**Постановка математической задачи**

Дано: медленно сходящаяся последовательность , где – частичные суммы ряда

Условие сходимости:

1. Последовательность сходится к пределу (т.е. ), но делает это медленно.
2. Предполагается, что погрешность допускает разложение по степеням.

Возможные формы :

Алгоритм используется в последовательностях с полиномиальной погрешностью.

Цель: обеспечить более быструю сходимость ряда к по сравнению с исходной последовательностью путем последовательного исключения членов погрешности с помощью экстраполяции Ричардсона.

**Метод экстраполяции Ричардсона.**

Во многих проблемах бесконечную последовательность *{An}* можно соотнести с функцией *A(y)*. Она определена для *y⊂* (0,*b] (b>0)*, и y либо дискретен, либо

непрерывен.

После чего справедливо отношение *An = A(yn) (n*⊂ *0)* для некоторой монотонно убывающей последовательности *{yn*}⊂(0,*b]*, которая удовлетворяет

Тогда задача нахождения предела последовательности становится эквивалентной задаче нахождения предела функции, т.е.

Рассматривать функцию намного удобней, в отличие от последовательностей, так как существует обширный математический аппарат, который может помочь нам при анализе поведения функции.

Например, во многих случаях функция *A(y)* может иметь хорошо определённое расширение при *y*→0+, чья форма нам известна.

Рассмотрим функцию *A(y)*.

Мы не предполагаем, что обязательно существует. Если он существует, то он равен пределу *A*, если нет, то антипределу *A*.

В нашем случае пусть A(y) удовлетворяет равенству для некоторого *s*⊂ ℕ0:

где . Если вышерассмотренное равенство справедливо для любого s⊂ и *Re* σ*1*<Re σ*2*<… так, что

Тогда у А(y) есть асимптотическое расширение:

Замечание: ряд в правой части может не сходиться, и на практике он часто расходится.

σ*k* нам известны, α*k* нам неизвестны, и, в общем случае, они нам не нужны. Большой интерес представляет нахождение *A* будь то предел или антипредел. Из вышерассмотренного равенства можно выразить, что:

А потому, было бы неплохо избавиться от и получить более качественную аппроксимацию к А.

С этим поможет метод экстраполяции Ричардсона.

Возьмем константу ω⊂*(0,1)* и *y*’=ω*y*.

Тогда из вышерассмотренного равенства получаем:

Домножим на :

И вычтем его из изначального равенства, получим:

Поделим на :

Пусть:

Тогда мы получаем новую аппроксимацию:

Причем:

Так как , то полученная аппроксимация будет лучше приближать А.

Так можно продолжать много и много раз, получаю аппроксимации вида:

Причем:

При каждой итерации мы строим новую аппроксимацию, которая приближает A все лучше и лучше. Для экстраполяции Ричардсона существует рекурсивный алгоритм, который выводится из равенства (12) через индукцию.

Пусть Очевидно, что {ym} убывающая последовательность, которая стремится к нулю.

Алгоритм:

1. Пусть
2. Пусть сn=, тогда:

Из рекурсивного алгоритма видно, что организуют некую структуру, которую можно организовать в виде таблицы:

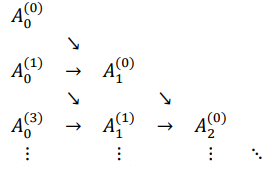


Рисунок 1 – Схема Ромберга

Схема, представленная выше называется таблицей Ромберга [1]. Стрелки означают поток вычислений.

Важное замечание: большое количество ускоряющих трансформаций организовываются в такие структуры, например, – трансформации Левина. Эти структуры могут быть многомерными.

**Grep**

Несмотря на практичность экстраполяционного процесса Ричардсона, его применение ограничено, т.е. класс последовательностей, к которым он может быть применён довольно узкий, поэтому было разработано обобщение GREP, решающее эту проблему.

Пусть

Возьмём убывающую положительную последовательность такую, что .

Пусть .

Тогда аппроксимации к А определены через линейную систему

* – вспомогательные N неизвестных

Видно, что формула получена из определения расширения функции, принадлежащей классу , заменой на асимптотическое расширение, которые мы отрезаем по .

Данное обобщение экстраполяционного процесса Ричардсона, которое генерирует , называется GREP(m).

GREP имеет несколько преимуществ перед экстраполяцией Ричардсона:

1. Вместо неизвестных констант теперь неизвестные гладкие функции , которые обладают асимптотическим расширением, форму которого мы знаем
2. Введены функции , которые не должны обладать какой-то определённой структурой и потому могут иметь различные темпы роста.
3. Функция A(y) представлена суммой асимптотических расширений.

