1 Цель

Требуемый функционал и примерное описание интерфейса. Программа должна иметь расчетный модуль, написанный на языке Haskell, который расчитывает значения семейства функций Бесселя: функции Инфельда и функции Макдональда.

Программа должна позволять вводить необходимый тип функции, ее порядок и аргумент, и выводить для них значение функции с точностью не хуже десяти знаков после запятой для порядков функции 0, 1 и 2, и с точностью не хуже пяти знаков после запятой для остальных порядков для проверки по таблицам.

Также программа должна позволять вводить интервалы значений аргумента и порядка для построения графиков, в том числе двухмерных. При построении двухмерных графиков серии значений для изменяющегося аргумента и постоянного порядка разделяются пустыми строками. Пример:

```
#Порядок  Аргумент  Значение
   0         0         0
   0         1         2
   0         2         3

   1         0         1
   1         1         3
   1         2         4
```

Все далее приведенные формулы будут иметь ссылку на таковые из справочника по специальным функциям под редакцией Абрамовица.

2 Порядок

Предпочтительный порядок воплощения требуемых функций:

1. функция Инфельда, 9.6.10.

2. функция Макдональда, 9.6.11 ($\nu \in Z$), иначе 9.6.2.

Здесь используется гамма-функция, ее описание будет позже отдельно.

Асимптотика. Для увеличения диапазона поддерживаемых аргументов требуется использовать асимптотические продолжения при больших значениях аргумента: 9.7.1 и 9.7.2.

Рекуррентные выражения. При запросе функций отрицательных порядков и функций порядка выше трех базовая формула не используется, для расчета требуется свести функцию к сумме функций порядка 0 и 1. Предлагается использовать рекуррентные выражения 9.6.26. Это выражение нужно использовать рекурсивно до тех пор, пока порядки используемых функций не станут нужными.

Гамма-функция. Для реализации функций Бесселя нецелых порядков требуется воплотить Гамма-функцию, она — близкий родственник факториала, но является его обобщением по области определения и значений с множества натуральных чисел плюс ноль на множество действительных чисел. Воплощение предлагается выполнить по формулам 6.1.15 и 6.1.34.

$$\begin{aligned} &\Gamma(z) = (z-1)\Gamma(z-1); \\ &\frac{1}{\Gamma(z)} = \sum_{k=0}^{\infty} c_k z^k \quad (|z| < \infty). \end{aligned} \tag{1}$$

На самом деле, применение второй формулы лучше ограничить интервалом $z \in [1, 2]$, приводя остальные значения к указанным через первую — рекуррентную формулу. Значения c_k:

$\#k$	c_k
1	1.00000 00000 000000
2	0.57721 56649 015329
3	-0.65587 80715 202538
4	-0.04200 26350 340952
5	0.16653 86113 822915
6	-0.04219 77345 555443
7	-0.00962 19715 278770
8	0.00721 89432 466630
9	-0.00116 51675 918591
10	-0.00021 52416 741149
11	0.00012 80502 823882
12	-0.00002 01348 547807
13	-0.00000 12504 934821
14	0.00000 11330 272320
15	-0.00000 02056 338417
16	0.00000 00061 160950
17	0.00000 00050 020075
18	-0.00000 00011 812746
19	0.00000 00001 043427
20	0.00000 00000 077823
21	-0.00000 00000 036968
22	0.00000 00000 005100
23	-0.00000 00000 000206
24	-0.00000 00000 000054
25	0.00000 00000 000014
26	0.00000 00000 000001

Поправка: на самом деле, можно брать базовый интервал например в пределах $z \in [0 : 1.2]$ или близких, на точности это даже скажется отчасти в большую сторону.

Общий способ расчета:

1. Определяем требуемый тип функции

2. Смотрим на порядок функции, если больше 1 или меньше нуля, то применяем рекуррентные выражения для сведения к порядкам от 0 до 2 (0 и 1 в случае целых порядков)

3. Смотрим на аргумент функции, если не превышает порога асимптотики (определяете его сами на основе критерия плавного сшивания), то используется базовая формула, если превышает, то используем асимптотику.

4. После того, как предыдущее сделано, добавляем комплексный тип аргумента

5. Готовимся строить горы графиков по запросу преподавателя. Построение в gnuplot