Сравнение версий эпсилон-алгоритма для трансформации Шенкса

Содержание

1 Введение

2 Базовая версия эпсилон-алгоритма (v1)

2.1 Описание

2.2 Математическая основа

2.3 Характеристики

2.4 Преимущества и недостатки

3 Улучшенная версия эпсилон-алгоритма (v2)

3.1 Описание

3.2 Математическая основа

3.3 Характеристики

3.4 Преимущества и недостатки

4 Оптимизированная версия эпсилон-алгоритма (v3)

4.1 Описание

4.2 Математическая основа

4.3 Вывод перекрестного правила

4.4 Характеристики

4.5 Преимущества и недостатки

5 Сравнение версий

6 Заключение

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 εk(n).

2.2 Математическая основа

Алгоритм основан на следующем правиле:

ε(k+1)^(n) = ε(k-1)^(n+1) + 1 / (εk(n+1) - εk(n)),

где:

- ε(-1)^(n) = 0,

- ε0(n) = S\_n, где S\_n — исходная последовательность,

- k, n = 0, 1, ...

Значения ε(2k)^(n) соответствуют преобразованию Шенкса e\_k(S\_n), выраженному через определители Ганкеля:

ε(2k)^(n) = e\_k(S\_n) = H\_(k+1)(S\_n) / H\_k(Δ²S\_n),

где H\_k(u\_n) — определитель Ганкеля.

2.3 Характеристики

- Сходимость: Эффективна для линейно сходящихся последовательностей вида S\_n = S + aλ^n + o(λ^n) при |λ| < 1, что доказано в [2].

- Сложность: O(n²) операций для последовательности длиной n, поскольку строится таблица n×n.

- Применение: Суммирование рядов, численное решение уравнений.

- Ограничения: Нет обработки особых случаев, возможны деления на ноль при H\_k(ΔS\_n) = 0.

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 представляет собой развитие v1, включающее эвристики для повышения численной устойчивости. Оптимизации включают обработку малых разностей.

3.2 Математическая основа

Формула та же, что и в v1, но с добавлением:

- Проверки |εk(n+1) - εk(n)| < ε (порог 10^(-10)),

- В этом случае возвращается ε(k-1)^(n+1).

3.3 Характеристики

- Сходимость: Лучше при медленной сходимости.

- Сложность: O(n²), но уменьшено число операций.

- Применение: Более сложные последовательности.

- Ограничения: Нет перекрестного правила.

3.4 Преимущества и недостатки

- Преимущества: Повышенная стабильность, более эффективные циклы.

- Недостатки: Сложности с осциллирующими последовательностями.

4 Оптимизированная версия эпсилон-алгоритма (v3)

4.1 Описание

Версия v3, вероятно реализованная в файле epsilon\_algorithm\_three.h, использует перекрестное правило, предложенное Винном в [3].

4.2 Математическая основа

Перекрестное правило:

1 / (ε(k+2)^(n) - εk(n+1)) + 1 / (ε(k-2)^(n+2) - εk(n+1)) =

1 / (εk(n+2) - εk(n+1)) + 1 / (εk(n) - εk(n+1))

Начальные условия: ε(-2)^(n) = ∞, ε(-1)^(n) = 0, ε0(n) = S\_n.

4.3 Вывод перекрестного правила

Правило вытекает из симметрии таблицы и свойств определителей Ганкеля. Позволяет избежать деления на малые разности.

4.4 Характеристики

- Сходимость: Высокая устойчивость к осцилляциям.

- Сложность: O(n²) с оптимизациями.

- Применение: Интегрирование, ОДУ, сложные последовательности.

- Особенности: Обработка H\_k(ΔS\_n) = 0.

4.5 Преимущества и недостатки

- Преимущества: Универсальность, надёжность.

- Недостатки: Сложность реализации.

5 Сравнение версий

- Стабильность: v1 — низкая, v2 — средняя, v3 — высокая.

- Обработка особых случаев: v1 — нет, v2 — частично, v3 — полностью.

- Сложность: Все версии O(n²), но v2 и v3 оптимизированы.

- Универсальность: v1 — простые случаи, v2 — средние, v3 — любые.

- Реализация: v1 — простая, v2 — умеренная, v3 — сложная.

6 Заключение

Версия v3 наиболее совершенная, она устойчиво работает даже в сложных случаях благодаря перекрестному правилу.

Список литературы:

[1] D. Shanks, 1955

[2] P. Wynn, 1956

[3] P. Wynn, 1962

[4] Brezinski & Redivo-Zaglia, 2018

[5] GitHub, shanks-university

[6] Higham, 2002