

## Постановка математической задачи

Дано: медленно сходящаяся или расходящаяся последовательность  $\{A_n\}$ , где  $A_n$  — частичные суммы ряда или элементы последовательности. Наиболее эффективно алгоритм работает с последовательностями вида:

### 1. Логарифмические

$$A_n \sim A + \sum_{i=0}^{\infty} a_i n^{\gamma-i} \text{ при } n \rightarrow \infty; a_0 \neq 0; \gamma \neq 0, 1, 2 \dots \#(1)$$

Здесь  $A = \lim_{n \rightarrow \infty} A_n$  при  $\Re \gamma < 0$ . При  $\Re \gamma \geq 0$ ,  $A$  является антипределом  $\{A_n\}$

### 2. Линейные

$$A_n \sim A + \zeta^n \sum_{i=0}^{\infty} a_i n^{\gamma-i} \text{ при } n \rightarrow \infty; a_0 \neq 0; \zeta \neq 0 \#(2)$$

Здесь  $A = \lim_{n \rightarrow \infty} A_n$  в случаях: (a)  $|\zeta| < 1$  или (b)  $|\zeta| = 1$  и  $\Re \gamma < 0$ . В остальных случаях  $A$  является антипределом.

### 3. Факториальные

$$A_n \sim A + \frac{\zeta^n}{(n!)^r} \sum_{i=0}^{\infty} a_i n^{\gamma-i} \text{ при } n \rightarrow \infty; a_0 \neq 0; r = 1, 2 \dots \#(3)$$

Цель: Ускорить сходимость последовательности  $\{A_n\}$  к пределу  $A$  (или антипределу) с помощью итерированного  $W$ -преобразования Любкина, устраняя медленно убывающие или осциллирующие члены погрешности

## Алгоритм Любкина

## Реализация алгоритма

### ВХОД:

- n: количество членов частичной суммы (целое беззнаковое)
- order: порядок преобразования (целое беззнаковое)
- S\_n: частичная сумма ряда для n членов (вещественное)
- j: параметр фиксации (целое беззнаковое, обычно 0 при вызове)

### ВЫХОД:

- Результат W-преобразования (вещественное)

### ФУНКЦИЯ calculate(n, order, S\_n, j):

- ДОБАВИТЬ к S\_n члены ряда от (n+1) до (n+j):  
ДЛЯ i ОТ 0 ДО j-1:  
     $S_n += \text{series}(n + 1 + i)$   
n += j
- ЕСЛИ order == 0:  
    ВЕРНУТЬ S\_n
- РЕКУРСИВНО вычислить промежуточные значения:  
     $W0 = \text{calculate}(n, \text{order}-1, S_n, 0)$   
     $W1 = \text{calculate}(n, \text{order}-1, S_n, 1)$   
     $W2 = \text{calculate}(n, \text{order}-1, S_n, 2)$   
     $W3 = \text{calculate}(n, \text{order}-1, S_n, 3)$
- ВЫЧИСЛИТЬ разности:  
     $Wo0 = (W1 - W0)$   
     $Wo1 = (W2 - W1)$   
     $Wo2 = (W3 - W2)$   
     $Woo1 = Wo0 * (Wo2 - Wo1)$   
     $Woo2 = Wo2 * (Wo1 - Wo0)$
- ВЫЧИСЛИТЬ результат по оптимизированной формуле:  
     $\text{result} = W1 - (Wo1 * Woo1) / (Woo2 - Woo1)$
- ЕСЛИ result не является конечным числом:  
    ВЫБРОСИТЬ ошибку переполнения
- ВЕРНУТЬ result

### ОСНОВНАЯ ФУНКЦИЯ (оператор вызова):

- ЕСЛИ order < 0:  
    ВЫБРОСИТЬ ошибку "отрицательный порядок"

ВЕРНУТЬ calculate(n, order, series.S\_n(n), 0)

Рисунок 1. Псевдокод алгоритма Любкина

## Список литературы

1. *Sidi A.* A Convergence and Stability Study of the Iterated Lubkin Transformation and the  $\theta$ -Algorithm // Mathematics of Computation. – 2003. – Vol. 72, № 241. – P. 419–433. – URL: <http://www.jstor.org/stable/4099999> (дата обращения: 05.06.2024).