МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ПАО ГИС

ОТЧЁТ

по практической работе

по дисциплине «Архитектура ПО ГАС»

Тема: Реализация алгоритма БПФ с помощью библиотеки fftw

Студентка гр. 5381	________________	Петрова Т.А.
Преподаватель	________________	Пуеров Г.Ю.

Санкт-Петербург
2020

Цель работы.

Целью работы является изучение механизма обработки сигнала с помощью быстрого преобразования Фурье (БПФ) и его программной реализации. Также решение задач по ДПФ.

Постановка задачи.

Требуется выполнить обработку wav файла, куда записан тональный сигнал, для определения частоты. Преобразование строиться с помощью библиотеки fftw на языке Си в операционной системе Linux. Также необходимо измерить производительность функции с разными планами на личных рабочих машинах.

Сигнал записан в файл ton_signal_4.wav.

Выполнение работы.

В ходе работы программы выполняется чтение данных из звукового файла. Реализованы две функции: для считывания структуры заголовки файла и функция для чтения из файла порции данных указанного размера. На основе полученных из заголовка данных определяется общее количество отсчетов, количество целых порций по N и остаточное количество дополняется до N нулями. Для каждой порции значений, в цикле, выполняется БПФ, с использованием функций библиотеки fftw. Затем в файл записывается накопление спектра, которое считается в функции по формуле

$$acc = \left(1 - \frac{1}{c}\right) acc + \left(\frac{1}{c}\right) Sq,$$

где $c = 4$ —некоторый коэффициент накопления, Sq —массив квадратов модулей на текущей итерации цикла. Далее полученные данные программой на языке Python выводятся в виде графика.

Реализация БПФ с помощью библиотеки fftw.

```
fftw_plan plan;
plan = fftw_plan_dft_r2c_1d(N, data_portion, out,  FFTW_ESTIMATE);
fftw_execute(plan);
fftw_destroy_plan(plan);
```

Для выполнения преобразования Фурье с помощью библиотеки необходимо сначала определить план, в который нужно записать параметры преобразования: размер БПФ, порцию обрабатываемых данных и определить флаг: FFTW_ESTIMATE или FFTW_MEASURE. Затем выполнить преобразование с заданным планом и очистить план как указано выше.

Результаты выполнения программы при разных размерах БПФ:

Ниже (см. рис.1) представлен график спектра сигнала при N = 4096. По графику видно, что частота сигнала около 1600-1700 Гц.

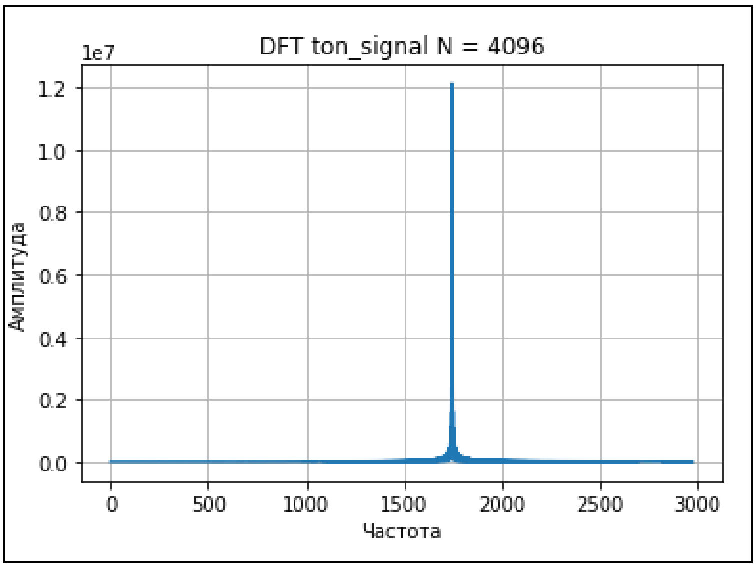

Рисунок 1 – График спектра сигнала при N = 4096

Далее представлен график спектра сигнала при N = 2048. На данном графике частота определяется также – 1600-1700 Гц.

