

Численная аппроксимация рядов

Или как сойти с небес сквозь double в ад.

13 декабря 2023 г.

Суммирование рядов неоспоримо является одной из наиболее естественных и классических задач математического анализа. От разрешения рекуррентных соотношений до простых чисел и задач тысячелетия - многие вопросы в математике требуют непосредственного взаимодействия с рядами. Тем не менее изучение рядов всегда имело два крайне разных аспекта:

- *теоретический*, основанный на фундаментальных вопросах определения областей сходимости, нахождения аналитических решений, формирования оценок;
- *практический*, посвящённый непосредственному эффективному численному вычислению сумм самых разнообразных числовых последовательностей.

1. Теоретическая оценка рядов

Несмотря на то, что данная практическая работа будет в основном ориентирована на вычислительную сторону числовых рядов, мы не сможем далее обойтись без некоторых фундаментальных результатов классического анализа (см., например, [1]).

Определение. Выражение $a_1 + a_2 + \dots + a_n + \dots$ обозначается $\sum_{n=1}^{\infty} a_n$ и обычно называют *рядом* или *бесконечным рядом*. Элементы последовательности $\{a_n\}$ называют *членами ряда*.

Определение. Суммы $S_n = \sum_{k=1}^n a_k$ называют *частичными суммами ряда*. Если последовательность $\{S_n\}$ сходится, то ряд называется *сходящимся* и $\lim_{n \rightarrow \infty} S_n = S$ называется *суммой ряда*. Если последовательность $\{S_n\}$ не имеет предела, то ряд называется *расходящимся*.

Определение. Ряд $R_n = \sum_{k=n+1}^{\infty} a_k$, полученный отбрасыванием от исходного n первых членов, называется *остатком ряда*. При этом имеет место следующая простейшая оценка суммы ряда:

$$S = S_n + R_n \implies |S - S_n| = |R_n|.$$

Утверждение (Признак Лейбница). Пусть дан знакочередующийся ряд $S = \sum_{k=1}^{\infty} (-1)^k b_k : b_k \geq 0$. Тогда остаток сходящегося знакочередующегося ряда будет по модулю меньше первого отброшенного слагаемого: $|R_n| < b_{n+1}$.

Таким образом, при суммировании знакочередующихся рядов с некоторой заранее заданной точностью ϵ признак Лейбница зачастую используется как **критерий остановки**.

2. Double: Арифметика чисел с плавающей запятой

Числа с плавающей запятой (Floating-point numbers, [2]) есть экспоненциальная форма представления вещественных чисел в виде мантиссы и порядка. Наиболее часто используется стандарт IEEE 754, согласно которому двойная точность (Double) занимает 64 бита и использует 52-битную на мантиссу.

3.14159	e4
+ 2.41421	e-1
<hr/>	
3.14159	e4
+ 0.0000241421	e4
<hr/>	
= 3.1416141421	e4
<hr/>	
= 3.14161	e4

Численный расчёт рядов требует суммирования большого количества чисел с плавающей запятой, поэтому рассмотрим на простом примере некоторые особенности операции сложения таких представлений. Будем использовать упрощённую модель – десятичную систему чисел с плавающей запятой, использующую 6-значные мантиссы. В рамках этой модели попробуем сложить пару чисел: $31415.9 + 0.241421 = 3.14159e4 + 2.41421e-1$. Для этого приведём оба числа к одному порядку (e4), произведём суммирование и округлим результат до 6-значной мантиссы. Нельзя не заметить, что при таком построении процесса суммирования чисел с плавающей запятой последние

5 цифр (41421) второго слагаемого бесследно исчезают. Этот курьёзный феномен называется **ошибкой округления** (round-off error) и, как мы увидим далее, может иметь крайне деструктивный эффект.

Рассмотрим сумму трёх слагаемых: $31415.9 + 0.241421 + 0.0433013 = 31416.1 + 0.0433013$. Повторяя уже описанный процесс, нетрудно заметить, что эта сумма будет равна 31416.1. С другой стороны, можно произвести суммирование в ином порядке и получить $31415.9 + (0.241421 + 0.0433013) = 31415.9 + 0.284722 = 31416.2$. Таким образом, мы на простейшем контрпримере проиллюстрировали **отсутствие ассоциативности операции сложения** чисел с плавающей запятой.

