k+1}^{(n)} = \Delta\varepsilon_{k-1}^{(n+1)} + \Delta D_k^{(n)} \tag{3}$$

В данном контексте оператор Δ действует исключительно на верхние индексы n алгоритма, а не на значения последовательности. Для произвольной величины $X^{(n)}$, зависящей от индекса n, он определяется как:

$$\Delta X^{(n)} = X^{(n+1)} - X^{(n)} \tag{4}$$

Этот оператор анализирует динамику алгоритма, путем отслеживания изменений результатов при переходе от индекса n к $n + 1$.

Пример вычислений:

1. Для ε-алгоритма: $\Delta\varepsilon_{k+1}^{(n)} = \varepsilon_{k+1}^{(n+1)} - \varepsilon_{k+1}^{(n)}$.
2. Для $\Delta D_k^{(n)}$: $\Delta D_k^{(n)} = \Delta D_k^{(n+1)} - \Delta D_k^{(n)}$.

Далее рассмотрим условие ускорения сходимости:

$$\lim_{n\to\infty} \frac{\Delta\varepsilon_{2k+2}^{(n)}}{\varepsilon_{2k}^{(n+1)}} = 0 \tag{5}$$

Для его выполнения необходимо и достаточно ([1] глава 2.9), чтобы:

$$\lim_{n\to\infty} \frac{\Delta D_{2k+1}^{(n)}}{\Delta\varepsilon_{2k}^{(n+1)}} = -1 \tag{6}$$

Доказательство следует из разложения отношения разностей и анализа предельного поведения компонент. Более подробно об этом пишет Брезински [1].

В случаях, когда условие (5) не выполняется, вводится дополнительный параметр ω_k:

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \omega_k D_{2k+1}^{(n)} \tag{7}$$

Оптимальным образом определить значение ω_k можно так:

$$\omega_k = -\lim_{n\to\infty} \frac{\Delta\varepsilon_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} \tag{8}$$

При таком выборе параметра ω_k последовательность (7) будет сходиться быстрее, чем $\Delta\varepsilon_{2k}^{(n)}$

На практике довольно часто вычисление предела затруднительно, поэтому можно использовать оценку [1, глава 2.9]:

$$\omega_k = -\frac{\Delta\varepsilon_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} \tag{9}$$

Рассмотрим полную схему Θ-алгоритма. Для удобства будем использовать обозначения Θ вместо ε.

Инициализация:

$$\Theta_{-1}^{(n)} = 0, \Theta_0^{(n)} = S_n \tag{10}$$

Рекуррентные правила:

$$\Theta_{2k+1}^{(n)} = \Theta_{2k-1}^{(n+1)} + D_{2k}^{(n)} \tag{11}$$

Формула (11) соответствует нечетному порядку $2k+1$ и напрямую следует из определения ε-алгоритма (формула (2)).

$$\Theta_{2k+2}^{(n)} = \Theta_{2k}^{(n+1)} + \omega_k D_{2k+1}^{(n)} = \Theta_{2k}^{(n+1)} - \frac{\Delta\Theta_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} D_{2k+1}^{(n)} \qquad (12)$$

Формула (12) соответствует четному порядку $2k+2$. При четных порядках необходимо ввести дополнительный параметр ω_k (описан в формуле (9)), который позволит лучше ускорять сходимость путем компенсации главного члена погрешности. Используя этот параметр как коэффициент при $D_{2k+1}^{(n)}$, можно добиться ускорения сходимости последовательности.

Иначе формулу четного порядка можно представить следующим образом:

$$\Theta_{2k}^{(n)} = \Theta_{2k-2}^{(n+1)} - \frac{(\Delta\Theta_{2k-2}^{(n+1)})^2}{\Delta^2\Theta_{2k-2}^{(n)}}, k \geq 1 \qquad (13)$$

$\Delta^2\Theta_{2k-2}^{(n)}$ называется второй разностью и вычисляется по формуле:

$$\Delta^2\Theta_{2k-2}^{(n)} = \Theta_{2k-2}^{(n+2)} - 2\Theta_{2k-2}^{(n+1)} + \Theta_{2k-2}^{(n)} \qquad (14)$$

Численные эксперименты [1] показали, что результаты работы $\Theta_2^{(n)}$ чаще всего почти так же хороши, как и лучшие результаты аналогичных алгоритмов.

