Постановка математической задачи для Θ – алгоритма

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

$$S_n = \sum_{k=0}^{n} a_k \qquad (1)$$

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

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

Θ – алгоритм

Классический ε-алгоритм может быть представлен в виде:

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

Более подробное разложение можно найти в книге Клода Брецински [1]. 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:

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

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:

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

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

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 Брецински [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:

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

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:

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

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

Обратите внимание, $D_{2k+1}^{(n)}$ описан в формуле (2) без учета замены ε на Θ. Весь алгоритм был предложен Клодом Брецински и дополнительное его описание можно найти в ранее указанной книге [1].

Численные эксперименты 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 (Sn) имеет одну из следующих форм:

1. Экспоненциальная:

$$S_n = S + (S_0 - S)\lambda^n, \lambda \neq 0{,}1 \tag{13}$$

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

2. Рациональная:

$$S_n = S + (S_0 - S)\prod_{i=0}^{n-1}\left[1 - \frac{d}{i-m}\right], \text{где } S_0 \neq S, d \neq 1, m, m + d \notin \mathbb{Z} \tag{14}$$

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

3. Специальные вырожденные случаи при

$$S_0 = S, S_n = S + (S_1 - S)\prod_{i=0}^{n-1}\left(1 - \frac{d}{i}\right) \text{ для } n \geq 1, \text{где } S_1 \neq S, d \notin \mathbb{Z} \tag{15}$$

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

Более подробное доказательство теоремы можно найти в книге Брецинского [1] в главе 2.9 (теорема 2.36).

Таким образом Θ – алгоритм демонстрирует устойчивость для широкого класса последовательностей, способен ускорять сходимости даже в логарифмических случаях и обдает хорошей устойчивостью к колебаниям членов последовательности.

Реализация алгоритма

```
Функция ThetaBrezinski(n, порядок):
    Если порядок нечетный или порядок < 0:
        Ошибка "Порядок должен быть четным числом"

    Если n < 0:
        Ошибка "n не может быть отрицательным"

    Если n == 0 или порядок == 0:
        Вернуть ЧастичнаяСумма(n)

    Вернуть Theta(n, порядок, ЧастичнаяСумма(n), 0)

Функция Theta(n, порядок, s_n, j):
    Если порядок == 1:
        res = 1 / a_{n+j+1}  # a_{n} - n-й член ряда
        Если res не конечно:
            Ошибка "Деление на ноль"
        Вернуть res

    # Обновляем частичную сумму
    Для tmp от n+1 до n+j:
        s_n += a_{tmp}
    n = n + j

    Если порядок == 0:
        Вернуть s_n

    порядок1 = порядок - 1
    порядок2 = порядок - 2

    # Рекурсивные вызовы
    t1_0 = Theta(n, порядок1, s_n, 0)
    t1_1 = Theta(n, порядок1, s_n, 1)
    t1_2 = Theta(n, порядок1, s_n, 2)
    t2_1 = Theta(n, порядок2, s_n, 1)

    Если порядок нечетный:
        delta = 1 / (t1_0 - t1_1)
        Если delta не конечно:
            Ошибка "Деление на ноль"
        Вернуть t2_1 + delta
    Иначе:  # порядок четный
```

```
        delta2 = 1 / (-2*t1_1 + t1_0 + t1_2)
        Если delta2 не конечно:
            Ошибка "Деление на ноль"

        delta_n = t2_1 - Theta(n, порядок2, s_n, 2)
        delta_n1 = t1_1 - t1_2

    Вернуть t2_1 + (delta_n * delta_n1 * delta2)
```
Рисунок 1 – 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.