	3.14161	e4
+	4.33013	e-2
<hr/>		
	3.14161	e4
+	0.00000433013	e4
<hr/>		
=	3.14161	e4

Простейшим способом борьбы с такими ошибками округления является превентивная *сортировка* последовательности и последующее суммирование чисел в порядке их возрастания. В общем случае этот метод не гарантирует повышения точности, но считается хорошей практикой. Более интересным решением является **алгоритм суммирования Кэхена** (Kahan summation, [3]), использующий принцип *компенсационного суммирования*.

```
double KahanSum(input) {
    double s = 0.0; //sum
    double c = 0.0; //compensation
    for (i=0; i< N; i++){
        double y = input[i] - c
        volatile double t = sum + y;
        volatile double z = t - sum;
        c = z - y;
        s = t;
    }
    return s;
}
```

Рассмотрим работу этого алгоритма на примере суммы $31415.9 + 0.241421 + 0.0433013$. В таблицах ниже приводятся все промежуточные значения.

Iteration 1.		Iteration 2.	
s =	3.14159 e4	s =	3.14161 e4
c =	0.00000 e0	c =	-4.14210 e-2
<hr/>		<hr/>	
y =	2.41421 e-1	y =	7.14733 e-2
t =	3.14161 e4	t =	3.14162 e4
z =	2.00000 e-1	z =	1.00000 e-1
<hr/>		<hr/>	
c =	-4.14210 e-2	c =	2.85267 e-2

В процессе суммирования Кэхена “потерянные” цифры высчитываются и сохраняются в компенсирующем слагаемом c , после чего они вычитаются из последующего слагаемого. Так, в нашем примере после первой итерации ($31415.9 + 0.241421$) переменная $c = -4.14210e-2$ и хранит в себе незначительную часть второго слагаемого 0.241421. Стоит заметить, что, хотя такой метод суммирования и лучше “наивного” суммирования, он всё равно может приводить к довольно значительным вычислительным неточностям. Так, большинство модификаций алгоритма суммирования Кэхена оказываются малоэффективны при **расчёте разницы двух почти равных чисел** (Catastrophic cancellation). Проблематичность этого феномена заключается в том, что при вычитании двух близких чисел (например $3.14159e4 - 3.141597e4 = 2.00000e-1$) значимыми цифрами в мантиссе становятся те цифры изначальных слагаемых, которые до этого не являлись значимыми и не участвовали в расчётах.

3. Практические сложности при расчёте рядов

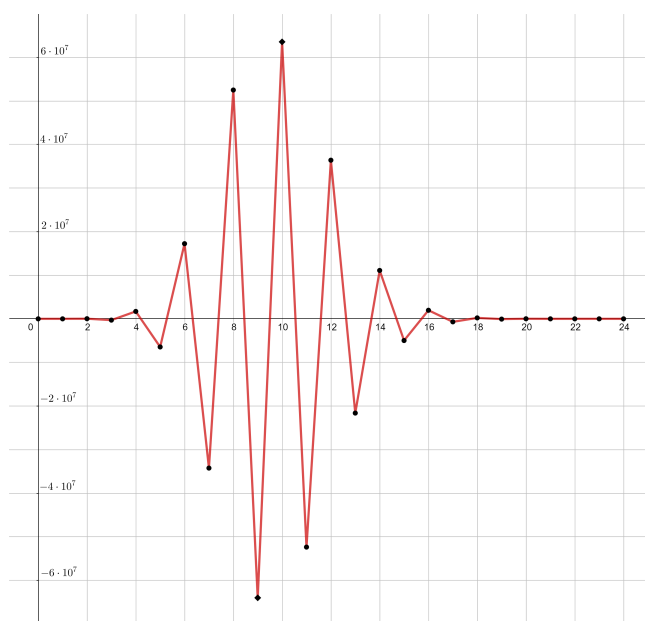


Рис. 1. Коэффициенты $a_n(x_0)$ ряда Маклорена (1)

