Оглавление

Введение

Метод Шенкса был разработан в 1955 году американским математиком Даниэлем Шенксом и с тех пор является важным инструментом в численном анализе и методах решения уравнений [5]. Он предназначен для ускорения сходимости последовательностей и суммируемых рядов. Скорость и порядок сходимости – это величины, которые представляют, насколько быстро последовательность приближается к своему пределу. Последовательность x_n, сходящаяся к x^*, имеет порядок сходимости $q \geq 1$ и скорость сходимости μ, если $\lim_{n\to\infty} \frac{|x_{n+1}-x^*|}{|x_n-x^*|^q} = \mu$. Во многих задачах, где требуется численное приближение, бывает необходимо вычислить сумму бесконечного ряда или оценить предел последовательности, однако, некоторые ряды сходятся медленно и требуют большого количества итераций для достижения точности.

Ускорение достигается путем изменения ряда с сохранением суммы. Это позволяет достичь лучшей сходимости и значительно уменьшить количество итераций.

1. Базовый метод трансформации Шенкса

Сущность метода Шенкса состоит в том, что по заданной последовательности строится новая. Нам необходимо определить S_n – частичную сумму бесконечной последовательности (1), S – предел последовательности (2):

$$S_n = \sum_{i=0}^{n} a_i, \tag{1}$$

$$S = \sum_{i=0}^{\infty} a_i \; . \tag{2}$$

Трансформация нацелена на ускорение сходимости S. Для этого запишем предел в виде (3), теперь у нас три неизвестные: S, ɣ, ν, где |ɣ|<1, а
ν - параметр [3].

$$S \equiv S_n - \nu ɣ^n \quad . \tag{3}$$

Их можно рассчитать на основе частичных сумм (4):

$$\begin{cases} S_{n-1} = S + \nu ɣ^{n-1} \\ S_n = S + \nu ɣ^n \\ S_{n+1} = S + \nu ɣ^{n+1} \end{cases} . \tag{4}$$

Предел S представлен в виде преобразования $T(S_n)$ частичной суммы порядка n:

$$S = T(S_n) = \frac{S_{n+1}S_{n-1} - S_n^2}{S_{n+1} + S_{n-1} - 2S_n}. \tag{5}$$

Важным свойством этого преобразования является то, что при повторном применении получается новая последовательность $\{T(S_1), T(S_1), \ldots, T(S_{n-1})\}$, имеющую на 2 члена меньше. Эту процедуру продолжаем пока в последовательности не останется 1 или 2 члена. (5) можно преобразовать в равенство (6), что является нашим первым шагом в процедуре трансформации:

$$T(S_n) = S_n + \frac{a_n a_{n+1}}{a_n - a_{n+1}}\,. \tag{6}$$

2. Многошаговый алгоритм Шенкса и $\boldsymbol{\varepsilon}$ – алгоритм

Далее S_n – последовательность, S – предел, T_n – новая последовательность. Преобразование последовательности называется методом ускорения сходимости или методом экстраполяции.

Пусть $S_n -$ последовательность, сходящаяся к пределу S, удовлетворяющему

$$\lim_{n\to\infty} \frac{S_{n+1}-S}{S_n-S} = \lambda.$$

Когда $0 < \lambda < 1$, последовательность S_n сходится линейно[1], когда $\lambda = 1$, последовательность сходится логарифмически[2], а когда $\lambda = 0$, у нее сверхлинейная[3] сходимость. Преобразование последовательности $T : (S_n) \to (T_n)$ преобразует эту последовательность в новую последовательность (T_n), которая быстрее сходится к тому же пределу S при некоторых предположениях, то есть

$$\lim_{n\to\infty} \frac{T_n-S}{S_n-S} = 0.$$

Идея, лежащая в основе таких преобразований, состоит в предположении, что преобразуемая последовательность S_n ведет себя как модельная последовательность[4]. Множество K_t этих модельных последовательностей называется ядром преобразования. Оно характеризуется тем, что каждая из его

---

[1] Последовательность x_k, сходящаяся к x^*, обладает линейной сходимостью если существует такая постоянная С $\in [0,1]$ и такой номер k_0, что $|x^* - x_{k+1}| \le C|x^* - x_k|$ для всех $k \ge k_0$.

[2] Положительная последовательность x_k обладает логарифмической сходимостью если существует такое α и такой номер k_0, что $\frac{ln\frac{1}{x_k}}{ln\,k} \ge 1 + \alpha$ для всех $k \ge k_0$.

[3] Последовательность x_k, сходящаяся к x^*, обладает сверхлинейной сходимостью если существует такая положительная последовательность C_k и такой номер k_0, что $\lim_{k\to\infty} C_k = 0$ и $|x^* - x_{k+1}| \le C_k|x^* - x_k|$ для всех $k \ge k_0$.

[4] Модельная последовательность – это последовательность, предел которой может быть точно вычислен с помощью алгебраического процесса.

последовательностей U_n (являющаяся подпоследовательностью S_n) удовлетворяет заданному алгебраическому соотношению в зависимости от своего неизвестного предела U (или антипредела[5], если он не сходится), и от p неизвестных параметров $a = (a_1, \ldots, a_p)^T$ и имеет вид:

$$\forall n, R\left(u_n, \ldots, u_{n+q}, u, a\right) = 0. \quad (7)$$

Это отношение называется неявной формой ядра. R представляет собой действительную матрицу вращения $p \times p$ элементов. Она вращает компоненты вектора в двумерном подпространстве, но не влияет на компоненты вектора в оставшемся $(p-2)$ - мерном подпространстве. Рассматривая определенное количество последовательных членов подлежащей преобразованию S_n, начиная с индекса n, то есть $S_n, \ldots, S_{n+p+q}$, мы ищем последовательность $(U_n) \in K_t$, удовлетворяющую следующим условиям:

$$u_i = S_i \text{ для } i = n, \ldots, n+p, \quad (8)$$

что есть

$$R\left(S_i, \ldots, S_{i+q}, u, a\right) = 0 \text{ для } i = n, \ldots, n+p. \quad (9)$$

Решая эту систему, мы получим $p + 1$ неизвестных a и u. Неизвестное значение u является пределом последовательности $(U_n) \in K_t$ и было получено путем экстраполяции. Поскольку оно зависит от n, то обозначим его через T_n. Следовательно, последовательность S_n была преобразована в новую последовательность T_n. Если последовательность (S_n) принадлежит ядру K_t,

[5] Антипредел – это математический термин, обозначающий обратный процесс к операции нахождения предела; является эквивалентом предела для расходящихся рядов. Вычисляется с помощью формулы для предела параметризованного ряда, применяемой к любым значениям параметров, в которых ряд не сходится.

то $\forall n\, T_n = S$, её точный предел S (или антипредел). Решение (7) дает замкнутую форму для U_n, которая называется явной формой ядра.

Способ 1. Рассматриваем скалярную последовательность, удовлетворяющую соотношению:

$$a_0(S_n - S) + a_1(S_{n+1} - S) + \cdots + a_k(S_{n+k} - S) = 0, n = 0,1, \ldots, \qquad (10)$$

где $a_0, \ldots, a_k$ – неизвестные константы, сумма которых не равна 0 и $a_0 \cdot a_k \neq 0$ (иначе k должно быть заменено на меньшее значение, так как порядок этого разностного уравнения уменьшается).

Последовательность (10) является обобщением последовательности S_n вида

$$S_{n+1} - S = \lambda(S_n - S), n = 0,1, \ldots, \qquad (10.1)$$

где известна λ.

$a_0 + \cdots + a_k = 1$ – условие нормализации. Набор последовательностей, удовлетворяющий этому разностному уравнению, называется ядром Шенкса. Опять же возникает проблема в вычислении неизвестного S. Когда $k = 1$, из (10) получается (10.1)

Поскольку (10) содержит $k+1$ неизвестный элемент, и мы не знаем S, напишем еще соотношение для индексов $n,\ldots,n+k$, которое представляет собой однородную линейную систему с ненулевым решением:

$$a_0(S_{n+i} - S) + a_1(S_{n+i+1} - S) + \cdots + a_k(S_{n+k+i} - S) = 0, i = 0, \ldots, k. \qquad (11)$$

Таким образом, ее определитель должен быть равен нулю, иначе система будет иметь нулевое решение:

$$\begin{vmatrix} S_n - S & S_{n+1} - S & \dots & S_{n+k} - S \\ S_{n+1} - S & S_{n+2} - S & \dots & S_{n+k+1} - S \\ \vdots & \vdots & & \vdots \\ S_{n+k} - S & S_{n+k+1} - S & \dots & S_{n+2k} - S \end{vmatrix} = 0. \tag{12}$$

Вычтем из каждой строки предыдущую и запишем определитель, как разность двух определителей:

$$\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \dots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \dots & \Delta S_{n+2k-1} \end{vmatrix} - S \begin{vmatrix} 1 & 1 & \dots & 1 \\ \Delta S_n & \Delta S_{n+1} & \dots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \dots & \Delta S_{n+2k-1} \end{vmatrix} = 0, \tag{13}$$

где $\Delta S_n = \alpha(\lambda - 1)\lambda^n$, $\lambda = \lim\limits_{n\to\infty} \frac{S_{n+1}-S}{S_n-S}$, $\alpha = S_0 - S$.

