# ThinkDSP. Лабораторная 7. Дискретное преобразование Фурье.

Шерепа Никита

13 мая 2021 г.

# Содержание

# Список иллюстраций

# Листинги

# 1 Упражнение 7.2

1. Задание

   Дан массив сигнала `y`. Разделите его на четные элементы `e` и нечетные элементы `o`.

   Вычислите ДПФ `e` и `o`, делая рекурсивные вызовы

   Вычислите ДПФ(`y`) для каждого значения `n`, используя лемму Дэниелсона-Ланцоша.

2. Ход работы

   Рассмотрим сигнал и вычислим его БПФ

```
ys = [-0.5, 0.1, 0.7, -0.1]
hs = np.fft.fft(ys)
print(hs)

Output
[ 0.2+0.j -1.2-0.2j 0.2+0.j -1.2+0.2j]
```

Листинг 1: БПФ сигнала

   Теперь реализуем ДПФ

```
def dft(ys):
N = len(ys)
ts = np.arange(N) / N
freqs = np.arange(N)
args = np.outer(ts, freqs)
M = np.exp(1j * PI2 * args)
amps = M.conj().transpose().dot(ys)
return amps
```

Листинг 2: ДПФ

   Применим ДПФ к нашему сигналу

```
hs2 = dft(ys)
np.sum(np.abs(hs - hs2))

Output
5.864775846765962e-16
```

Листинг 3: Применение ДПФ

Видим, что ДПФ даёт почти такой же результат. Разница в результатах = 5.864775846765962e-16

Прежде чем перейти к созданию рекурсивного БПФ, рассмотрим версию, которая использует `np.fft.fft` для вычисления БПФ половин

```
def fft_norec(ys):
N = len(ys)
He = np.fft.fft(ys[::2])
Ho = np.fft.fft(ys[1::2])

ns = np.arange(N)
W = np.exp(-1j * PI2 * ns / N)

return np.tile(He, 2) + W * np.tile(Ho, 2)
```

Листинг 4: Вычисление БПФ половин

Применим её к сигналу

```
hs3 = fft_norec(ys)
np.sum(np.abs(hs - hs3))

Output
0.0
```

Листинг 5: Применение к сигналу

Видим, что результат такой же. Разница = 0.0

Теперь заменим `np.fft.fft` на рекурсивные вызовы

```
def fft(ys):
N = len(ys)
if N == 1:
return ys

He = fft(ys[::2])
Ho = fft(ys[1::2])

ns = np.arange(N)
W = np.exp(-1j * PI2 * ns / N)

return np.tile(He, 2) + W * np.tile(Ho, 2)
```

Листинг 6: Добавление рекурсивных вызовов

И применим результат к сигналу

```
hs4 = fft(ys)
np.sum(np.abs(hs - hs4))

Output
1.6653345369377348e-16
```

Листинг 7: Применение к сигналу

Разница в результатах = 1.6653345369377348e-16

Данная реализация имеет сложность O(nlogn) по времени и по памяти.

# 2 Вывод

Во результате выполнения работы получены навыки работы с Дискретным Преобразованием Фурье (ДПФ) и Быстрым Преобразованием Фурье (БПФ).