Θ – алгоритм

Классический ε-алгоритм может быть представлен в виде:

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

Более подробное разложение можно найти в книге Клода Брецински [1]. Нам важно то, что данная формула подчеркивает двух шаговую природу алгоритма, где каждый новый элемент зависит от элементов на двух предыдущих уровнях.

Применяя оператор конечной разности Δ мы можем получить соотношение:

$$\Delta\varepsilon_{k+1}^{(n)} = \Delta\varepsilon_{k-1}^{(n+1)} + \Delta D_k^{(n)} \quad (2)$$

Далее рассмотрим условие ускорения сходимости:

$$\lim_{n\to\infty} \frac{\Delta\varepsilon_{2k+2}^{(n)}}{\varepsilon_{2k}^{(n+1)}} = 0 \quad (3)$$

Для его выполнения необходимо и достаточно, чтобы:

$$\lim_{n\to\infty} \frac{\Delta D_{2k+1}^{(n)}}{\Delta\varepsilon_{2k}^{(n+1)}} = -1 \quad (4)$$

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

В случаях, когда условие (3) не выполняется, вводится дополнительный параметр ω_k:

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \omega_k D_{2k+1}^{(n)} \quad (5)$$

Оптимальным образом определить значение ω_k можно так:

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

При таком выборе параметра ω_k последовательность (5) будет сходиться быстрее, чем $\Delta\varepsilon_{2k}^{(n)}$

На практике довольно часто вычисление предела затруднительно, поэтому можно использовать оценку:

$$\omega_k = -\frac{\Delta\varepsilon_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} \quad (7)$$

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

Инициализация:

$$\Theta_{-1}^{(n)} = 0, \Theta_0^{(n)} = S_n \quad (8)$$

Рекуррентные правила:

$$\Theta_{2k+1}^{(n)} = \Theta_{2k-1}^{(n+1)} + D_{2k}^{(n)} \quad (9)$$

$$\Theta_{2k+2}^{(n)} = \Theta_{2k}^{(n+1)} - \frac{\Delta\Theta_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}} D_{2k+1}^{(n)} \quad (10)$$

Обратите внимание, $D_{2k+1}^{(n)}$ описан в формуле (1) 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 (S_n) имеет одну из следующих форм:

1. Экспоненциальная: $S_n = S + (S_0 - S)\lambda^n, \lambda \neq 0{,}1$
2. Рациональная: $S_n = S + (S_0 - S)\prod_{i=0}^{n-1}\left[1 - \frac{d}{i-m}\right]$, где $S_0 \neq S, d \neq 1$, $m, m + d \notin \mathbb{Z}$
3. Специальные вырожденные случаи при $S_0 = S$.

Заметим, что в первом случае (S_n) сходится тогда и только тогда, $|\lambda| < 1$, тогда как в двух других случаях последовательность сходится тогда и только тогда, когда вещественная часть d строго положительная. В случае сходимости (S_n) стремится к S при n, стремящимся к бесконечности. 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.

Список литературы

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