Высокоточная и эффективная аппроксимация конечных сумм является довольно специфичной областью математической и компьютерной науки (см. [4]). Существует множество различных алгоритмов и предложений, нацеленных как на ускорение сходимости рядов (см., например, метод Эйлера-Абеля, ϵ -алгоритм Винна и другие), так и на увеличение точности расчётов. В данной практической работе мы постарались ознакомить читателя с базовыми методами уменьшения вычислительных погрешностей, характерных для численного суммирования. *Ошибки округления* могут приводить к понижению точности расчётов многих рядов, особенно в случае их знакопостоянства. Особенно легко с этим столкнуться, вычисляя значения функций в точках $x : |x| \gg 0$, посредством их ряда Маклорена. Тем не менее **catastrophic cancellation** зачастую имеет более деструктивное влияние на точность расчётов.

$$\sin(x) = \sum_{n=0}^{\infty} a_n(x) = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{(2n+1)!}. \quad (1)$$

Рассмотрим простейший пример - вычисление (по формуле 1) с точностью $\varepsilon = 10^{-12}$ значения функции $\sin(x)$ в точке $x_0 = 13\pi/2$. Из признака Лейбница можно определить, что для достижения заданной точности нам потребуется 38 слагаемых. Тем не менее при непосредственном “наивном” вычислении такой аппроксимации точного значения $\sin(13\pi/2) = 1$ мы получаем значение $s = 1.00000000485172$. Иными словами, на практике точность аппроксимации оказалась на несколько порядков меньше ожидаемой теоретической: $\delta \approx 4.85 \cdot 10^{-9} > \varepsilon$. Это поясняется накапливанием вычислительных погрешностей, порождённых необходимостью складывать числа схожие по порядку, но отличающиеся по знаку (см. рис. 1).

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n! (2n+1)}.$$

Конечно, в реальной практике трудности, описанные в прошлом примере, разрешаются банально просто – используется периодичность синуса и расчёты производятся для точек отрезка $[-\pi; \pi]$. Но что делать в более специфичных случаях? Например, как рассчитать с высокой точностью значение функции *ошибок* $\operatorname{erf}(x)$, имеющей огромную значимость в теории вероятности и математической статистике? Периодичность здесь уже не применить, а вот проблемы возникают примерно того же типа, что и при вычислении $\sin(x)$... В действительности решение вопросов такого рода зачастую требует изучения и применения новых представлений, аппроксимаций решений соответствующих ОДУ и использования асимптотических аппроксимаций, а это уже не входит в рамки предлагаемой практической работы.

4. Условия вариантов и рекомендации

Задание 1. Суммирование знакопередающихся числовых последовательностей:

Цель данной задачи состоит в ознакомлении с описанными выше методами на примере суммирования знакопередающегося функционального ряда при двух разных значениях переменного x . Требуется:

1. Определить по признаку Лейбница количество слагаемых n , необходимое для достижения теоретической оценки $\varepsilon = 10^{-12}$ максимального отклонения конечной суммы от точного значения ряда;
2. Аппроксимировать значение ряда Σ_i , используя: “наивное” суммирование; суммирование в порядке возрастания слагаемых; алгоритм суммирования по Кэхену. В каждом случае рассчитать отклонение δ_i от известного точного значения S .

Отразить в отчёте полученные значения n , Σ_i , δ_i (точность записи: 15 значимых цифр) и сделать выводы о точности аппроксимаций. Дополнительно предоставить файл с исходным кодом. Для повышения точности расчётов рекомендуется производить рекуррентный пересчёт коэффициентов ряда, а не прямые вычисления, основанные на явных формулах (см., например, прикреплённые файлы).

Рекомендация: На практике оказывается куда удобнее и разумнее рассчитывать коэффициенты рядов Тейлора рекуррентным образом. Это уменьшает количество операций и в некоторых случаях позволяет увеличить точность расчётов.