Если S_n не удовлетворяет (10), то определители всё ещё могут быть вычислены, но их отношение больше не будет равно S для всех n, а равно числу, зависящему от k и n, и обозначается $e_k(S_n)$. Данный случай рассмотрен в теореме 1.

Таким образом, последовательность S_n была преобразована в набор последовательностей $\{(e_k(S_n))\}$. Это и есть определение трансформации Шенкса.

Для получения формулы трансформации в числителе происходит замена каждой строки на её сумму с предыдущей, а в знаменателе каждого столбца – на его разность с предыдущим, что приводит к следующему выражению:

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \dots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \dots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \dots & 1 \\ \Delta S_n & \Delta S_{n+1} & \dots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \dots & \Delta S_{n+2k-1} \end{vmatrix}} = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)}, k, n = 0,1, \dots, \tag{14}$$

где $H_k(u_n)$ обозначает определитель Ганкеля:

$$H_k(u_n) = \begin{vmatrix} u_n & u_{n+1} & \cdots & u_{n+k-1} \\ u_{n+1} & u_{n+2} & \cdots & u_{n+k} \\ \vdots & \vdots & & \vdots \\ u_{n+k-1} & u_{n+k} & \cdots & u_{n+2k-2} \end{vmatrix}, \tag{15}$$

причем $H_0(u_n) = 1$. Из-за своей особой формы такой определитель также называется симметричным в том смысле, что все его элементы на любой диагонали под прямым углом к главной диагонали одинаковы.

Выражения (14) и (15) могут быть рекурсивно вычислены с помощью тождества Сильвестра, которое в данном случае дает:

$$H_{k+2}(u_n)H_k(u_{n+2}) = H_{k+1}(u_n)H_{k+1}(u_{n+2}) - [H_{k+1}(u_{n+1})]^2, \tag{16}$$

учитывая $H_0(u_n) = 1$ и $H_1(u_n) = u_n$.

Распишем (16) более подробно:

Тождество Сильвестра утверждает, что для произвольной матрицы A выполнено:

$$|A||A_{r\cap s,p\cap q}| = |A_{r,p}| \cdot |A_{s,q}| - |A_{r,q}| \cdot |A_{s,p}|,$$

где $A_{u,w}$ это подматрица A, образованная пересечением подмножества строк u и столбцов w исходной матрицы, а $|A|$ − определитель матрицы A.

Тогда в формуле (16) подразумевается, что:

$$|A| = H_{k+2}(u_n) = \begin{vmatrix} u_n & u_{n+1} & \cdots & u_{n+k+1} \\ u_{n+1} & u_{n+2} & \cdots & u_{n+k+2} \\ \vdots & \vdots & & \vdots \\ u_{n+k+1} & u_{n+k+2} & \cdots & u_{n+2k+2} \end{vmatrix},$$

$$|A_{r\cap s,p\cap q}| = H_k(u_{n+2}) = \begin{vmatrix} u_{n+2} & u_{n+3} & \cdots & u_{n+k+1} \\ u_{n+3} & u_{n+4} & \cdots & u_{n+k+2} \\ \vdots & \vdots & & \vdots \\ u_{n+k+1} & u_{n+k+2} & \cdots & u_{n+2k} \end{vmatrix},$$

$$|A_{r,p}| = H_{k+1}(u_n) = \begin{vmatrix} u_n & u_{n+1} & \cdots & u_{n+k} \\ u_{n+1} & u_{n+2} & \cdots & u_{n+k+1} \\ \vdots & \vdots & & \vdots \\ u_{n+k} & u_{n+k+1} & \cdots & u_{n+2k} \end{vmatrix},$$

$$|A_{s,q}| = H_{k+1}(u_{n+2}) = \begin{vmatrix} u_{n+2} & u_{n+3} & \cdots & u_{n+k+2} \\ u_{n+3} & u_{n+4} & \cdots & u_{n+k+3} \\ \vdots & \vdots & & \vdots \\ u_{n+k+2} & u_{n+k+3} & \cdots & u_{n+2k+2} \end{vmatrix},$$

$$|A_{r,q}| = |A_{s,p}| = H_{k+1}(u_{n+1}) = \begin{vmatrix} u_{n+1} & u_{n+2} & \cdots & u_{n+k+1} \\ u_{n+2} & u_{n+3} & \cdots & u_{n+k+2} \\ \vdots & \vdots & & \vdots \\ u_{n+k+1} & u_{n+k+2} & \cdots & u_{n+2k+1} \end{vmatrix},$$

где $r -$ подмножество первых $k+1$ строк, $s -$ подмножество всех строк, кроме первой, $p -$ подмножество первых $k+1$ столбцов, $q -$ подмножество всех столбцов, кроме первого.

Поскольку $e_k(S_n) = H_{k+1}(S_n)/H_k(\Delta^2 S_n)$, данное отношение можно вычислить, применяя отдельно предыдущее рекуррентное соотношение к его числителям и знаменателям. Шенкс действовал таким образом для рекурсивной реализации своего преобразования.

Формула (14) показывает, что $e_k(S_n)$ представляет собой линейную комбинацию $S_n, \dots, S_{n+k}$, коэффициенты которой обозначены через $a_i^{(n)}$, зависят от n (также и от k).

Заменяя каждую строку на её разность с предыдущей и повторяя эту операцию несколько раз и выполняя её также над столбцами, получим:

$$H_k(u_n) = \begin{vmatrix} u_n & \cdots & u_{n+k-1} \\ \Delta u_n & \cdots & \Delta u_{n+k-1} \\ \vdots & & \vdots \\ \Delta^{k-1} u_n & \cdots & \Delta^{k-1} u_{n+k-1} \end{vmatrix} = \begin{vmatrix} u_n & \Delta u_n & \cdots & \Delta^{k-1} u_n \\ \Delta u_n & \Delta^2 u_n & \cdots & \Delta^k u_n \\ \vdots & & & \vdots \\ \Delta^{k-1} u_n & \Delta^k u_n & \cdots & \Delta^{2k-2} u_n \end{vmatrix}.$$

Заменяя в числителе и знаменателе (14) каждый столбец, начиная со второго, на его разность с предыдущим, получим:

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & S_n & \dots & S_{n+k-1} \\ \Delta S_n & \Delta^2 S_n & \dots & \Delta^2 S_{n+k-1} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta^2 S_{n+k-1} & \dots & \Delta^2 S_{n+2k-2} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_n & \dots & \Delta^2 S_{n+k-1} \\ \vdots & & \vdots \\ \Delta^2 S_{n+k-1} & \dots & \Delta^2 S_{n+2k-2} \end{vmatrix}}, \text{где } \Delta^2 S_n = \Delta S_{n+1} - \Delta S_n.$$

Эта формула показывает, что благодаря детерминантной формуле Шура [10], $e_k(S_n)$ является дополнением Шура, то есть

$$e_k(S_n) = S_n - (\Delta S_n, \dots, \Delta S_{n+k-1}) \cdot \begin{pmatrix} \Delta^2 S_n & \dots & \Delta^2 S_{n+k-1} \\ \vdots & & \vdots \\ \Delta^2 S_{n+k-1} & \dots & \Delta^2 S_{n+2k-2} \end{pmatrix}^{-1} \begin{pmatrix} \Delta S_n \\ \vdots \\ \Delta S_{n+k-1} \end{pmatrix}.$$

Способ 2. Трансформацию Шенкса можно получить другим путем. Из (10) уравнения, при раскрытии скобок и применении условия нормировки, получается:

$$e_k(S_n) = a_0^{(n)} S_n + \dots + a_k^{(n)} S_{n+k}, \tag{17}$$

где $a_i^{(n)}$ удовлетворяет системе линейных уравнений:

$$\begin{array}{cccc} a_0^{(n)} & + \dots + & a_k^{(n)} & = 1 \\ a_0^{(n)} \Delta S_n & + \dots + & a_k^{(n)} \Delta S_{n+k} & = 0 \\ \vdots & & \vdots & \vdots \\ a_0^{(n)} \Delta S_{n+k-1} & + \dots + & a_k^{(n)} \Delta S_{n+2k-1} & = 0 \end{array}. \tag{18}$$

По правилу Крамера:

$$a_i^{(n)} = \frac{\begin{vmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \Delta S_n & \dots & \Delta S_{n+i-1} & 0 & \Delta S_{n+i+1} & \dots & \Delta S_{n+k} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+k+i-2} & 0 & \Delta S_{n+k+i} & \dots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \Delta S_n & \dots & \Delta S_{n+k} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+2k-1} \end{vmatrix}}. \tag{19}$$

Преобразуем (19) для получения формулы преобразования Шенкса (14).