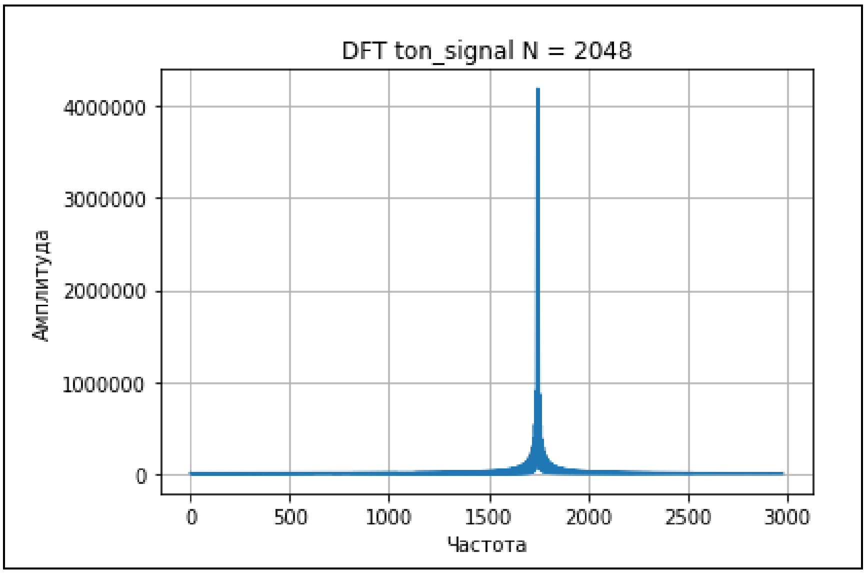

Рисунок 2 – График спектра сигнала при N=2048

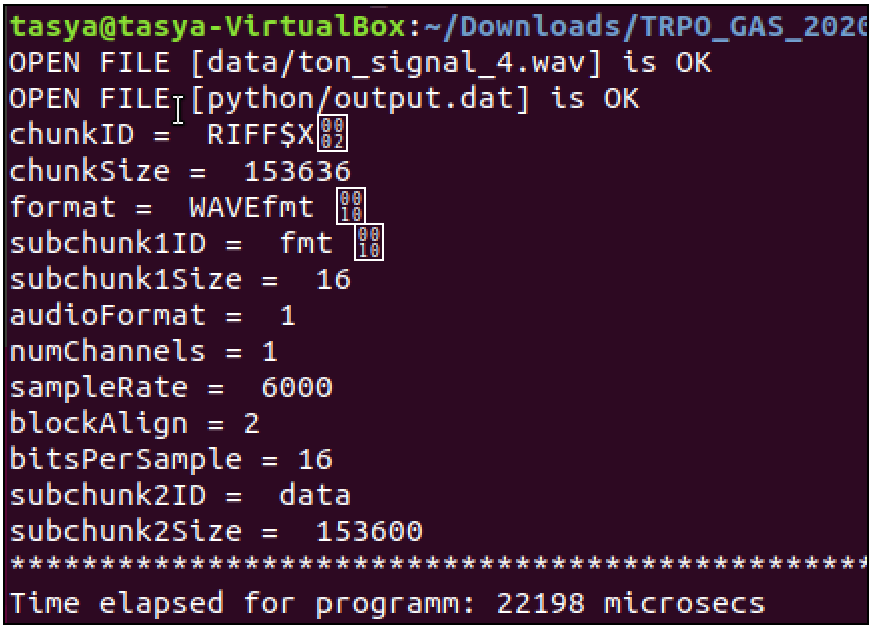

Рисунок 3 – вывод программы.

Для замера производительности функции fft был использован таймер. Для разных планов были получены следующие результаты (см.табл.1).

Таблица 1.

К/Флаг	FFTW_ESTIMATE	FFTW_MEASURE
1	22198 мкс.	227982 мкс.
2	29734 мкс.	301457 мкс.
3	21457 мск.	214083 мкс.
4	26505 мск	230731 мкс.