№	Знакопередающийся ряд	Точка 1		Точка 2	
1	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n}}{(2n+1)!}$	$x = \frac{\pi}{2}$	$S = \frac{2}{\pi}$	$x = \frac{15\pi}{2}$	$S = \frac{-2}{15\pi}$
2	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n-1}}{(2n)!}$	$x = \pi$	$S = \frac{-1}{\pi}$	$x = 9\pi$	$S = \frac{-1}{9\pi}$
3	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{n-1}}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = 11$	$S = \frac{1}{11e^{11}}$
4	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-2(n+1)}}{4^n (2n+1) (n!)^2}$	$x = 100$	$S = \frac{1}{100} \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{9}{8}$	$S = \frac{8}{9} \operatorname{arsh} \left(\frac{8}{9} \right)$
5	$\frac{\ln(2x)}{x} - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{2n+1}}$	$x = 100$	$S = \frac{1}{100} \operatorname{arsh}(100)$	$x = \frac{9}{8}$	$S = \frac{8}{9} \operatorname{arsh} \left(\frac{9}{8} \right)$
6	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n+1}}{(2n+1)!}$	$x = \sqrt{\frac{\pi}{2}}$	$S = \sqrt{\frac{2}{\pi}}$	$x = \sqrt{\frac{15\pi}{2}}$	$S = -\sqrt{\frac{2}{15\pi}}$

№	Знакопередающий ряд	Точка 1		Точка 2	
7	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n-1}}{(2n)!}$	$x = \sqrt{\pi}$	$S = \frac{-1}{\sqrt{\pi}}$	$x = 3\sqrt{\pi}$	$S = \frac{-1}{3\sqrt{\pi}}$
8	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n-1}}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = \sqrt{11}$	$S = \frac{1}{\sqrt{11} e^{11}}$
9	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-4n-3}}{4^n (2n+1) (n!)^2}$	$x = 10$	$S = \frac{1}{10} \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{3}{2\sqrt{2}}$	$S = \frac{2\sqrt{2}}{3} \operatorname{arsh} \left(\frac{8}{9} \right)$
10	$\frac{\ln(2x^2)}{x} - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{4n+1}}$	$x = 10$	$S = \frac{1}{10} \operatorname{arsh}(100)$	$x = \frac{3}{2\sqrt{2}}$	$S = \frac{2\sqrt{2}}{3} \operatorname{arsh} \left(\frac{9}{8} \right)$
11	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{(2n+1)!}$	$x = \frac{\pi}{2}$	$S = 1$	$x = \frac{13\pi}{2}$	$S = 1$
12	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n}}{(2n)!}$	$x = \pi$	$S = -1$	$x = 7\pi$	$S = -1$
13	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^n}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = 10$	$S = \frac{1}{e^{10}}$
14	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-2n-1}}{4^n (2n+1) (n!)^2}$	$x = 100$	$S = \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{8}{7}$	$S = \operatorname{arsh} \left(\frac{7}{8} \right)$
15	$\ln(2x) - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{2n}}$	$x = 100$	$S = \operatorname{arsh}(100)$	$x = \frac{8}{7}$	$S = \operatorname{arsh} \left(\frac{8}{7} \right)$
16	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n+2}}{(2n+1)!}$	$x = \sqrt{\frac{\pi}{2}}$	$S = 1$	$x = \sqrt{\frac{13\pi}{2}}$	$S = 1$
17	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n}}{(2n)!}$	$x = \sqrt{\pi}$	$S = -1$	$x = \sqrt{7\pi}$	$S = -1$
18	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n}}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = \sqrt{10}$	$S = \frac{1}{e^{10}}$
19	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-4n-2}}{4^n (2n+1) (n!)^2}$	$x = 10$	$S = \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{2\sqrt{2}}{\sqrt{7}}$	$S = \operatorname{arsh} \left(\frac{7}{8} \right)$
20	$\ln(2x^2) - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{4n}}$	$x = 10$	$S = \operatorname{arsh}(100)$	$x = \frac{3}{2\sqrt{2}}$	$S = \operatorname{arsh} \left(\frac{8}{7} \right)$
21	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2(n+1)}}{(2n+1)!}$	$x = \frac{\pi}{2}$	$S = \frac{\pi}{2}$	$x = \frac{11\pi}{2}$	$S = -\frac{11\pi}{2}$
22	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{(2n)!}$	$x = \pi$	$S = -\pi$	$x = 8\pi$	$S = 8\pi$
23	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{n+1}}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = 8$	$S = \frac{8}{e^8}$
24	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-2n}}{4^n (2n+1) (n!)^2}$	$x = 100$	$S = 100 \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{10}{9}$	$S = \frac{10}{9} \operatorname{arsh} \left(\frac{9}{10} \right)$
25	$x \ln(2x) - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{2n-1}}$	$x = 100$	$S = 100 \operatorname{arsh}(100)$	$x = \frac{10}{9}$	$S = \frac{10}{9} \operatorname{arsh} \left(\frac{10}{9} \right)$
26	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n+3}}{(2n+1)!}$	$x = \sqrt{\frac{\pi}{2}}$	$S = 1$	$x = \sqrt{\frac{11\pi}{2}}$	$S = -\sqrt{\frac{11\pi}{2}}$

