

Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский Государственный Университет имени М.В.Ломоносова»
Физический факультет

Отчёт по практическому заданию №1

Быстрое дискретное комплексное преобразование Фурье

Студен группы №437: Белашов Егор Юрьевич

Оглавление

Постановка задачи	3
Используемые пакеты.....	3
Листинг программы	3
Таблицы значений:.....	5
Сигнал	5
Спектр с нумерацией из текста задания	5
Спектр с исправленной нумерацией (из numru)	5
Графики.....	6

Постановка задачи

Необходимо используя библиотечные функции написать программу, способную выполнить быстрое дискретное комплексное преобразование Фурье. Исходные данные необходимо было выгрузить из файла. Впоследствии необходимо было провести анализ полученных результатов.

Используемые пакеты

Программа написана на языке **python** с использованием следующих библиотек:

- **numpy**
Библиотека реализующая множество тензорных вычислений. В пакете присутствуют функции способные производить быстрое прямое и обратное комплексное преобразование Фурье
- **matplotlib**
Библиотека для построения и отображения графиков
- **belashovplot**
Небольшая надстройка над matplotlib, позволяющая строить «тайловые» графики, не беспокоясь о поехавшем тексте и форматировании.

Листинг программы

Программа реализована в виде одной единственной функции, которая принимает на вход путь к файлу с данными. Далее она выгружает данные, обрабатывает их, выводит таблицы в консоль, а в конце рисует графики и выводит их на экран.

```
def discrete_fourier_transformation_test(file_name:str= '58.txt'):  
    # Импорт данных из файла  
    data = []  
    with open(file_name, 'r') as file:  
        for line in file:  
            try:  
                data.append(complex(line))  
            except Exception as error:  
                print(f'Ошибка при сериализации строки {line[:-1]} в complex <{error}>.')  
  
    # Непосредственно вычисления  
    signal = numpy.array(data)  
    spectrum = numpy.fft.fftshift(numpy.fft.fft(signal))  
    signal_recalculated = numpy.fft.ifft(numpy.fft.ifftshift(spectrum))  
    signal_deviations = numpy.abs(signal - signal_recalculated)  
  
    # Вычисление рзмера данных  
    N = len(signal)  
  
    # Вывод таблиц результатов  
    for name, (data, data_range) in {  
        'Сигнал': (signal, numpy.arange(0, N)),  
        'Спектр': (spectrum, numpy.linspace(-N / 2 + 1, N / 2, N)),  
        'Спектр (исправленная нумерация)': (spectrum, numpy.fft.fftshift(numpy.fft.fftfreq(N,  
1)))  
    }.items():  
        print("Сигнал: ")  
        for n, value in zip(data_range, data):  
            print(f'{n}\t{value.real}\t{value.imag}')  
        print('')  
  
    # Инициализация параметров построения "тайлового" графика для сокращения количества кода  
    graphs_data = {  
        "Сигнал" : signal,  
        "Спектр [-N/2+1, N/2]" : spectrum,  
        "Спектр [-N/2-1, N/2]" : spectrum,  
        "Восстановленный сигнал" : signal_recalculated,  
        "Отклонения сигнала" : signal_deviations  
    }  
    graph_ranges = [  
        numpy.arange(0, N),  
        numpy.linspace(-N / 2 + 1, N / 2, N),  
        numpy.fft.fftshift(numpy.fft.fftfreq(N, 1)),
```

```

        numpy.arange(0, N),
        numpy.arange(0, N)
    ]
    extract_functions = {
        'g': lambda x: numpy.abs(x),
        'y': lambda x: numpy.angle(x),
        'r': lambda x: numpy.real(x),
        'b': lambda x: numpy.imag(x)
    }
    graph_types = [
        "амплитуда",
        "фаза",
        "реальная часть",
        "мнимая часть"
    ]

    # Настройка аннотаций пространства графиков
    plot = TiledPlot(MaxWidth=12*(21/9), MaxHeight=12)
    plot.FontLibrary.MultiplyFontSize(0.7)
    plot.title('Сравнение')
    plot.description.top("Спектр, полученный быстрым дискретным преобразованием Фурье исходного
сигнала и сравнение исходного сигнала с восстановленным из спектра")
    plot.description.row.left    ("Амплитуда",          0)
    plot.description.row.left    ("Фаза",                1)
    plot.description.row.left    ("Реальная часть",      2)
    plot.description.row.left    ("Мнимая часть",        3)
    plot.description.column.top  ("Сигнал",              0)
    plot.description.column.top  ("Спектр",              1)
    plot.description.column.top  ("Спектр (нумерация)",  2)
    plot.description.column.top  ("Восстановленный",     3)
    plot.description.column.top  ("Отклонения",         4)

    # Построение и добавление описаний
    for col, ((tittle, data), graph_range) in enumerate(zip(graphs_data.items(), graph_ranges)):
        for row, ((color, function), graph_type) in enumerate(zip(extract_functions.items(),
graph_types)):
            axes = plot.axes.add(col, row)
            axes.plot(graph_range, function(data), f'--{color}')
            axes.grid(True)
            plot.graph.description(f'{tittle} : {graph_type}')

    # Вывод графиков на экран
    plot.show()