Разложим $\begin{vmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \Delta \mathrm{S_n} & \dots & \Delta \mathrm{S_{n+i-1}} & 0 & \Delta \mathrm{S_{n+i+1}} & \dots & \Delta \mathrm{S_{n+k}} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \Delta \mathrm{S_{n+k-1}} & \dots & \Delta \mathrm{S_{n+k+i-2}} & 0 & \Delta \mathrm{S_{n+k+i}} & \dots & \Delta \mathrm{S_{n+2k-1}} \end{vmatrix}$ по его i-му столбцу $\begin{vmatrix} \Delta S_n & \dots & \Delta S_{n+k} \\ \Delta S_{n+1} & \dots & \Delta S_{n+k+1} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+2k-1} \end{vmatrix}$. Добавим нулевую первую строку, заменяя в i-ом столбце ноль на единицу, $\begin{vmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \Delta S_n & \dots & \Delta S_{n+i-1} & \Delta S_{n+i} & \Delta S_{n+i+1} & \dots & \Delta S_{n+k} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+k+i-2} & \Delta S_{n+k+i-1} & \Delta S_{n+k+i} & \dots & \Delta S_{n+2k-1} \end{vmatrix}$. При таком преобразовании сохраняется размерность определителя и его значение, т.к. при вычислении, путем разложения по первой строке, получаем $\begin{vmatrix} \Delta S_n & \dots & \Delta S_{n+k} \\ \Delta S_{n+1} & \dots & \Delta S_{n+k+1} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+2k-1} \end{vmatrix}$. Далее:

$$a_i^{(n)} = \frac{\begin{vmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \Delta S_n & \dots & \Delta S_{n+i-1} & \Delta S_{n+i} & \Delta S_{n+i+1} & \dots & \Delta S_{n+k} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+k+i-2} & \Delta S_{n+k+i-1} & \Delta S_{n+k+i} & \dots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \Delta S_n & \dots & \Delta S_{n+k} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \dots & \Delta S_{n+2k-1} \end{vmatrix}}. \tag{20}$$

Умножая каждый $a_i^{(n)}$ на S_{n+i}, а затем суммируя их и используя (17), получаем (14).

Способ 3. Этот способ получения преобразования Шенкса основывается на том, что (10) эквивалентно:

$$S_n = S + \alpha_0^{(n)} \Delta S_n + \cdots + \alpha_{k-1}^{(n)} \Delta S_{n+k-1}. \tag{21}$$

Запись этого соотношения для индексов $n, \ldots, n+k$ приводит к системе:

$$\begin{pmatrix} 1 & \Delta S_n & \ldots & \Delta S_{n+k-1} \\ 1 & \Delta S_{n+1} & \ldots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ 1 & \Delta S_{n+k} & \ldots & \Delta S_{n+2k-1} \end{pmatrix} \begin{pmatrix} S \\ \alpha_0^{(n)} \\ \vdots \\ \alpha_{k-1}^{(n)} \end{pmatrix} = \begin{pmatrix} S \\ S_{n+1} \\ \vdots \\ S_{n+k} \end{pmatrix}. \tag{22}$$

Покажем, что решение (22) для неизвестного S дает (14).

По правилу Крамера выразим $S = \dfrac{\begin{vmatrix} S & \ldots & \Delta S_n & \ldots & \Delta S_{n+k-1} \\ S_{n+1} & \ldots & \Delta S_{n+1} & & \Delta S_{n+k} \\ \vdots & & \vdots & & \\ S_{n+k} & \ldots & \Delta S_{n+k} & \ldots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \ldots & \Delta S_n & \ldots & \Delta S_{n+k-1} \\ 1 & \ldots & \Delta S_{n+1} & & \Delta S_{n+k} \\ \vdots & & \vdots & & \\ 1 & \ldots & \Delta S_{n+k} & \ldots & \Delta S_{n+2k-1} \end{vmatrix}}$, при транспонировании матрицы определитель не меняется, поэтому равенство можно переписать: $S = \dfrac{\begin{vmatrix} S_n & S_{n+1} & \ldots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \ldots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \ldots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \ldots & 1 \\ \Delta S_n & \Delta S_{n+1} & \ldots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \ldots & \Delta S_{n+2k-1} \end{vmatrix}}$, что равно (14).

Способ 4. Четвертая возможность получения преобразования Шенкса состоит в рассмотрении следующей линейной системы, где c – любая ненулевая константа:

$$\beta_0^{(n)} S_{n+i} + \cdots + \beta_k^{(n)} S_{n+k+i} = c, i = 0, \ldots, k, \tag{23}$$

где $\beta_i = 1 - \sum_{j=0}^{i} \alpha_j$ и содержит $e_k(S_n) = \frac{c}{\sum_{i=0}^{k} \beta_i^{(n)}}$.

Система (23) получена из (11) уравнения, при раскрытии скобок и применении условия нормировки.

Замечание. Первые два способа получения преобразования Шенкса как отношения детерминантов могут быть обобщены на случай, когда S_n являются элементами общего векторного пространства.

Теорема 1.

Достаточным условием того, что $e_k(S_n) = S$ для всех n, является то, что последовательность S_n удовлетворяет (7) вместе с $a_0 \cdot a_k \neq 0$, где $a_0, \dots, a_k$ – неизвестные константы, сумма которых не равна 0 (иначе k должно быть заменено на меньшее значение, так как порядок этого разностного уравнения уменьшается) и $a_0 + \cdots + a_k \neq 0$. Если $\forall n, H_k(\Delta S_n) \neq 0$, то условие тоже необходимо [1].

Доказательство.

Следуя (14), мы имеем $e_k(S_n) = a_0^{(n)} S_n + \cdots + a_k^{(n)} S_{n+k}$, где $a_i^{(n)}$ является решением системы (18) и задается формулами (19).

Тогда условие $e_k(S_n) = S$ упрощает $a_0^{(n)} S_n + \cdots + a_k^{(n)} S_{n+k} = S$. Используя (20), мы получаем $\forall n,\ H_{k+1}(S_n - S) = 0$, и легко выводим:

$$a_0^{(n)}(S_{n+i} - S) + \cdots + a_k^{(n)}(S_{n+k+i} - S) = 0 \text{ для } i = 0, \dots, k. \qquad (24)$$

Аналогично

$$a_0^{(n+1)}(S_{n+i} - S) + \cdots + a_k^{(n+1)}(S_{n+k+i} - S) = 0 \text{ для } i = 0, \dots, k+1. \qquad (25)$$

Из уравнений, соответствующих $i = 1, \dots, k$ из этих двух групп получается

$$(a_0^{(n+1)} - a_0^{(n)})(S_{n+i} - S) + \dots + (a_k^{(n+1)} - a_k^{(n)})(S_{n+k+i} - S) = 0 \qquad (26)$$

для $i = 0, \dots, k$.

Поскольку коэффициенты $a_i^{(n)}$ и $a_i^{(n+1)}$ суммируются до 1, получается

$$\left(a_0^{(n+1)} - a_0^{(n)}\right) + \dots + \left(a_k^{(n+1)} - a_k^{(n)}\right) = 0. \qquad (27)$$

Таким образом, мы получаем однородную систему k *+1* уравнений в k *+1* неизвестных $\left(a_i^{(n+1)} - a_i^{(n)}\right),\ i = 0, \dots, k$.

$$\begin{vmatrix} 1 & \dots & 1 \\ S_{n+1} - S & \dots & S_{n+k+1} - S \\ \vdots & & \vdots \\ S_{n+k} - S & \dots & S_{n+2k} - S \end{vmatrix} = \begin{vmatrix} 1 & 0 & \dots & 1 \\ S_{n+1} - S & \Delta S_{n+1} & \dots & S_{n+k} \\ \vdots & & & \vdots \\ S_{n+k} - S & \Delta S_{n+k} & \dots & S_{n+2k-1} \end{vmatrix} = \qquad (28)$$

$$\begin{vmatrix} \Delta S_{n+1} & \dots & \Delta S_{n+k} \\ \vdots & & \vdots \\ \Delta S_{n+k} & \dots & \Delta S_{n+2k-1} \end{vmatrix},$$

которое, как и предполагалось, отличалось от нуля для всех n. Следовательно, предыдущая однородная система неособая, и из этого следует, что ее решение равно нулю, что означает $a_i^{(n+1)} = a_i^{(n)}$, что не зависит от *n* при *i = 0, …, k*. Из (14) (или (19) и (20)) мы видим, что числитель коэффициента $a_i^{(0)}$ равен $H_k(\Delta S_{n+1})$. К тому же, $a_i^{(n)} \neq 0$ и $H_k(\Delta S_n) \neq 0$. Условие $\forall n, H_k(\Delta S_n) \neq 0$ больше не требуется для доказательства необходимого условия теоремы 1, если преобразование Шенкса реализовано с помощью ε - алгоритма.