№	Знакопередающийся ряд	Точка 1		Точка 2	
27	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{4n+1}}{(2n)!}$	$x = \sqrt{\pi}$	$S = -\sqrt{\pi}$	$x = 2\sqrt{2\pi}$	$S = 2\sqrt{2\pi}$
28	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{n!}$	$x = 1$	$S = \frac{1}{e}$	$x = 2\sqrt{2}$	$S = \frac{2\sqrt{2}}{e^8}$
29	$\sum_{n=0}^{\infty} (-1)^n \cdot \frac{(2n)! x^{-4n-1}}{4^n (2n+1) (n!)^2}$	$x = 10$	$S = 10 \operatorname{arsh} \left(\frac{1}{100} \right)$	$x = \frac{\sqrt{10}}{3}$	$S = \frac{\sqrt{10}}{3} \operatorname{arsh} \left(\frac{9}{10} \right)$
30	$x \ln(2x^2) - \sum_{n=1}^{\infty} \frac{(-1)^n (2n-1)!!}{2n (2n)!! x^{4n-1}}$	$x = 10$	$S = 10 \operatorname{arsh}(100)$	$x = \frac{\sqrt{10}}{3}$	$S = \frac{\sqrt{10}}{3} \operatorname{arsh} \left(\frac{10}{9} \right)$

Задание 2. Табуляция специальных функций:

Цель данной задачи состоит в применении изученных методов к более сложным объектам и кратком знакомстве с многими распространёнными специальными функциями. В каждом варианте требуется реализовать табуляцию заданной специальной функции на равномерной сетке $\omega = \{x = a + j \cdot \Delta, j = 0, \dots, 100\}$, где шаг разбиения $\Delta = \frac{b-a}{100}$.

- **Знакопередающиеся ряды:** необходимо использовать принцип Лейбница для определения числа слагаемых n , обеспечивающего ожидаемую точность $\varepsilon = 10^{-8}$. Если $n > 100$, использовать только первые 100 слагаемых. В каждой точке рассчитать сумму ряда, используя “наивное” суммирование (`sum_1`) и предварительную сортировку слагаемых (`sum_2`). Результаты расчётов предоставить в `csv` файле структуры "`x_i, n, sum_1, sum_2`" с 15 значимыми цифрами;
- **Варианты 3-8, 15-18, 27-30:** На основании первых 100 слагаемых рассчитать сумму ряда, используя “наивное” суммирование (`sum_1`) и алгоритм суммирования Кэхена (`sum_2`). Результаты расчётов предоставить в `csv` файле структуры "`x_i, sum_1, sum_2`" с 15 значимыми цифрами.

В отчёте сделать выводы о точности аппроксимаций и возникающих трудностях. Дополнительно предоставить файл с исходным кодом. *Замечание:* Далее $\gamma = 0.5772156649\dots$ есть константа Эйлера-Маскерони.