```

Таблицы значений:

Сигнал

Номер	Координата	Вещ. часть	Мним. часть
1	0	0.211313	0.0
2	1	0.90473	0.0
3	2	1.051812	0.0
4	3	0.784871	0.0
5	4	-0.214828	0.0
6	5	-1.136924	0.0
7	6	-1.123443	0.0
8	7	-0.690541	0.0
9	8	0.220547	0.0
10	9	1.022713	0.0
11	10	1.169522	0.0
12	11	0.688723	0.0
13	12	-0.213614	0.0
14	13	-1.136187	0.0
15	14	-0.98588	0.0
16	15	-0.664737	0.0

Спектр с нумерацией из текста задания

Номер	Координата	Вещ. часть	Мним. часть
1	-7.0	0.3427810000000001	0.0
2	-6.0	0.052890198009562725	0.2148426367407888
3	-5.0	-0.10962364383152623	0.6398977924490294
4	-4.0	0.1015132536924411	0.009937194047247688
5	-3.0	-0.10859299999999983	-0.46398399999999995
6	-2.0	-0.14805763554623416	-0.3486453446604194
7	-1.0	1.8302276438315261	9.301211792449028
8	0.0	-0.043281816155769634	-0.14859590196687827
9	1.0	-0.11192300000000033	0.0
10	2.0	-0.043281816155769634	0.14859590196687827
11	3.0	1.8302276438315261	-9.301211792449028
12	4.0	-0.14805763554623416	0.3486453446604194
13	5.0	-0.10859299999999983	0.46398399999999995
14	6.0	0.1015132536924411	-0.009937194047247688
15	7.0	-0.10962364383152623	-0.6398977924490294
16	8.0	0.052890198009562725	-0.2148426367407888

Спектр с исправленной нумерацией (из numpy)

Номер	Координата	Вещ. часть	Мним. часть
1	-0.5	0.3427810000000001	0.0
2	-0.4375	0.052890198009562725	0.2148426367407888
3	-0.375	-0.10962364383152623	0.6398977924490294
4	-0.3125	0.1015132536924411	0.009937194047247688
5	-0.25	-0.10859299999999983	-0.46398399999999995
6	-0.1875	-0.14805763554623416	-0.3486453446604194
7	-0.125	1.8302276438315261	9.301211792449028
8	-0.0625	-0.043281816155769634	-0.14859590196687827
9	0.0	-0.11192300000000033	0.0
10	0.0625	-0.043281816155769634	0.14859590196687827
11	0.125	1.8302276438315261	-9.301211792449028
12	0.1875	-0.14805763554623416	0.3486453446604194
13	0.25	-0.10859299999999983	0.46398399999999995
14	0.3125	0.1015132536924411	-0.009937194047247688
15	0.375	-0.10962364383152623	-0.6398977924490294
16	0.4375	0.052890198009562725	-0.2148426367407888

Сравнение

Спектр, полученный быстрым дискретным преобразованием Фурье исходного сигнала и сравнение исходного сигнала с восстановленным из спектра

