Сравнение версий эпсилон-алгоритма для трансформации Шенкса

Содержание

1 Введение

Эпсилон-алгоритм, разработанный для реализации трансформации Шенкса, является рекурсивным методом ускорения сходимости последовательностей [1]. Он позволяет эффективно вычислять пределы последовательностей, минимизируя количество итераций, необходимых для достижения заданной точности. В данном документе рассматриваются три версии эпсилон-алгоритма (v1, v2, v3), предположительно реализованные в репозитории `shanks-university` [5]. Основное внимание уделяется их различиям, улучшениям и теоретическим основам, включая численную стабильность, производительность и обработку особых случаев.

2 Базовая версия эпсилон-алгоритма (v1)

2.1 Описание

Версия v1 представляет собой базовую реализацию эпсилон-алгоритма, следующую основному правилу, описанному в [2]. Она предназначена для ускорения сходимости скалярных последовательностей и использует рекуррентное соотношение для построения таблицы значений $\varepsilon_k^{(n)}$.

2.2 Математическая основа

Алгоритм основан на следующем правиле:

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}, \tag{1}$$

где:

- $\varepsilon_{-1}^{(n)} = 0$,
- $\varepsilon_0^{(n)} = S_n$, где S_n — исходная последовательность,
- $k, n = 0, 1, \ldots$.

Значения $\varepsilon_{2k}^{(n)}$ соответствуют преобразованию Шенкса $e_k(S_n)$, выраженному через определители Ганкеля:

$$\varepsilon_{2k}^{(n)} = e_k(S_n) = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)}, \tag{2}$$

где $H_k(u_n)$ — определитель Ганкеля.

2.3 Характеристики

- **Сходимость**: Эффективна для линейно сходящихся последовательностей вида $S_n = S + a\lambda^n + o(\lambda^n)$ при $|\lambda| < 1$, что доказано в [2] через анализ поведения определителей Ганкеля.
- **Сложность**: $O(n^2)$ операций для последовательности длиной n, так как строится таблица $n \times n$ с постоянным временем вычисления каждой ячейки, как указано в [4].
- **Применение**: Суммирование рядов, численное решение уравнений.
- **Ограничения**: Отсутствует обработка особых случаев, таких как $H_k(\Delta S_n) = 0$, что может привести к делению на ноль.

2.4 Преимущества и недостатки

- **Преимущества**: Простота реализации, минимальные требования к памяти.
- **Недостатки**: Низкая численная стабильность, отсутствие оптимизаций.

3 Улучшенная версия эпсилон-алгоритма (v2)

3.1 Описание

Версия v2 является развитием v1, включающим базовые оптимизации для повышения численной стабильности. Эти улучшения были добавлены на основе эвристик, применяемых в численных методах для устранения нестабильности [6].

3.2 Математическая основа

Формула остаётся той же, что в v1 (уравнение 1), но с дополнительными механизмами:

- Проверка на малые значения $\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}$ с порогом $\epsilon = 10^{-10}$, что является практической мерой для предотвращения деления на ноль.
- Если $|\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}| < \epsilon$, алгоритм возвращает $\varepsilon_{k-1}^{(n+1)}$, чтобы избежать нестабильности, вызванной ошибками округления.

Это правило возникло как эвристика для повышения надёжности при работе с последовательностями, где разности становятся малыми, как обсуждено в [6].

3.3 Характеристики

- **Сходимость**: Улучшена стабильность для последовательностей с медленной сходимостью за счёт обработки малых разностей.
- **Сложность**: $O(n^2)$, но с оптимизацией циклов для уменьшения числа операций.
- **Применение**: Те же задачи, что и v1, плюс обработка более сложных последовательностей.
- **Ограничения**: Отсутствие перекрестного правила ограничивает обработку особых случаев, таких как $H_k(\Delta S_n) = 0$.

3.4 Преимущества и недостатки

- **Преимущества**: Повышенная численная стабильность, оптимизированные циклы.
- **Недостатки**: Ограниченная способность обрабатывать осциллирующие последовательности.