Теорема 1.

Необходимое и достаточное условие того, что $\forall n, \Theta_2^{(n)} = S$, заключается в том, что (S_n) имеет одну из следующих форм:

1. Экспоненциальная:

$$S_n = S + (S_0 - S)\lambda^n, S \neq S_0, \lambda \neq 0{,}1 \qquad (15)$$

Данная последовательность сходится при условии $|\lambda| < 1$.

Доказательство:

Рассмотрим последовательность S_n из формулы (15), обозначив $(S_0 - S) = C \neq 0$.

$$\Theta_1^{(n)} = 0 + \frac{1}{C\lambda^{n+1} - C\lambda^n} = \frac{1}{C\lambda^n(\lambda - 1)} = \frac{\lambda^{-n}}{C(\lambda - 1)}$$

Учитывая формулу (13), получим:

$$\Theta_2^{(n)} = \Theta_0^{(n+1)} - \frac{(\Delta\Theta_0^{(n+1)})^2}{\Delta^2\Theta_0^{(n)}}$$

Вычислим $\Delta\Theta_0^{(n+1)}$:

$$\Delta\Theta_0^{(n+1)} = \Theta_0^{(n+2)} - \Theta_0^{(n+1)} = C\lambda^{n+2} - C\lambda^{n+1} = C\lambda^{n+1}(\lambda - 1)$$

$$\Theta_0^{(n+1)} = S_{n+1} = S + C\lambda^{n+1}$$
$$\Theta_0^{(n+2)} = S_{n+2} = S + C\lambda^{n+2}$$

Вторая разность $\Delta^2\Theta_0^{(n)}$ вычисляется как:

$$\Delta^2\Theta_0^{(n)} = \Theta_0^{(n+2)} - 2\Theta_0^{(n+1)} + \Theta_0^{(n)} = (S + C\lambda^{n+2}) - 2(S + C\lambda^{n+1}) + (S + \mathrm{C}\lambda^n)$$
$$\Delta^2\Theta_0^{(n)} = C\lambda^n(\lambda^2 - 2\lambda + 1)$$
$$\Delta^2\Theta_0^{(n)} = \mathrm{C}\lambda^n(\lambda - 1)^2$$

Выполнив подстановку, получим:

$$\Theta_2^{(n)} = S + C\lambda^{n+1} - \frac{\mathrm{C}^2\lambda^{2n+2}(\lambda-1)^2}{\mathrm{C}\lambda^n(\lambda-1)^2} = S + C\lambda^{n+1} - C\lambda^{n+1} = S$$

2. Рациональная:

$$S_n = S + (S_0 - S)\prod_{i=0}^{n-1}\left[1 - \frac{d}{i-m}\right], \text{где } S_0 \neq S, d \neq 1, m, m+d \notin \mathbb{Z} \quad (16)$$

Сходимость этой последовательности достигается тогда, когда вещественная часть d строго положительна.

Для удобства обозначим:

$$\mathrm{C} = S_0 - S \neq 0 \text{ и } P_n = \prod_{i=0}^{n-1}\left[1 - \frac{d}{i-m}\right]$$

Тогда формула (16) примет вид:

$$S_n = S + \mathrm{C}P_n$$

Согласно формуле (13) для четного порядка

$$\Theta_2^{(n)} = \Theta_0^{(n+1)} - \frac{(\Delta\Theta_0^{(n+1)})^2}{\Delta^2\Theta_0^{(n)}}$$

Выразим $\Theta_0^{(n+1)}$:

$$\Theta_0^{(n+1)} = S + \mathrm{C}P_{n+1}$$

Выразим $\Delta\Theta_0^{(n+1)}$:

$$\Delta\Theta_0^{(n+1)} = \Theta_0^{(n+2)} - \Theta_0^{(n+1)} = S_{n+2} - S_{n+1} = C(P_{n+2} - P_{n+1})$$
$$P_{n+2} - P_{n+1} = P_{n+1}\left(\left(1 - \frac{d}{(n+1)-m}\right) - 1\right) = -P_{n+1}\frac{d}{(n+1)-m}$$
$$\Delta\Theta_0^{(n+1)} = -CP_{n+1}\frac{d}{n+1-m}$$

