Практика 2

1. Реализовать алгоритм вычисления логарифма. Исследовать возможность получения переполнения/исчезновения порядка в этом алгоритме в случае выполнения операций в неправильном порядке.
2. Реализовать алгоритм решения нелинейного уравнения методом деления отрезка пополам.
3. Реализовать алгоритм решения уравнения методом Ньютона. Соответствующая функция должна быть обобщенной, она должна позволять вычислять вещественный нуль функции одного переменного, комплексный корень функции одного комплексного переменного (например, многочлена). Также эта функция должна позволять находить корень системы из n нелинейных уравнений, задаваемых функциями n переменных.

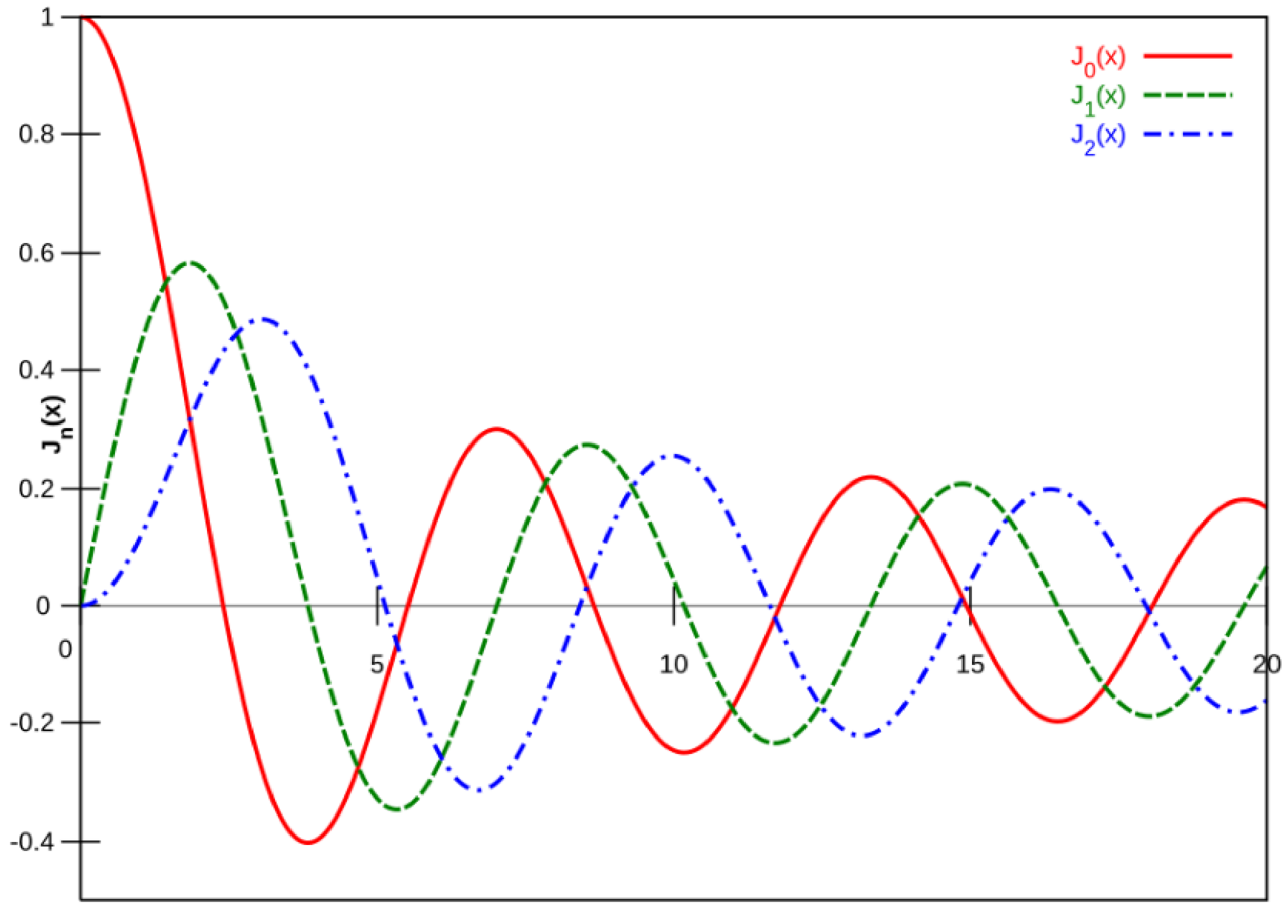
Протестировать этот алгоритм, выполнив следующие задания.