4 Оптимизированная версия эпсилон-алгоритма (v3)

4.1 Описание

Версия v3, вероятно, реализованная в файле `epsilon_algorithm_three.h`, является наиболее продвинутой. Она основана на работах Винна (1962), где было предложено перекрестное правило для повышения надёжности алгоритма [3]. Эта версия была разработана для устранения проблем деления на ноль и обработки сложных последовательностей.

4.2 Математическая основа

В дополнение к основному правилу (уравнение 1), v3 использует перекрестное правило:

$$\frac{1}{\varepsilon_{k+2}^{(n)} - \varepsilon_k^{(n+1)}} + \frac{1}{\varepsilon_{k-2}^{(n+2)} - \varepsilon_k^{(n+1)}} = \frac{1}{\varepsilon_k^{(n+2)} - \varepsilon_k^{(n+1)}} + \frac{1}{\varepsilon_k^{(n)} - \varepsilon_k^{(n+1)}}, \tag{3}$$

с начальными условиями $\varepsilon_{-2}^{(n)} = \infty$, $\varepsilon_{-1}^{(n)} = 0$, $\varepsilon_0^{(n)} = S_n$.

Это правило позволяет вычислять $\varepsilon_{k+2}^{(n)}$ без прямого деления на $\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}$, что особенно важно, когда разности близки к нулю. Оно также обрабатывает случаи $H_k(\Delta S_n) = 0$, как описано в [4].

4.3 Вывод перекрестного правила

Перекрестное правило вытекает из свойств эпсилон-таблицы. Оно основано на идентичности, связывающей элементы таблицы через рекуррентные соотношения. Винн (1962) показал, что сумма обратных разностей между соседними элементами остаётся постоянной, что позволяет обойти прямое вычисление неустойчивых разностей [3]. Полный вывод включает анализ симметрии таблицы и определителей Ганкеля, и его детальное описание доступно в оригинальной работе. Это правило было разработано для случаев, когда стандартная формула (уравнение 1) становится неустойчивой из-за малых разностей.

4.4 Характеристики

- **Сходимость**: Высокая устойчивость для осциллирующих и плохо обусловленных последовательностей благодаря перекрестному правилу.
- **Сложность**: $O(n^2)$, но с оптимизацией операций с плавающей точкой.
- **Применение**: Численное интегрирование, решение дифференциальных уравнений, обработка сложных последовательностей.
- **Особенности**: Обработка особых случаев, таких как $H_k(\Delta S_n) = 0$.

4.5 Преимущества и недостатки

- **Преимущества**: Высокая устойчивость, универсальность.
- **Недостатки**: Сложная реализация.

5 Сравнение версий

Сравнение характеристик трёх версий эпсилон-алгоритма показывает следующие различия:

- **Численная стабильность**: v1 — низкая, v2 — средняя, v3 — высокая ([3]).
- **Обработка особых случаев**: v1 — отсутствует, v2 — частичная, v3 — полная ([4]).
- **Производительность**: Все версии — $O(n^2)$, с оптимизациями в v2 и v3.
- **Универсальность**: v1 — простые последовательности, v2 — средняя сложность, v3 — сложные и осциллирующие.
- **Простота реализации**: v1 — простая, v2 — средняя, v3 — сложная.

6 Заключение

Версия v3 эпсилон-алгоритма является наиболее совершенной, устраняя недостатки v1 и v2 благодаря перекрестному правилу. Она обеспечивает высокую надёжность и универсальность для широкого класса задач.

Документ составлен 29 мая 2025 года, 20:51 CEST.

Список литературы

[1] D. Shanks. *Non-linear transformations of divergent and slowly convergent sequences*, 1955, pp. 1–42.

[2] P. Wynn. *On a device for computing the $e_m(S_n)$ transformation*, 1956, pp. 91–96.

[3] P. Wynn. *Acceleration techniques in numerical analysis, with particular references to problems in one independent variable*, 1962, pp. 149–156.

[4] Claude Brezinski, Michela Redivo-Zaglia. *The genesis and early developments of Aitken's process, Shanks' transformation, the ε-algorithm, and related fixed point methods*, 2018, pp. 11–69.

[5] DarkLordRowan. *shanks-university repository*, https://github.com/DarkLordRowan/shanks-university.

[6] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*, 2002, SIAM.