Характеристики виртуальной машины: Ubuntu 18.04 64-bit, 1024 Мб основная память, 1 ЦП.

Полученные результаты показывают, что на данных характеристиках, время создания плана с использованием флага FFTW_MEASURE намного больше, чем FFTW_ESTIMATE.

Вывод

При выполнении данной работы, был получен навык построения БПФ на языке С. Были оценены временные характеристики различных планов.

Поскольку, FFTW_MEASURE дает команду FFTW запустить и измерить время выполнения нескольких FFT, чтобы найти лучший способ вычислить преобразование размера N. Этот процесс занимает некоторое время (обычно несколько секунд), в зависимости от вашей машины и размера преобразования. FFTW_ESTIMATE, напротив, не выполняет никаких вычислений, а просто строит разумный план, который, вероятно, не является оптимальным. То есть если программа выполняет много преобразований одного размера и время инициализации не важно, выгоднее использовать FFTW_MEASURE.

Задачи

Задача 1. Найдите ДПФ сигнала $x \in \mathbb{C}_N$,

$$x_j = \sin\frac{2\pi j}{N}\cos\frac{2\pi j}{N}, \qquad j \in 0\colon N-1$$

Воспользуемся формулой $X(k) = \sum_{j=0}^{N-1} x_j w_N^{-kj} = \sum_{j=0}^{N-1} x_j e^{-kj\frac{2\pi i}{N}}$:

$$X(k) = \sum_{j=0}^{N-1} x_j e^{-ik\frac{2\pi j}{N}} = \sum_{j=0}^{N-1} \sin\frac{2\pi j}{N}\cos\frac{2\pi j}{N} e^{-ik\frac{2\pi j}{N}} =$$

$$= \sum_{j=0}^{N-1}\left(\frac{e^{i\frac{2\pi j}{N}} - e^{-i\frac{2\pi j}{N}}}{2i}\right)\left(\frac{e^{i\frac{2\pi j}{N}} + e^{-i\frac{2\pi j}{N}}}{2}\right)e^{-ik\frac{2\pi j}{N}} =$$

$$= \sum_{j=0}^{N-1}\frac{1}{4i}\left(e^{2i\frac{2\pi j}{N}} - e^{-2i\frac{2\pi j}{N}}\right)e^{-ik\frac{2\pi j}{N}} = \frac{1}{4i}\sum_{j=0}^{N-1} e^{(2-k)i\frac{2\pi j}{N}} - e^{-(2+k)i\frac{2\pi j}{N}} =$$

$$= \left[w_N^j = e^{\frac{2\pi i}{N}j}\right] = \frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k+2)j}$$

При $k = 2$*:*

$$w_N^{-(k-2)} = 1$$

Следовательно,

$$\frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i}\left(N - \frac{1 - w_N^{-(k+2)N}}{1 - w_N^{-(k+2)}}\right) = \frac{N}{4i}$$

При $k = -2$*:*

$$w_N^{-(k+2)} = 1$$

Следовательно,

$$\frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i}\left(\frac{1 - w_N^{-(k-2)N}}{1 - w_N^{-(k-2)}} - N\right) = -\frac{N}{4i}$$

При остальных k:

$$\frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i}\sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i}\left(\frac{1 - w_N^{-(k-2)N}}{1 - w_N^{-(k-2)}} - \frac{1 - w_N^{-(k+2)N}}{1 - w_N^{-(k+2)}}\right) = 0$$

$$X(k) = \begin{cases} k = 2, & \frac{N}{4i} \\ k = N-2, & -\frac{N}{4i} \\ k \neq 2\,, & 0 \end{cases}$$