3. ε – алгоритм

Теперь, поскольку детерминанты долго вычислять, необходим другой алгоритм реализации преобразования Шенкса. ε – алгоритм является рекурсивным алгоритмом, разработанным для реализации Шенкса без вычисления определителей Ганкеля, фигурирующих в (14) [1].

Основное правило [11]:

Если

$$\varepsilon_{2k}^{(n)} = e_k(s_n),$$

$$\varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta s_n)},$$

Тогда

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}},\ k, n = 0,1, \dots, \qquad (29)$$

учитывая $\varepsilon_{-1}^{(n)} = 0$ и $\varepsilon_0^{(n)} = S_n$, $n = 0,1, \dots$.

Связь между ε – алгоритмом и преобразованием Шенкса определяется выражением:

$$\varepsilon_{2k}^{(n)} = e_k(S_n) \text{ и } \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta S_n)},\ k, n = 0,1, \dots . \qquad (30)$$

Таким образом, $\varepsilon_{2k+1}^{(n)}$ являются промежуточным результатом:

$$\varepsilon_{2k}^{(n)} = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)} \quad \text{и} \quad \varepsilon_{2k+1}^{(n)} = \frac{H_k(\Delta^3 S_n)}{H_{k+1}(\Delta S_n)} . \qquad (31)$$

Величины $\varepsilon_k^{(n)}$ обычно отображаются в виде двумерного массива (ε -массив), где нижний индекс k остается неизменным в столбце таблицы, а верхний индекс n остается неизменным по нисходящей диагонали:

$$
\begin{array}{llllll}
\varepsilon_{-1}^{(0)} = 0 & & & & & \\
 & \varepsilon_0^{(0)} = S_0 & & & & \\
\varepsilon_{-1}^{(1)} = 0 & & \varepsilon_1^{(0)} & & & \\
 & \varepsilon_0^{(1)} = S_1 & & \varepsilon_2^{(0)} & & \\
\varepsilon_{-1}^{(2)} = 0 & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & \\
 & \varepsilon_0^{(2)} = S_2 & & \varepsilon_2^{(1)} & & \ddots \\
\varepsilon_{-1}^{(3)} = 0 & & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & \\
\vdots & \varepsilon_0^{(3)} = S_3 & & \varepsilon_2^{(2)} & & \ddots \; . \\
\vdots & \vdots & \varepsilon_1^{(3)} & & \varepsilon_3^{(2)} & \\
\vdots & \vdots & \vdots & \varepsilon_2^{(3)} & & \ddots \\
\vdots & \vdots & \vdots & \vdots & \varepsilon_3^{(3)} & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}
\tag{32}
$$

Правило (29) ε – алгоритма связывает величины, расположенные в четырех вершинах ромба, в трёх столбцах и двух нисходящих диагоналях:

$$
\begin{array}{ccc}
 & \varepsilon_k^{(n)} & \\
 & & \\
\varepsilon_{k-1}^{(n+1)} & & \varepsilon_{k+1}^{(n)}. \\
 & & \\
 & \varepsilon_k^{(n+1)} &
\end{array}
\tag{33}
$$

Для эффектной реализации ε – алгоритма, наилучший метод, по мнению Винна [6,7], состоит в сохранении последней возрастающей диагонали ε - массива (в этой диагонали сумма нижнего и верхнего индексов постоянна, например, $\varepsilon_0^{(m)} = S_m,\ \varepsilon_1^{(m-1)}, \dots, \varepsilon_m^{(0)}$ и складывая по одному, то есть $\varepsilon_0^{(m+1)} = S_{m+1}$) члены последовательности, подлежащей преобразованию. Это позволяет уйти от хранения величин в двумерном массиве к хранению в одномерном. Данные, которые теперь не будут запоминаться могут быть восстановлены с помощью соотношений, представленных ранее.

Теорема 2.

$$\varepsilon_{2k}^{(n)} = e_k(S_n) = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)} \quad \text{и} \quad \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta S_n)} = \frac{H_k(\Delta^3 S_n)}{H_{k+1}(\Delta S_n)}, \forall k, n. \tag{34}$$

Этот результат показывает, что значения $\varepsilon_{2k+1}^{(n)}$ являются промежуточными и необходимы для вычисления величин с четными индексами [4].

У нас так же есть следующее уравнение:

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n)} - \frac{[H_{k+1}(S_n)]^2}{H_k(\Delta^2 S_n) H_{k+1}(\Delta^2 S_n)}. \tag{35}$$

Поскольку в рекурсивной формуле есть возможность идти с шагом 2, то члены с нечетным нижним индексом могут быть исключены.

Из правила (35) ε – алгоритма имеем:

$$\varepsilon_{k+1}^{(n)} - \varepsilon_{k-1}^{(n+1)} = \left(\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}\right)^{-1}, \tag{36}$$

$$\varepsilon_{k+1}^{(n+1)} - \varepsilon_{k-1}^{(n+2)} = (\varepsilon_k^{(n+2)} - \varepsilon_k^{(n+1)})^{-1}. \tag{37}$$

Вычитая первое соотношение из второго, левая часть становится

$$\left(\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}\right) - \left(\varepsilon_k^{(n+2)} - \varepsilon_k^{(n+1)}\right) = \left(\varepsilon_{k+2}^{(n)} - \varepsilon_k^{(n+1)}\right)^{-1} - \left(\varepsilon_k^{(n+1)} - \varepsilon_{k-2}^{(n+2)}\right)^{-1}, \qquad (38)$$

и мы получаем перекрестное правило:

$$\frac{1}{\varepsilon_{k+2}^{(n)} - \varepsilon_k^{(n+1)}} + \frac{1}{\varepsilon_{k-2}^{(n+2)} - \varepsilon_k^{(n+1)}} = \frac{1}{\varepsilon_k^{(n+2)} - \varepsilon_k^{(n+1)}} + \frac{1}{\varepsilon_k^{(n)} - \varepsilon_k^{(n+1)}}. \qquad (39)$$

при начальных условиях $\varepsilon_{-2}^{(n)} = \infty, \varepsilon_{-1}^{(n)} = 0$ и $\varepsilon_0^{(n)} = S_n$, для n = 0, 1, ….
Это правило касается таблицы, где сохранены только величины с меньшим индексом того же соотношения, пять величин, обозначаемых сторонами света и отображаемых следующим образом:

$$\begin{array}{ccc} & N = \varepsilon_k^{(n)} & \\ W = \varepsilon_{k-2}^{(n+2)} & C = \varepsilon_k^{(n+1)} & E = \varepsilon_{k+2}^{(n)} \\ & S = \varepsilon_k^{(n+2)}. & \end{array} \qquad (40)$$

и перекрестное правило записывается (где C – центр):

$$\frac{1}{N-C} + \frac{1}{S-C} = \frac{1}{W-C} + \frac{1}{E-C}\,. \qquad (41)$$

Когда две величины в знаменателе равны (или близки), происходит деление на ноль (или на малую величину). Основное правило, как показал Винн, позволяет избежать такую сингулярность[6] и продолжить вычисление [6,12]. А именно:

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

[6] Сингулярность - точка, в которой математическая функция стремится к бесконечности или имеет какие-либо иные нерегулярности поведения.

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

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

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

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

или мы можем это переписать, как $\varepsilon_{k+1}^{(n)} = \frac{a}{1 - \frac{\varepsilon_{k-3}^{(n+2)}}{\varepsilon_{k-1}^{(n+1)} - \varepsilon_{k-3}^{(n+2)}} + \frac{\varepsilon_{k-1}^{(n+2)}}{\varepsilon_{k-1}^{(n+1)} - \varepsilon_{k-1}^{(n+2)}} + \frac{\varepsilon_{k-1}^{(n)}}{\varepsilon_{k-1}^{(n+1)} - \varepsilon_{k-1}^{(n)}}}$,

где a – числитель дроби, написанной ранее. Отсюда видно, что если $\varepsilon_{k-2}^{(n+1)}$ и $\varepsilon_{k-2}^{(n+2)}$ близки по значению, то $\varepsilon_{k-1}^{(n-1)}$ является неопределенным, но $\varepsilon_{k}^{(n)}$, $\varepsilon_{k}^{(n+1)}$ и $\varepsilon_{k+1}^{(n)}$ вычисляются из основного правила и тождества, представленного выше. А если $\varepsilon_{k-2}^{(n+1)}$ и $\varepsilon_{k-2}^{(n+2)}$ равны, то перекрестное правило для алгоритма становится проще: $\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+2)} + \varepsilon_{k-1}^{(n)} - \varepsilon_{k-3}^{(n+2)}$ [13]. Для данного метода в статье [13] представлен код, его блок-схема представлена в Приложении 1.

