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

Отчёт по практическому заданию №1

Быстрое

Студен группы №437: Белашов Егор Юрьевич

Оглавление

[Постановка задачи 3](#_Toc160570510)

[Используемые пакеты 3](#_Toc160570511)

[Листинг программы 3](#_Toc160570512)

[Таблицы значений: 5](#_Toc160570513)

[Сигнал 5](#_Toc160570514)

[Спектр с нумерацией из текста задания 5](#_Toc160570515)

[Спектр с исправленной нумерацией (из numpy) 5](#_Toc160570516)

[Графики 6](#_Toc160570517)

# Постановка задачи

Необходимо использую библиотечные функции написать программу, способную выполнить быстрое дискретное комплексное преобразование Фурье. Исходные данные необходимо было выгрузить из файла. В последствие 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 серилизации 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 | -0.0069951875000000205 | 0.0 |
| 2 | -6.0 | -0.002705113509735602 | 0.009287243872929892 |
| 3 | -5.0 | 0.11438922773947038 | -0.5813257370280642 |
| 4 | -4.0 | -0.009253602221639635 | 0.021790334041276213 |
| 5 | -3.0 | -0.006787062499999989 | 0.028998999999999997 |
| 6 | -2.0 | 0.006344578355777569 | -0.0006210746279529805 |
| 7 | -1.0 | -0.006851477739470389 | -0.03999361202806434 |
| 8 | 0.0 | 0.0033056373755976703 | -0.0134276647962993 |
| 9 | 1.0 | 0.021423812500000007 | 0.0 |
| 10 | 2.0 | 0.0033056373755976703 | 0.0134276647962993 |
| 11 | 3.0 | -0.006851477739470389 | 0.03999361202806434 |
| 12 | 4.0 | 0.006344578355777569 | 0.0006210746279529805 |
| 13 | 5.0 | -0.006787062499999989 | -0.028998999999999997 |
| 14 | 6.0 | -0.009253602221639635 | -0.021790334041276213 |
| 15 | 7.0 | 0.11438922773947038 | 0.5813257370280642 |
| 16 | 8.0 | -0.002705113509735602 | -0.009287243872929892 |

## Спектр с исправленной нумерацией (из numpy)

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Координата | Вещ. часть | Мним. часть |
| 1 | 0.0 | -0.0069951875000000205 | 0.0 |
| 2 | 1.0 | -0.002705113509735602 | 0.009287243872929892 |
| 3 | 2.0 | 0.11438922773947038 | -0.5813257370280642 |
| 4 | 3.0 | -0.009253602221639635 | 0.021790334041276213 |
| 5 | 4.0 | -0.006787062499999989 | 0.028998999999999997 |
| 6 | 5.0 | 0.006344578355777569 | -0.0006210746279529805 |
| 7 | 6.0 | -0.006851477739470389 | -0.03999361202806434 |
| 8 | 7.0 | 0.0033056373755976703 | -0.0134276647962993 |
| 9 | -8.0 | 0.021423812500000007 | 0.0 |
| 10 | -7.0 | 0.0033056373755976703 | 0.0134276647962993 |
| 11 | -6.0 | -0.006851477739470389 | 0.03999361202806434 |
| 12 | -5.0 | 0.006344578355777569 | 0.0006210746279529805 |
| 13 | -4.0 | -0.006787062499999989 | -0.028998999999999997 |
| 14 | -3.0 | -0.009253602221639635 | -0.021790334041276213 |
| 15 | -2.0 | 0.11438922773947038 | 0.5813257370280642 |
| 16 | -1.0 | -0.002705113509735602 | -0.009287243872929892 |

# Графики