Задача 2. Найдите взаимосвязь между ДПФ исходного сигнала $x \in \mathbb{C}$ и ДПФ преобразованного сигнала $y_l \in \mathbb{C}$

$$y_l(j) = -2i\,Sin\left(\frac{2\pi l j}{N}\right)x(j), \quad j, l \in \mathbb{Z}$$

$$Y_l(k) = \sum_{j=0}^{N-1}\left(w_N^{-lj} - w_N^{lj}\right)x(j)w_N^{-kj} = \left(\sum_{j=0}^{N-1} x(j)w_N^{-(k+l)j} - \sum_{j=0}^{N-1} x(j)w_N^{-(k-l)j}\right)$$

$$= X(k+l) - X(k-l)$$

Задача 3. Найдите взаимосвязь между ДПФ исходного сигнала $x \in \mathbb{C}$ и ДПФ преобразованного сигнала $y \in \mathbb{C}$

$$y(j) = \overline{x(N-j)},\ \ j \in 0\colon N-1$$

$$Y(k) = \sum_{j=0}^{N-1} \overline{x(N-j)}\, w_N^{-kj}$$

$$\overline{Y(k)} = \overline{\sum_{j=0}^{N-1} \overline{x(N-j)}\, w_N^{-kj}} = \sum_{j=0}^{N-1} x(N-j)\, w_N^{kj} = [N-j=p]$$

$$= \sum_{p=1}^{N} x(p) w_n^{k(N-p)} = w_N^{kN} \sum_{p=1}^{N} x(p) w_n^{-pk} = X(k)$$

Следовательно, результат:

$$Y(k) = \overline{X(p)}$$

Задача 4. Используя один из алгоритмов Кули-Тьюки, найдите ДПФ исходного сигнала $x \in \mathbb{C}_{2M}$

$$x(j) = (-1)^j\,, \qquad j \in 0\colon 2M-1$$

Используем алгоритм Кули-Тьюки с прореживанием по времени по основанию 2. Что разделает множество компонент исходного вектора на два подмножества – четные и нечетные индексы.

$$X_k = \sum_{j=0}^{M-1} x_{2j} w_M^{kj} + w_{2M}^{k} \sum_{j=0}^{M-1} x_{2j+1}\, w_M^{kj} = \sum_{j=0}^{M-1} w_M^{kj} - w_{2M}^{k} \sum_{j=0}^{M-1} w_M^{kj} = 0$$

$$X_{k+\frac{n}{2}} = \sum_{j=0}^{M-1} x_{2j} w_M^{(k+M)j} - w_{2M}^{k+M} \sum_{j=0}^{M-1} x_{2j+1}\, w_M^{(k+M)j} =$$

$$= \sum_{j=0}^{M-1} w_M^{(k+M)j} + w_{2M}^{k+M} \sum_{j=0}^{M-1} w_M^{(k+M)j} = 2M\delta_M(k+M)$$

Задача 5. Пусть p – простое число, $N = p^n$. Докажите, что количество умножений при N-точечном БПФ алгоритме не превосходит $C(p)Nlog_pN$, где константа C(p) зависит от p, но не зависит от n. Например, можно взять $C(p) = p^2$.

Для n-точечного БПФ алгоритма количество умножений вычисляется по формуле:

$$M(n) = n(n_1 + n_2 + 1)$$

Поскольку $N = p^n$ БПФ приводится к p-точечному БПФ, можно записать формулу в виде:

$$p = p_1 * p_2 = p * 1$$

$$M(p) = p(p_1 + p_2 + 1) = p(p + 2)$$

Докажем по индукции.

База, n=2.

$$N = p^2 = C(p) = p^2$$

$$p(p + 2) \leq p^2 * p^2 * 2 \log_{\mathrm{p}} p$$

$$p + 2 \leq 2p^3$$

Верно, поскольку p – простое число.

Индуктивный переход. Предположим, что:

$$p(p + 2) \leq p^2 p^n \log_{\mathrm{p}} p^n$$

$$p + 2 \leq np^{n+1}$$

Тогда, для n=n+1:

$$p(p + 2) \leq (n + 1)p^{n+3}$$

$$p + 2 \leq np^{n+2} + p^{n+2}$$

Из индуктивного перехода верно, что:

$$p + 2 \leq np^{n+1}$$

$$np^{n+1} \leq np^{n+2} \leq np^{n+2} + p^{n+2}$$

Значит:

$$p + 2 \leq np^{n+2} + p^{n+2}$$

Следовательно, утверждение верно.