Предположим, что условие $\forall n, H_k(\Delta S_n) \neq 0$ не выполняется. Тогда это означает, что существует, по крайней мере, один индекс n такой, что $H_k(\Delta S_{\tilde{n}}) = 0$.

Теперь реализуем преобразование Шенкса, используя ε – алгоритм без его основного правила (31).

Поскольку $\varepsilon_{2k-1}^{(n)} = \frac{H_k(\Delta^3 S_n)}{H_k(\Delta S_n)}$, мы имеем $\varepsilon_{2k-1}^{(\tilde{n})} = \infty$ неопределенным.

Далее предположим, что $H_k(\Delta S_{\tilde{n}}) = 0$. Таким образом, $\varepsilon_{2k-1}^{(\tilde{n}-1)} = \infty$ является неопределенным. То же самое с $\varepsilon_{2k}^{(\tilde{n}-1)} = \varepsilon_{2k-2}^{(\tilde{n})} + \frac{1}{(\varepsilon_{2k-1}^{(\tilde{n})} - \varepsilon_{2k-1}^{(\tilde{n}-1)})}$. Такое обоснование применимо и к $\tilde{n}$ +1.

Значит, индекс $\tilde{n}$ изолированный, такой, что $H_k(\Delta S_{\tilde{n}}) = 0$, то есть не может быть одновременно $\varepsilon_{2k-1}^{(\tilde{n})} = \infty$ и $\varepsilon_{2k-1}^{(\tilde{n}-1)} = \infty$ (или $\varepsilon_{2k-1}^{(\tilde{n}+1)} = \infty$).

Поскольку $\varepsilon_{2k-1}^{(\tilde{n})} = \infty$ или $\varepsilon_{2k-1}^{(\tilde{n}-1)} = \infty$, $\varepsilon_{2k}^{(\tilde{n})} = \varepsilon_{2k-2}^{(\tilde{n}+1)} + \frac{1}{(\varepsilon_{2k-1}^{(\tilde{n}+1)} - \varepsilon_{2k-1}^{(\tilde{n})})} = \varepsilon_{2k-2}^{(\tilde{n}+1)}$ и, аналогично, $\varepsilon_{2k}^{(\tilde{n}-1)} = \varepsilon_{2k-2}^{(\tilde{n})}$.

Теперь так как изначальное предположение состояло в том, что $\forall n, \varepsilon_{2k}^{(n)} = S$(что, очевидно, означает, что эти величины могут быть вычислены), оно содержит $\varepsilon_{2k}^{(\tilde{n})} = \varepsilon_{2k-2}^{(\tilde{n}+1)} = \varepsilon_{2k}^{(\tilde{n}-1)} = \varepsilon_{2k-2}^{(\tilde{n})} = S$. Отсюда следует, что при попытке вычисления следующих столбцов ε – алгоритма происходит деление на ноль, алгоритм должен быть остановлен, и эти столбцы не могут быть получены. Мы переходим к *теореме 3*.

Теорема 3.

Необходимым и достаточным условием того, что $\varepsilon_{2k}^{(n)} = S$ для всех n, является то, что последовательность S_n удовлетворяет (7) вместе с $a_0 \cdot a_k \neq 0$ и $a_0 + \cdots + a_k \neq 0$ [2].

Разница между *теоремой 1* и *теоремой 3* заключается в том, что *теорема 1* касается преобразования Шенкса, когда как *теорема 3* связана с рекурсивным алгоритмом, используемым для его реализации. Поскольку преобразование Шенкса может быть реализовано и другими алгоритмами, для их правильной работы, возможно, придется повторно ввести условие $H_k(\Delta S_n) \neq 0 \; \forall n$, отсутствующее в *теореме 3,* и тогда, *теорема 3* справедлива только для ε – алгоритма.

4. Сравнение алгоритмов

Сравним многошаговое преобразование Шенкса и ε – алгоритм. Оба алгоритма хорошо подходят для осциллирующих[7] последовательностей и плохо подходят для монотонных[8], т.к. происходит накопление ошибки [2]. Также преобразование Шенкса подходит для ускорения рядов с факториалами и тригонометрическими функциями, что показал Винн, изучая сходимость рядов Ньютона и Дирихле [2]. Но главным недостатком преобразования является затрата сил на вычисления определителей, чего нет у ε – алгоритма. ε – алгоритм используется для неявных и устойчивых методов интегрирования Рунге-Кутта, а также для решения краевых задач системы обыкновенных дифференциальных уравнений [8]. Но точно сказать, что последовательность, преобразованная с помощью ε –алгоритма, будет намного лучше сходится относительно искомой нельзя, ускорение может быть малозаметным [14].

Точность каждого метода зависит от множества факторов, таких как исходный ряд, метод ускорения и параметры алгоритма. В общем случае, чем больше количество членов ряда, тем точнее результат. Однако, точность алгоритма Шенкса может быть немного ниже, чем у алгоритма Винна. Такое различие появляется из-за разного подхода к вычислению $e_k(S_n)$. Эпсилон алгоритм основан на идее рекуррентного использования аппроксимаций сумм S_n для вычисления новых значений $e_k(S_n)$. Алгоритм Шенкса, с другой стороны, предлагает использование линейной комбинации трех соседних приближений.

Рассмотрим работу двух алгоритмов на конкретных примерах. Ниже предоставлены графики, демонстрирующие работу алгоритмов ускорения

---

[7] Осциллирующая последовательность — это последовательность, у которой значения оказываются близкими к какому-то числу через один номер, а через другой — к другому числу, при этом значения не стремятся к какому-то определенному числу.

[8] Монотонная последовательность — это последовательность, элементы которой с увеличением номера не возрастают, или, наоборот, не убывают.

сходимости рядов, используемых в проекте, для первых 25 членов каждого из рассматриваемых ниже рядов. В случае, если какая-то из функций расчета частичной суммы T_n ускоренного ряда выкидывает исключение (деление на ноль), график будет обрываться.

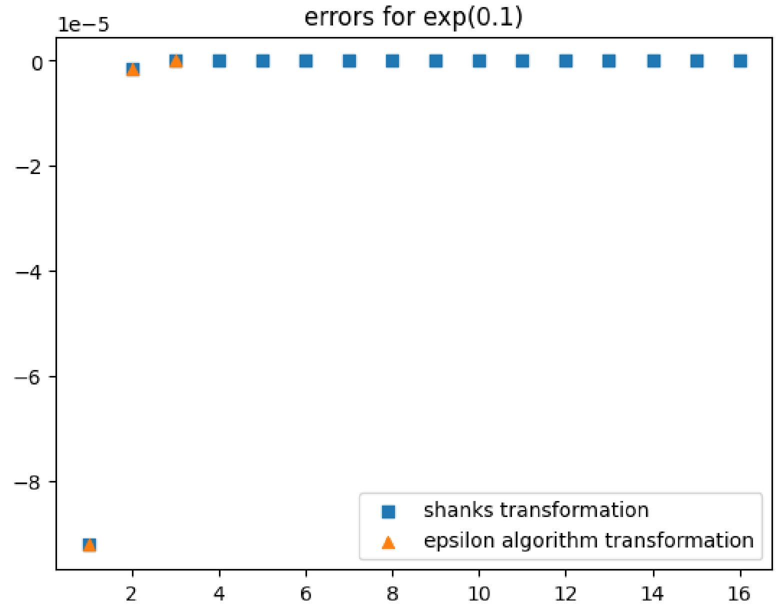

Рисунок 1. График ошибок $exp(0.1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 1. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
3	0	0

Время работы алгоритмов:
Алгоритм Шенкса – 0.0105918 секунд
ε-алгоритм – 0.0249597 секунд

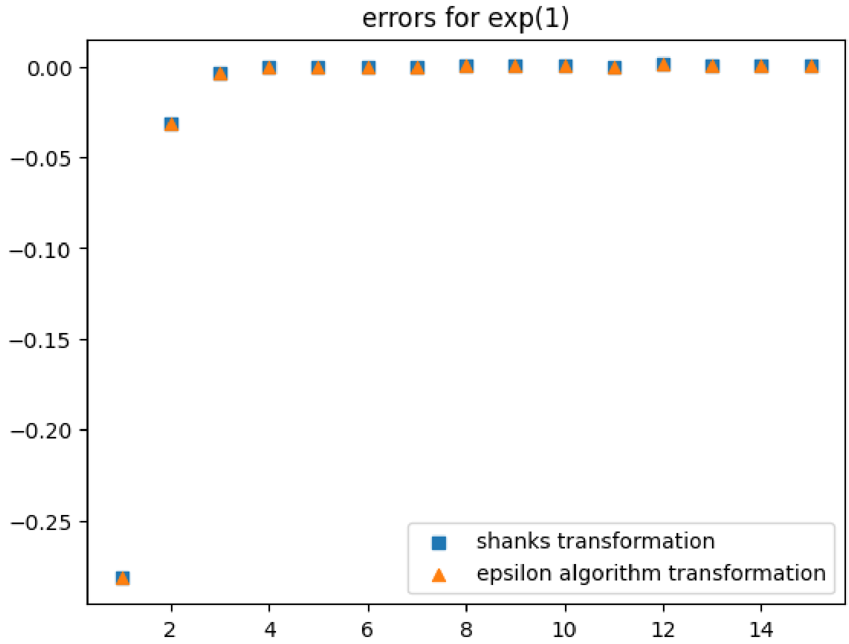

Рисунок 2. График ошибок $exp(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 2. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
15	0.000800848	0.000800848