Вторая разность $\Delta^2\Theta_0^{(n)}$ определяется как:

$$\Delta^2\Theta_0^{(n)} = \Theta_0^{(n+2)} - 2\Theta_0^{(n+1)} + \Theta_0^{(n)} = \mathrm{C}(P_{n+2} - 2P_{n+1} + P_n)$$

Выразим P_{n+2} и P_{n+1} через P_n:

$$P_{n+1} = P_n\left(1 - \frac{d}{n+1-m}\right)$$
$$P_{n+1} = P_n\frac{n-m-d}{n-m}$$

$$P_{n+2} = P_n\left(1 - \frac{d}{n+1-m}\right)\left(1 - \frac{d}{n+1-m}\right)$$

Подставим P_{n+2} и P_{n+1} в $\Delta^2\Theta_0^{(n)}$ и после упрощения получим:

$$\Delta^2\Theta_0^{(n)} = CP_n\frac{d(d+1)}{(n-m)(n+1-m)}$$

После подстановки имеющих значений в формулу (13) получается $\Theta_2^{(n)} = S$ за счет взаимного уничтожения других членов.

3. Специальные вырожденные случаи при

$$S_0 = S, S_n = S + (S_1 - S)\prod_{i=0}^{n-1}\left(1 - \frac{d}{i}\right) \text{ для } n \geq 1, \text{где } S_1 \neq S, d \notin \mathbb{Z} \qquad (17)$$

Сходимость этой последовательности достигается тогда, когда вещественная часть d строго положительна. Доказательство аналогично рациональному случаю и использует рекуррентное свойство произведения. Подробное доказательство теоремы можно найти в книге Брецинского [1] в главе 2.9 (теорема 2.36).

Θ-алгоритм, как показано в теореме 1, эффективно работает с последовательностями экспоненциального и рационального типов, включая их смешанные формы. Численные эксперименты подтверждают его способность ускорять сходимость даже для сложных случаев, таких как логарифмические последовательности. Кроме того, алгоритм демонстрирует устойчивость к колебаниям членов последовательности благодаря введению параметра ω_k, который компенсирует главный член погрешности.

Реализация алгоритма

```
Вход:
   n: int          - количество членов ряда для частичной суммы
   order: int      - порядок преобразования (должен быть четным)
   series: callable - функция, возвращающая a_k (k-й член ряда)

Выход:
   T               - ускоренное значение частичной суммы

Получить n, order, series

if order % 2 != 0 или order < 0:
   return "Ошибка: Порядок должен быть четным неотрицательным числом"

if n < 0:
   return "Ошибка: n не может быть отрицательным"

if n == 0 или order == 0:
   return sum_{k=0}^n series(k)  # Частичная сумма S_n

Инициализировать:
   S_n = sum_{k=0}^n series(k)
   j = 0

Вычислить Theta(n, order, S_n, j, series):

Функция Theta(n, order, S_n, j, series):
   if order == 1:
      res = 1 / series(n + j + 1)
      if not isfinite(res):
         return "Ошибка: Деление на ноль"
      return res

   # Обновление частичной суммы
   for k от n+1 до n+j:
      S_n += series(k)
   n = n + j

   if order == 0:
      return S_n

   order1 = order - 1
   order2 = order - 2
```

```
# Рекурсивные вызовы
t1_0 = Theta(n, order1, S_n, 0, series)
t1_1 = Theta(n, order1, S_n, 1, series)
t1_2 = Theta(n, order1, S_n, 2, series)
t2_1 = Theta(n, order2, S_n, 1, series)

if order % 2 != 0:  # Нечетный порядок
    delta = 1 / (t1_0 - t1_1)
    if not isfinite(delta):
        return "Ошибка: Деление на ноль"
    return t2_1 + delta

else:  # Четный порядок
    delta2 = 1 / (-2*t1_1 + t1_0 + t1_2)
    if not isfinite(delta2):
        return "Ошибка: Деление на ноль"
    delta_n = t2_1 - Theta(n, order2, S_n, 2, series)
    delta_n1 = t1_1 - t1_2
    return t2_1 + (delta_n * delta_n1 * delta2)
```

Рисунок 2 – Псевдокод алгоритма Тета-Брезински

Список литературы

1. Brezinski C. / *Extrapolation Methods: Theory and Practice* / C. Brezinski, M. Redivo Zaglia. — Amsterdam : North-Holland, 1991. — 353 p.