

# Прикладной вейвлет-анализ

## Лабораторная работа №1

### Быстрое преобразование Фурье

#### Постановка задачи

Написать программу, которая реализует функцию быстрого преобразования Фурье `fft`, а также обратную к ней `ifft`, указанным в варианте способом. Язык программирования может быть любым, но должен иметь библиотеку или встроенные функции для вычисления быстрого преобразования Фурье (эту функцию обозначим `truefft` и будем использовать для проверки результатов).

Провести вычислительные эксперименты по следующей схеме. Для  $n = 2^k, k = 1, 2, \dots, 16$  сгенерировать случайный вектор  $x^n \in \mathbb{C}^n$  и вычислить для него векторы  $y^n = \text{fft}(x^n)$ ,  $\tilde{x}^n = \text{ifft}(y^n)$  и  $z^n = \text{truefft}(x^n)$ . Вычислить  $\epsilon^n = \|x^n - \tilde{x}^n\|$  и  $\delta^n = \|y^n - z^n\|$ . Время вычисления вектора  $y^n$  обозначим  $t_y^n$ , время вычисления вектора  $z^n$  обозначим  $t_z^n$ .

#### Требования к содержанию отчета

- 1) Векторы  $x^8, \tilde{x}^8, y^8, z^8, \epsilon^8, \delta^8$ .
- 2) Точечные графики  $t_y^n$  и  $t_z^n$  (на одной координатной плоскости).
- 3) Таблицу, каждая строка которой содержит  $k, \epsilon^n, \delta^n, t_y^n, t_z^n$ .
- 4) Ваши комментарии и выводы.
- 5) Исходный код программы.

Отчет о работе печатается на принтере. Титульный лист не нужен! В заголовке работы указывается номер и тема лабораторной работы, имя и фамилия автора, а также скриншот требований к содержанию из настоящего документа. Листы отчета должны быть скреплены!

Вместе с отчетом на защиту лабораторной работы приносится ноутбук с разработанной программой, которую нужно будет выполнить.

## Варианты

### Вариант 1

- 1) Предварительное задание: сгенерировать в явном виде матрицу  $F_8$  и ее факторизацию Кули-Тьюки (вывести на печать и занести в отчет).

Проверить тождество  $F_8 = A_3 A_2 A_1 P_8^T$ .

2) Выполнить основное задание с использованием *рекурсивного* алгоритма БПФ.

## **Вариант 2**

Реализовать алгоритм БПФ на основе факторизации Кули-Тьюки. Перестановку  $P_n^T$  рекомендуется вычислять с помощью бит-реверсии.