Время работы алгоритмов:
Алгоритм Шенкса – 0.0115879 секунд
ε-алгоритм – 0.014777 секунд

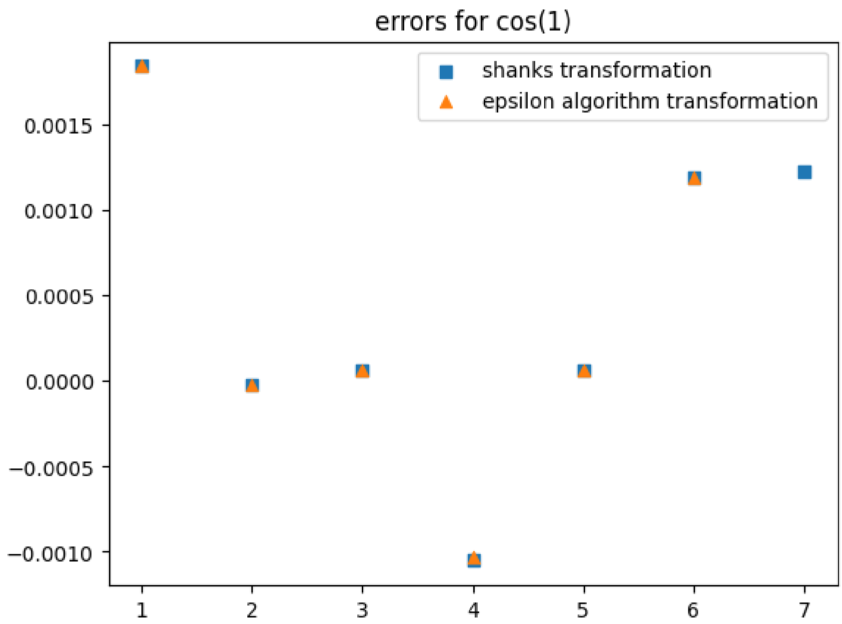

Рисунок 3. График ошибок $cos(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 3. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
6	0.0011903	0.0011903

Время работы алгоритмов:
Алгоритм Шенкса – 0.0180526 секунд
ε-алгоритм – 0.0274622 секунд

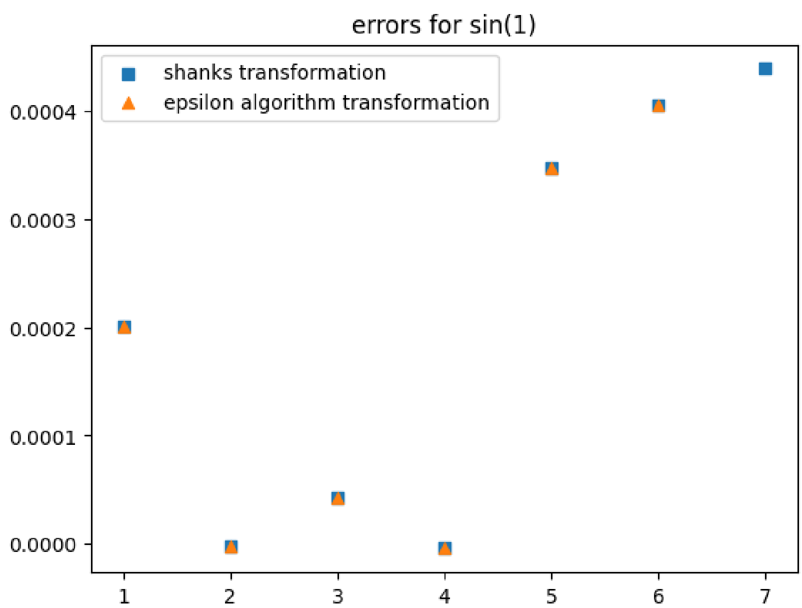

Рисунок 4. График ошибок $sin(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 4. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
6	0.000405312	0.00040549

Время работы алгоритмов:
Алгоритм Шенкса – 0.036316 секунд
ε-алгоритм – 0.0182762 секунд

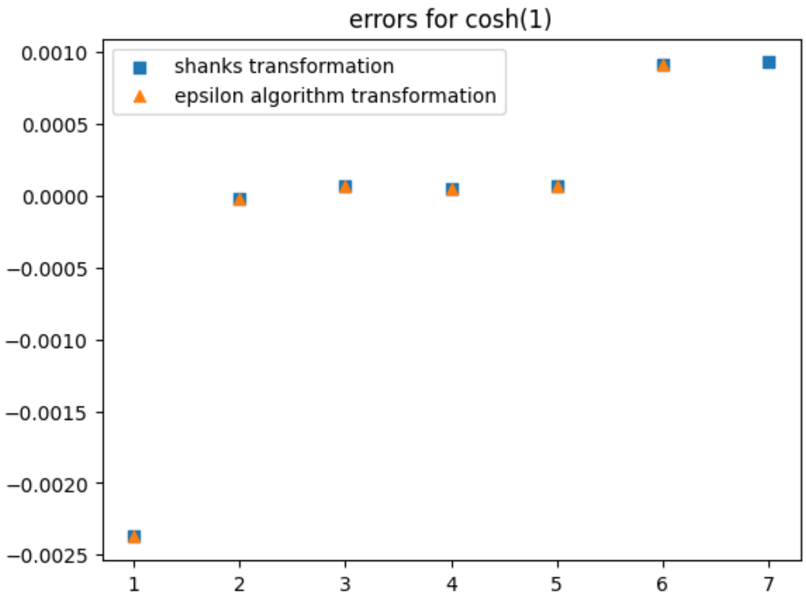

Рисунок 5. График ошибок $cosh(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 5. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
6	0.000911474	0.000911474

Время работы алгоритмов:
Алгоритм Шенкса – 0.0178368 секунд
ε-алгоритм – 0.0179569 секунд

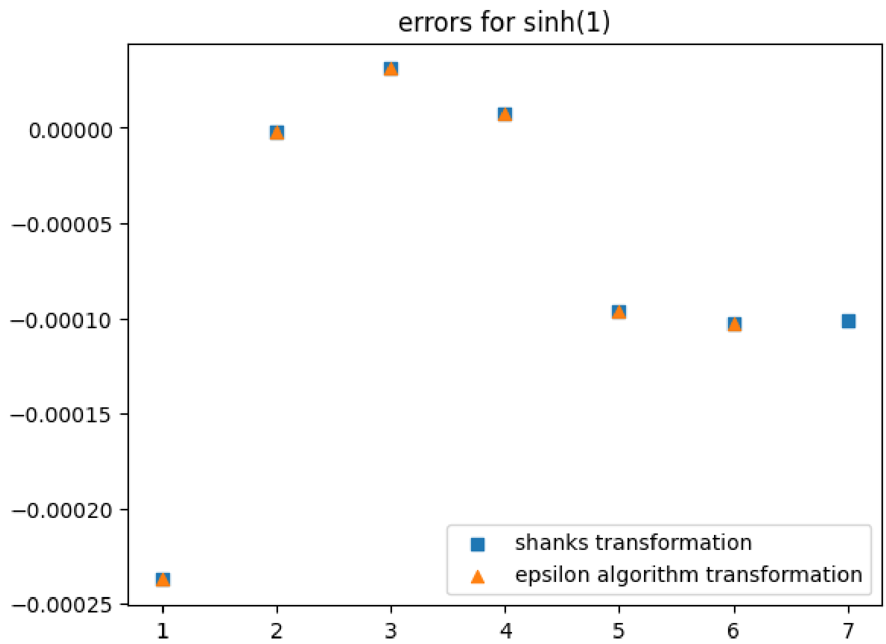

Рисунок 6. График ошибок $sinh(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 6. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
6	-0.000102878	-0.000102878

Время работы алгоритмов:
Алгоритм Шенкса – 0.0167468 секунд
ε-алгоритм – 0.0187359 секунд