Благодаря этому GREP имеет несколько преимуществ:

1. Более широкий класс функций, к которым может быть применён метод.
2. Так как в формуле присутствует конечное число функций , а функции , в сущности, представляют из себя полиномы, то это позволяет придумать алгоритмы, которые будут эффективными.
3. не являются уникальными, а потому они могут быть заменены другими функциями, имеющими расширение той же формы.

GREP также можно расширить на последовательности, у которых асимптотическое расширение функций имеет вид:

Где и они известны, также

Это асимптотическое расширение можно записать в общей форме:

Где функции образуют асимптотическую последовательность, т.е. .

Тогда расширение GREP примет вид:

Нетрудно заметить, что экстраполяционный метод Ричардсона есть ни что иное как расширение GREP(1):

Возьмем и , получим экстраполяционный метод Ричардсона:

**Реализация алгоритма**

|  |
| --- |
| Функция Richardson\_Transform(ряд, n, order):  Вход:  ряд - исходный ряд, для которого ускоряется сходимость  n - количество членов частичной суммы  order - порядок преобразования (не используется в текущей реализации)    Выход:  Ускоренная частичная сумма после преобразования Ричардсона  Если n < 0:  Вызвать ошибку "отрицательное число на входе"    Если n == 0:  Вернуть DEF\_UNDEFINED\_SUM (по умолчанию 0)  Создать таблицу e размером 2 x (n + 1), инициализированную нулями  Заполнить первую строку таблицы e[0] частичными суммами ряда:  Для i от 0 до n:  e[0][i] = S\_n(i) // S\_n(i) - частичная сумма ряда до i-го члена  Инициализировать a = 1  Для l от 1 до n:  a = a \* 4  b = a - 1  Для m от l до n:  // Вычисление преобразования Ричардсона  e[1][m] = (a \* e[0][m] - e[0][m - 1]) / b    Поменять местами e[0] и e[1]  Определить результат:  Если n четное:  res = e[0][n]  Иначе:  res = e[1][n]  Если res не является конечным числом:  Вызвать ошибку "деление на ноль"  Вернуть res |

**Рисунок 1** – Псевдокод экстраполяции Ричардсона

**Экстраполяция Ричардсона. Дополнительно об аппроксимации.**

Экстраполяцию Ричардсона можно рассматривать как общий метод повышения точности приближений, когда известна структура погрешности. Для улучшения аппроксимации, нам потребуется более глубокое понимание структуры погрешности. Поэтому начнём с разложений Тейлора для f(x ± h) вокруг точки x:

Отсюда получаем (более подробно об этом разложении пишет А. Самарский [2]):

Перепишем формулу (25) другом виде:

Где D – аппроксимация, а (величина, которую мы хотим аппроксимировать).

Выразим D:

При таком выражении D погрешность равна:

где еi обозначает коэффициент при hi в формуле (25), также отметим независимость коэффициентов от h. Мы предполагаем, что в общем случае ei≠0. Таким образом, мы получили аппроксимацию, основанную на значениях f(x) в точках x±h. Чтобы улучшить её, нам необходимо исключить e2h2 из погрешности. Реализовать это можно путем записи аппроксимации, основанной на значениях функции в других точках. Например:

Основная идея состоит в комбинации выражений (29) и (26) для исключения h2. Заметим, что после вычислений в формуле (29) коэффициент при h2 будет равен 4e2. Для получения аналогичного коэффициента в формуле (26) необходимо умножить обе части выражения на 4.

Вычтем выражения друг из друга и получим:

Таким образом нам удалось повысить точность аппроксимации за счет использования большего числа точек. Этот алгоритм можно продолжать и дальше, каждый раз убирая некоторые слагаемые и, тем самым, увеличивая точность вычисления.

Стоит отметить, что в формуле (29) можно использовать другие точки, к примеру h/2. Благодаря этому можно будет получать аппроксимации, основанные на других точках по схеме, описанной выше. Аналогичное описание алгоритма приведено в статье Дорона Леви [3] (p. 88)

Список литературы

1. Лукьяненко М. В., Численные методы / М. В. Лукьяненко. — [Электронный ресурс]. — URL: https://teach-in.ru/ (дата обращения: 11.04.2025).
2. Самарский А. А., Гулин А. В. Численные методы : учебное пособие / А. А. Самарский, А. В. Гулин. — Москва : Наука, 1989. — 432 с.
3. Introduction to Numerical Analysis // Levy D. – 2012. – P. 88-98.