№	Функция	Границы отрезка	
1	$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n! (2n+1)}$	$a = 0$	$b = 10$
2		$a = -10$	$b = 0$
3	$\zeta(x) = \sum_{n=1}^{\infty} \frac{1}{n^x}$	$a = 2$	$b = 4$
4		$a = 5$	$b = 10$
5	$B(x, \pi) = \frac{1}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{n! (x+n)} \prod_{k=0}^n (\pi - k)$	$a = 1$	$b = 2$
6		$a = 2$	$b = 3$
7	$B(x, e) = \frac{1}{e} \sum_{n=0}^{\infty} \frac{(-1)^n}{n! (x+n)} \prod_{k=0}^n (e - k)$	$a = 1$	$b = 2$
8		$a = 2$	$b = 3$
9	$J_0(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(n!)^2 4^n}$	$a = 0$	$b = 20$
10		$a = -20$	$b = 0$
11	$J_1(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n! (n+1)! 2^{2n+1}}$	$a = 0$	$b = 20$
12		$a = -20$	$b = 0$
13	$J_2(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2(n+1)}}{n! (n+2)! 4^{n+1}}$	$a = 0$	$b = 20$
14		$a = -20$	$b = 0$
15	$\operatorname{li}(x) = \gamma + \ln(\ln x) + \sum_{n=1}^{\infty} \frac{(\ln x)^n}{n \cdot n!}$	$a = 10$	$b = 20$
16		$a = 5$	$b = 10$

№	Функция	Границы отрезка	
17	$\text{Ei}(x) = \gamma + \ln x + \sum_{n=1}^{\infty} \frac{x^n}{n \cdot n!}$	$a = 10$	$b = 20$
18		$a = 5$	$b = 10$
19	$\text{Si}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)(2n+1)!}$	$a = 0$	$b = 25$
20		$a = -25$	$b = 0$
21	$\text{Ci}(x) = \gamma + \ln x + \sum_{n=1}^{\infty} \frac{(-1)^n x^{2n}}{(2n)(2n)!}$	$a = 5$	$b = 10$
22		$a = 10$	$b = 20$
23	$S(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{4n+3}}{(4n+3)(2n+1)!}$	$a = 0$	$b = 10$
24		$a = -10$	$b = 0$
25	$C(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{4n+1}}{(4n+1)(2n)!}$	$a = 0$	$b = 10$
26		$a = -10$	$b = 0$
27	$\text{Shi}(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)(2n+1)!}$	$a = 0$	$b = 25$
28		$a = -25$	$b = 0$
29	$\text{Chi}(x) = \gamma + \ln x + \sum_{n=1}^{\infty} \frac{x^{2n}}{2n(2n)!}$	$a = 5$	$b = 10$
30		$a = 10$	$b = 20$

Направления для развития.

- Все специальные функции из второго задания можно рассчитать, используя встроенный функционал некоторых математических программных пакетов. В случае языка C++ для этого можно использовать библиотеки `boost` и/или `ALGLIB`. Таким образом, факультативным расширением данной практической работы является расчёт значений искомых функций (`f_i`), используя их библиотечные реализации, и оценка отклонений результатов, полученных в задаче 2: `d_1 = f_i - sum_1`; `d_2 = f_i - sum_2`.
- В тех случаях, когда высокая точность играет куда большую роль чем скорость расчётов, с вычислительными погрешностями можно бороться переходя к более “вместительным” типам данных. Так, формат IEEE Quad использует 128 битов (из которых 113 есть мантисса) для хранения чисел с плавающей запятой. Программная реализация этого формата для языка C++ доступна, например, в библиотеке `boost` и достаточно проста в использовании.

Список литературы

- [1] В.А. Зорич, *Математический Анализ*, **2**, МЦНМО, Москва, 110–121.
- [2] D. Goldberg, “What every computer scientist should know about floating-point arithmetic”, *ACM Computing Surveys*, **23**:1, 5–48.
- [3] W. Kahan, “Further remarks on reducing truncation errors”, *Communications of the ACM*, **8**:1 (1965), 40.
- [4] Computational Science Stack Exchange, *How to avoid the round-off errors in the larger calculations?*, (version: 2015-12-05), <https://scicomp.stackexchange.com/q/21483>.

L^AT_EX