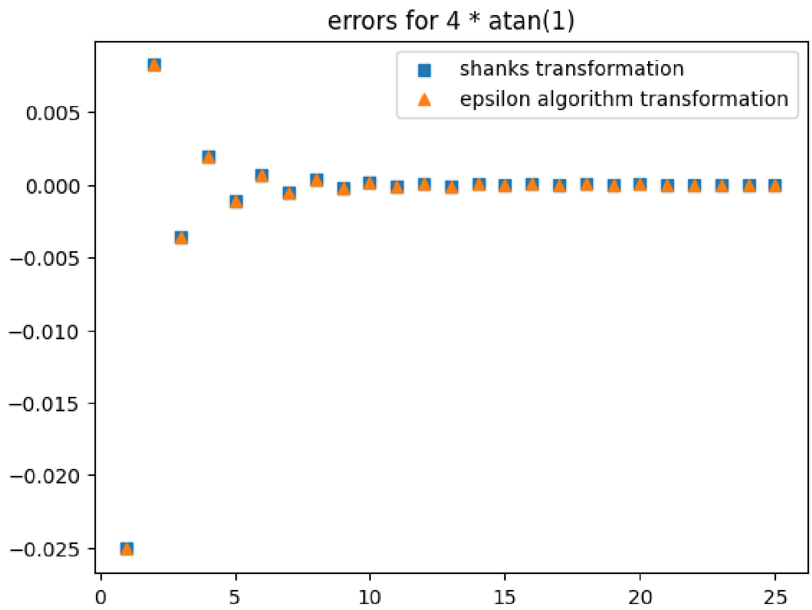

Рисунок 7. График ошибок $4atan(1)$ для преобразования Шенкса и ε-алгоритма.

Таблица 7. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	-1.40667e-05	-1.40667e-05

Время работы алгоритмов:
Алгоритм Шенкса – 0.0013494 секунд
ε-алгоритм – 0.0016063 секунд

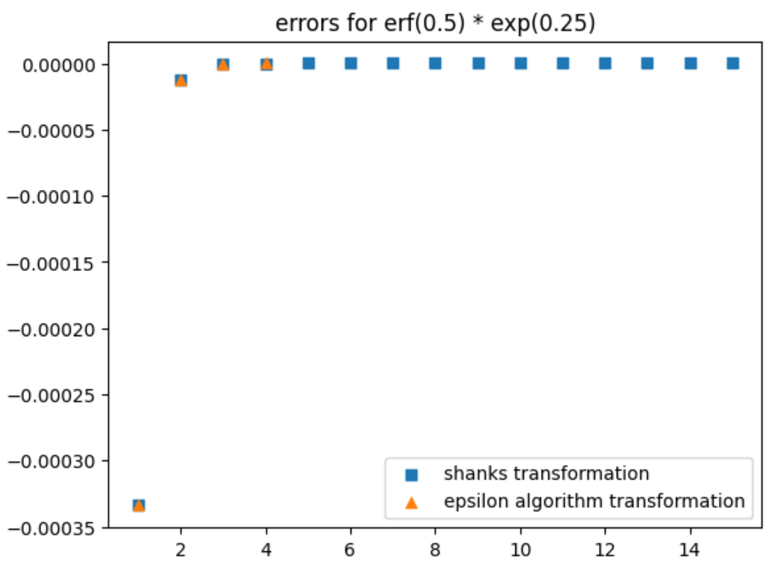

Рисунок 8. График ошибок $erf(0.5) \cdot exp(0.25)$ для преобразования Шенкса и ε-алгоритма.

Таблица 8. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
4	1.19209e-07	1.19209e-07

Время работы алгоритмов:
Алгоритм Шенкса – 0.0111114 секунд
ε-алгоритм – 0.0199663 секунд

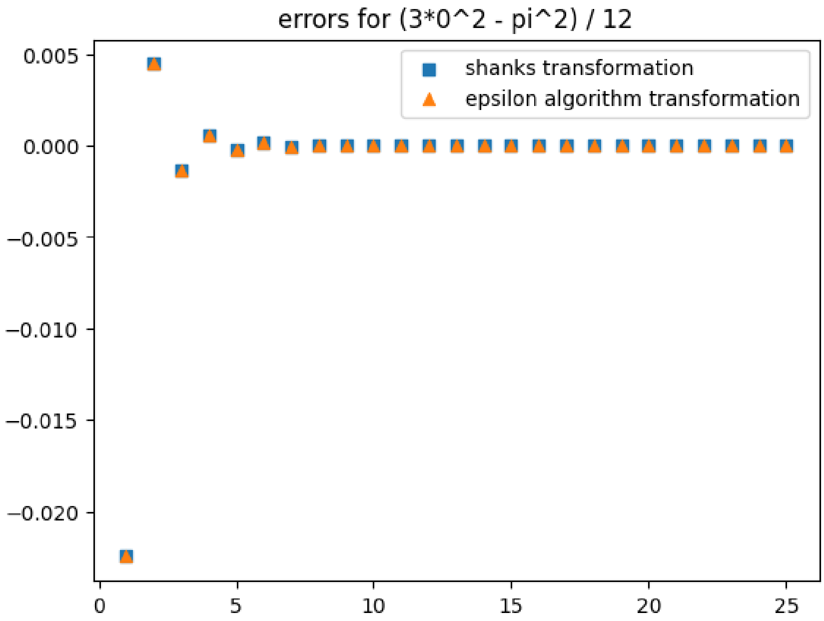

Рисунок 9. График ошибок $\frac{1}{12}(3x^2 - \pi^2)$ в точке $x = 0$ для преобразования Шенкса и ε-алгоритма.

Таблица 9. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	-5.96046e-07	-5.96046e-07

Время работы алгоритмов:
Алгоритм Шенкса – 0.0014506 секунд
ε-алгоритм – 0.0018257 секунд

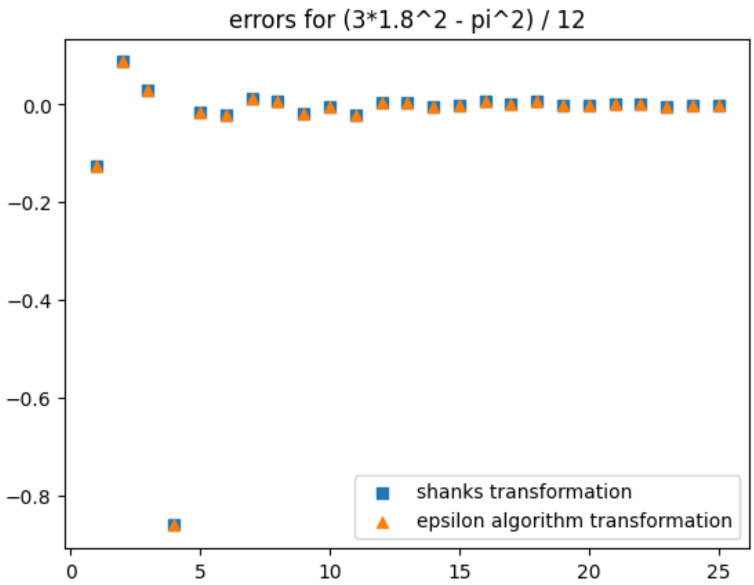

Рисунок 10. График ошибок $\frac{1}{12}(3x^2 - \pi^2)$ в точке $x = 1.8$ для преобразования Шенкса и ε-алгоритма.

Таблица 10. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	-0.00257336	-0.00257338

Время работы алгоритмов:
Алгоритм Шенкса – 0.0019258 секунд
ε-алгоритм – 0.0020634 секунд

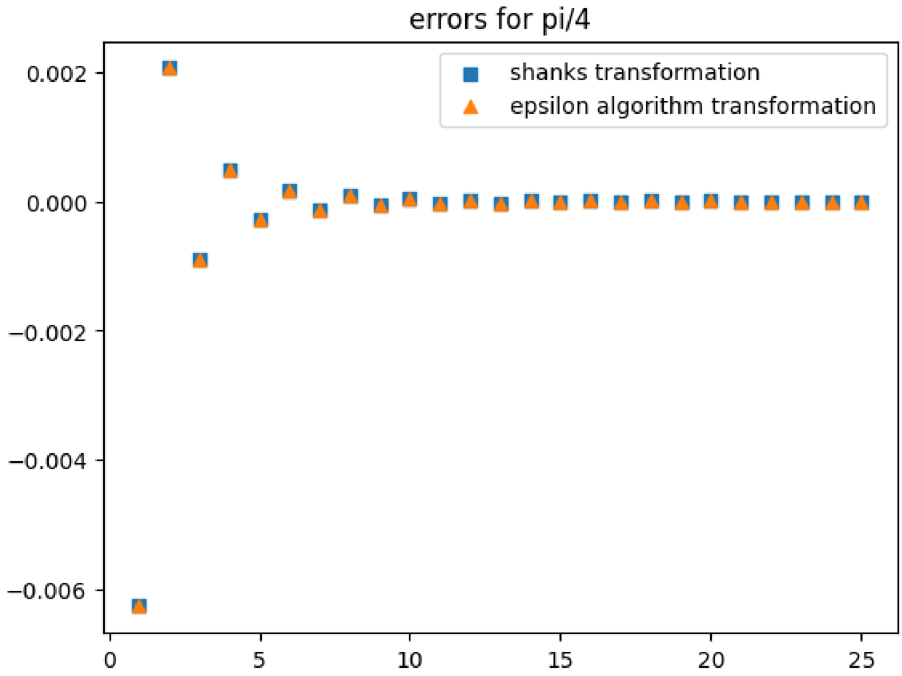

Рисунок 11. График ошибок $\sum_{i=1}^{\infty} \frac{(-1)^n}{2n+1}$

Таблица 11. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	-3.51667e-06	-3.51667e-06