- Решить уравнение $\cos(x) = x$, выполнив дифференцирование вручную.
- Решить то же уравнение, заменив производную отношением конечных разностей при достаточно малом приращении аргумента (приращение аргумента следует брать порядка `1e-8` при использовании типа `Float64`).
- Найти корни уравнения $x^3 + x^2 + x + 1 = 0$, воспользовавшись алгоритмом, реализующим обобщенную схему Горнера, возвращающим кортеж из значения многочлена и его производной в заданной точке.
- Найти корни всех указанных выше уравнений, используя автоматическое дифференцирование на основе дуальных чисел (см. следующий пункт задания).
- Решить систему уравнений: $x^2 + y^2 = 2, y = x^3$, выполнив дифференцирование вручную.
- Решить ту же систему уравнений, используя автоматическое дифференцирование на основе дуальных чисел.

3. Реализовать параметрический тип `Dual{T}`, реализующий дуальные числа, с помощью которого можно реализовать автоматическое дифференцирование вычисляемых функций.
4. Написать алгоритм, реализующий вычисление функции Бесселя заданного (натурального) порядка https://ru.wikipedia.org/wiki/Функции_Бесселязаданного , определяемую своим рядом Тейлора:

$$J_\alpha(x) = \sum_{m=0}^{+\infty} \frac{(-1)^m}{m!(m+\alpha)!} \left(\frac{x}{2}\right)^{2m+\alpha}$$

где $\alpha = 0, 1, 2, 3, \ldots$ - порядок функции Бесселя. Графики этой функции при различных значениях параметра (порядка) $\alpha = 0, 1, 2$ выглядят так



Указания.

- Обеспечить, чтобы сложность алгоритма оценивалась, как O(n), где n - максимальная учтенная степень в разложении Тейлора (для этого необходимо получить и использовать соответствующие рекуррентные формулы).
- Реализовать функцию так, чтобы вычисления выполнялись с точностью "до машинного эпсилон".
- Реализовать построение графиков семейства функции Бесселя (для заданного набора значений порядка) .
- Реализовать построение графика функции Бесселя заданного порядка и его производной, воспользовавшись для вычисления производной автоматическим дифференцированием на основе дуальных чисел.
- Убедиться, что при достаточно боельших значениях аргумента точности Float64 не достаточно для обеспечения численной устойчивости.
- Для вычислений функции Бесселя для достаточно высоких значений аргумента воспользоваться стандартным (в Julia) типом `BigFloat`

Для построения графиков можно воспользоваться, например, пакетом `Plots.jl` (https://docs.juliaplots.org/stable/), или более мощным пакетом `Makie.jl` (https://docs.makie.org/v0.22/). Для Julia имеется также пакет PyPlot.jl (это обертка пакета `matplotlib` для Python).

6. Построить график фрактала Кэлли
   Постановка задачи.
   Рассматривается уравнение вида:

$$z^3 = 1$$

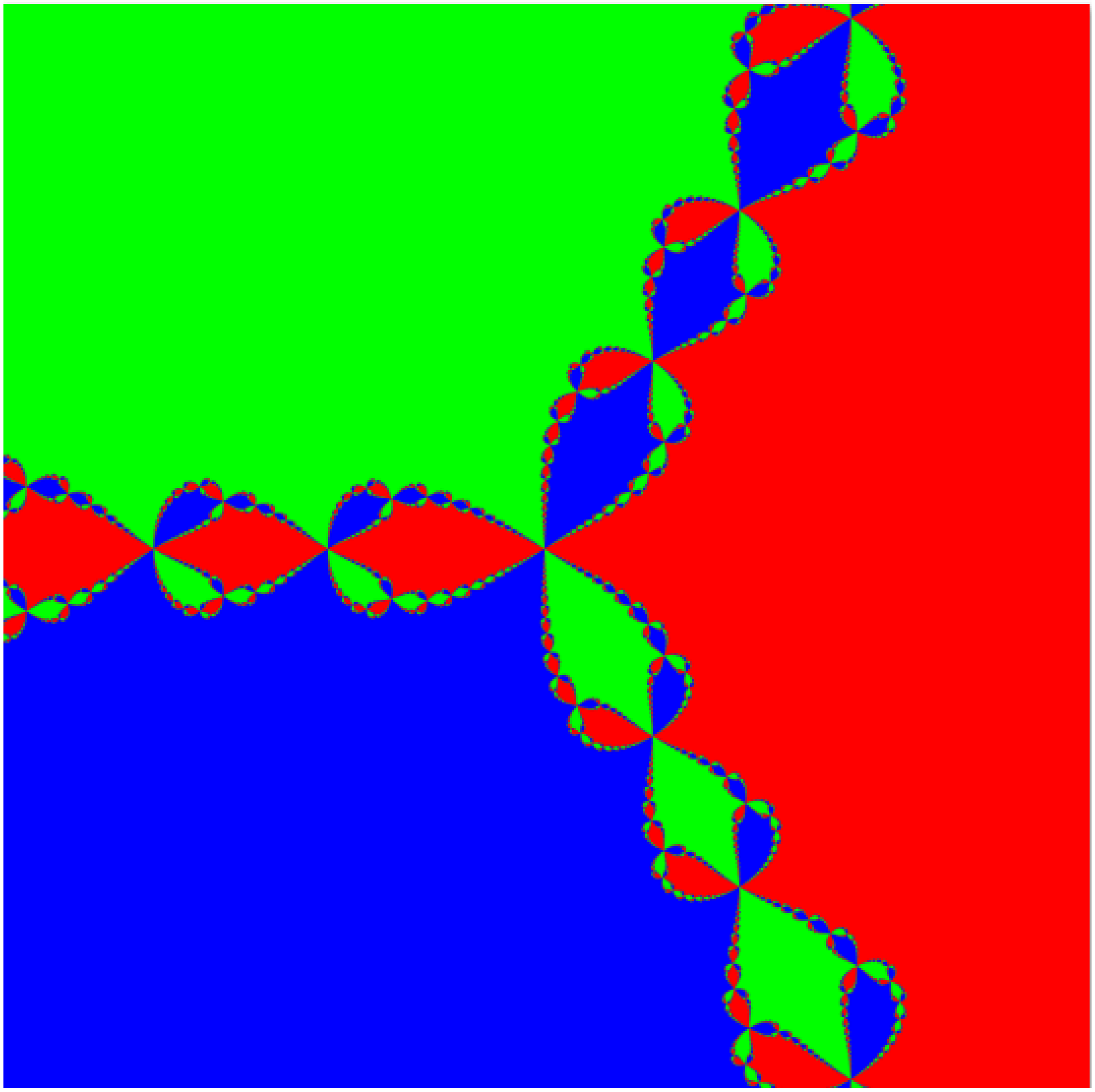
   Как известно, это уравнение имеет 3 комплексных корня, расположенных эквидистантно на единичной окружности: $z_{1,2,3} = (e^{i2\pi k/3}, k \in 0, 1, 2)$.

Эти корни можно пытаться вычислять решая уравнение $f(z) = z^3 - 1 = 0$ методом Ньютона при выбранном начальном значении z_0.

Таким образом, вся комплексная плоскость может быть разделена на 3 области - области "притяжения" к одному из 3-х комплексных корней уравнения. Мы будем говорить, что точка принадлежит области притяжения корня z_k $(k = 0, 1, 2,)$, если итерации метода Ньютона дают последовательность, сходящуюся к этому корню. На самом деле на комплексной плоскости имеется еще множество точек не принадлежащих ни одной из 3-х областей притяжения - это точки границ между этими областями.

Задача состоит в том, чтобы раскрасить точки комплексной плоскости в 3 цвета (каждый цвет соответствует одной из 3-х областей притяжения). Оставшиеся не закрашенные точки будут граничными точками.

Оказывается, что границы между этими 3-мя областями притяжения устроены очень сложно. Это будут не прямые лучи, выходящие из начала координат, как можно было бы предположить, это будут фрактальные множества. Фрактал Кэлли выглядит следующим образом.



Для графического отображения массива точек нужно использовать функцию image из пакета Makie.jl (функции `plot` или `scatter` для этой цели не подойдут).