Время работы алгоритмов:
Алгоритм Шенкса – 0.0014423 секунд
ε-алгоритм – 0.0009331 секунд

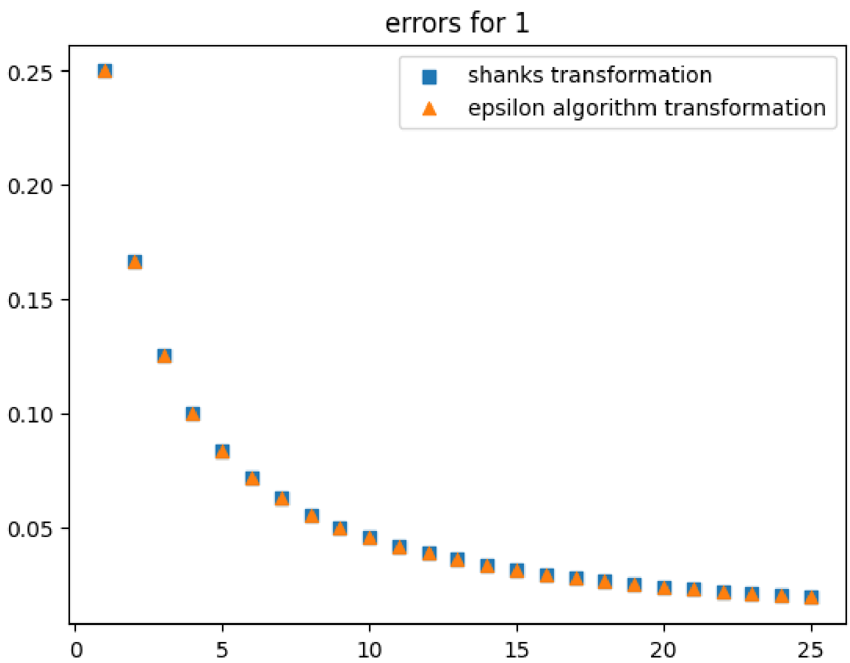

Рисунок 12. График ошибок $\sum_{i=1}^{\infty} \frac{1}{n^2+n}$

Таблица 12. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	0.0192305	0.0192314

Время работы алгоритмов:
Алгоритм Шенкса – 0.0015767 секунд
ε-алгоритм – 0.0014957 секунд

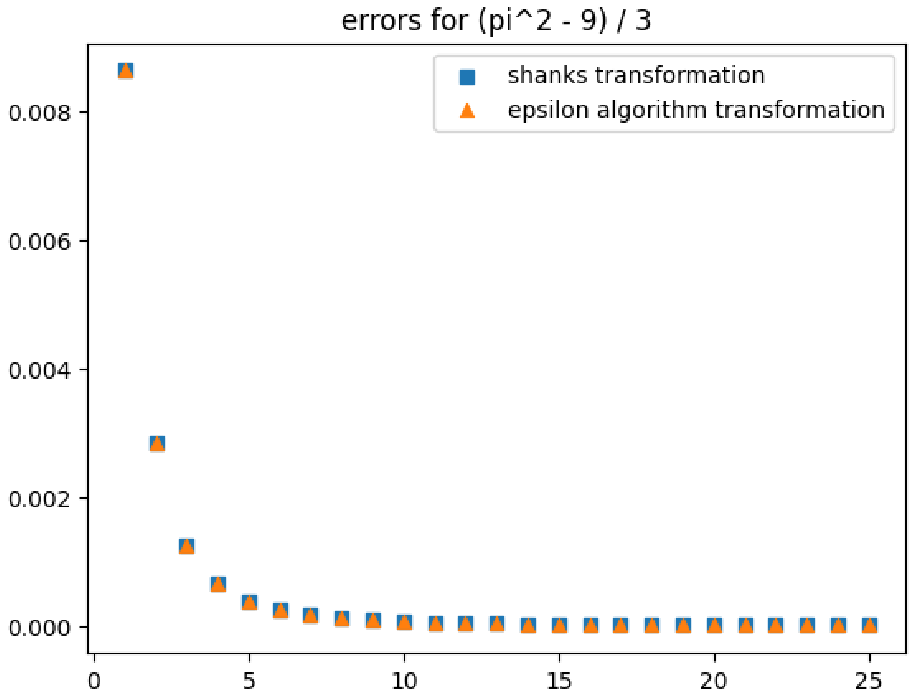

Рисунок 13. График ошибок $\sum_{i=1}^{\infty} \frac{1}{n^2(n+1)^2}$

Таблица 13. Сравнение преобразования Шенкса и ε-алгоритма на последней итерации.

номер итерации n	преобразования Шенкса	ε-алгоритм
25	4.70877e-06	4.38094e-06

Время работы алгоритмов:
Алгоритм Шенкса – 0.0024177 секунд
ε-алгоритм – 0.0017156 секунд

Значения разницы между настоящей суммой и аппроксимацией для двух методов показывают, что результаты идентичны, но алгоритм преобразования Шенкса часто может посчитать больше преобразования для бо́льшего n, прежде чем программа выдаст исключение *division by zero*. Скорее всего, это связано с тем, что в алгоритме преобразования Шенкса в знаменателе стоит разность членов ряда a_{n+1} - a_n, которая проходит проверку на отличие от нуля для бо́льших n, чем знаменатель разности промежуточных приближенных величин, полученных ε-алгоритмом. На степенных и тригонометрических рядах алгоритм преобразования Шенкса работает немного быстрее, а на чисто числовых рядах - эпсилон алгоритм.

Заключение

Метод Шенкса является мощным инструментом в численном анализе и численных методах решения уравнений. Он позволяет улучшить сходимость суммируемых рядов и повысить точность численных приближений. Из всех рассмотренных подходов к решению задачи нельзя сделать однозначный вывод о том, какой алгоритм лучше. У каждого есть свои преимущества.

Список литературы

1. The genesis and early developments of Aitken's process, Shanks' transformation, the ε–algorithm, and related fixed point methods // Claude Brezinski & Michela Redivo-Zaglia - 2018. – P. 11 - 69.

2. On condition numbers of the Shanks transformation // M.N. Senhadji - 1999. – P. 5 - 21.

3. Efficient Capacitance Extraction for Periodic Structures by Shanks Transformation // Ye Liu, Mei Xue, Zheng-Fan Li, Rui-Feng Xue - 2004. – P. 1 - 5.

4. An Extended Multistep Shanks Transformation and Convergence Acceleration Algorithm with Their Convergence and Stability Analysis // Jian-Qing Sun, Xiang-Ke Chang, Xing-Biao Hu, Yi He - 2013. – P. 8 - 25.

5. Non-linear transformations of divergent and slowly convergent sequences // D. Shanks – 1955. – P. 1–42.

6. Acceleration techniques in numerical analysis, with particular references to problems in one independent variable // P. Wynn – 1962. – P. 149–156.

7. An arsenal of Algol procedures for the evaluation of continued fractions and for effecting the epsilon algorithm // P. Wynn - 1966 – P. 327–362.

8. Generalisations de la transformatiost de Shanks, de la table de pade et de L' ε-algorithme // C. Brezinski – 1975. – P. 350-358.

9. Reanalysis of linear and nonlinear structures using iterated Shanks // Jorge Hurtado – 2002. – P. 4220 - 4222.

10. The Schur Complement and its Applications // Zhang - 2005. – P. 4 – 12.

11. On a device for computing the $e_m(S_n)$ transformation // P. Wynn – 1956. – P. 91-96.
12. Upon Systems of Recursions which Obtain Among the Quotients of the Pade Table // P. Wynn – 1965. – P. 3- 6.
13. Singular rules for certain non-linear algorithms // P. Wynn – 1963. – P. 175 – 195.
14. On the convergence and stability of the epsilon algorithm // P. Wynn – 1966. – P. 100 – 110.

Приложение 1

Блок-схема для реализации теоремы 2