

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

ЗВІТ

про виконання лабораторних робіт
з дисципліни

«МАТЕМАТИЧНІ ОСНОВИ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ»

Виконала:
студентка групи ПМ-33
Вітюк Аліна Сергіївна

Прийняв:
канд. фіз.-мат. наук, доц.
Пабірівський Віктор
Володимирович

Лабораторна робота №2

Тема: дискретне перетворення Фур'є

Мета: розробити комп'ютерну програму для виконання дискретного перетворення Фур'є (ДПФ) та оберненого дискретного перетворення Фур'є (ОДПФ).

Варіант: 2

Постановка задачі:

Реалізувати на мові програмування: C++, C#, Python, JavaScript:

1. Підпрограму (процедуру чи функцію), що виконуватиме обчислення одного (k -го) члена ряду Фур'є для ДПФ у тригонометричній формі.
2. Підпрограму (процедуру чи функцію), що обчислюватиме коефіцієнт Фур'є $C_k = A_k + jB_k$, як суму ряду з N членів ($N = 10 + n$, де n – номер студента у журналі) ($k = 0, 1, 2, \dots, N - 1$).
3. При складанні підпрограм оцінювати такі показники:
 - а. Час обчислення.
 - б. Кількість операцій (множення, додавання).
4. Побудувати графік функцій спектру амплітуд та фаз для N .
5. Головну програму для обчислення ДПФ з довільною вхідною множиною значень сигналу. Згенерувати довільний вхідний вектор $(f_0, f_1, f_2, f_3, \dots, f_{N-1})$ ($N = 10 + n$, де n – номер студента у журналі) та обчислити п.1,2,4.

Теоретичні відомості:

Коефіцієнти Фур'є C_k дискретного перетворення Фур'є можна подати як:

$$C_k = A_k + jB_k, \text{ де } j = \sqrt{-1}$$

де

$$A_k = \frac{1}{N} \sum_{i=0}^{N-1} f_i \cos\left(\frac{2\pi}{N} ki\right);$$

$$B_k = -\frac{1}{N} \sum_{i=0}^{N-1} f_i \sin\left(\frac{2\pi}{N} ki\right);$$

$$C_k = A_k + jB_k = \frac{1}{N} \sum_{i=0}^{N-1} f_i \left(\cos\left(\frac{2\pi ki}{N}\right) - j \sin\left(\frac{2\pi ki}{N}\right) \right), \quad (k = 0, 1, 2, \dots, N-1) \quad (*)$$

Множину абсолютних величин коефіцієнтів $C_k \{k = 0, \pm 1, \pm 2, \dots\}$:

$$|C_k| = \sqrt{A_k^2 + B_k^2} \quad (**)$$

називають спектром амплітуд, який показує, наскільки велика складова кожної гармоніки всередині сигналу.

Сукупність аргументів C_k

$$\arg C_k = \arctg \frac{B_k}{A_k} \quad (***)$$

- називають спектром фаз.

Хід роботи:

Для реалізації поставлених завдань обрала мову програмування Matlab.

1. Функція для обчислення одного (k -го) члена ряду Фур'є для ДПФ у тригонометричній формі.

```
% Підпрограма для обчислення k-го члена ряду Фур'є
function [c] = fourier_coefficient_k(k, x)
N = length(x);
n = 0:N-1;
c = exp(-2i * pi * k * n / N); c = dot(x, c);
end
```

2. Функція для обчислення коефіцієнта Фур'є $C_k = A_k + jB_k$, як суму ряду з N членів ($N = 10 + 2$, де n – номер студента у журналі) ($k = 0, 1, 2, \dots, N-1$).

```
% Підпрограма для обчислення коефіцієнта Фур'є C_k
function Ck = fourier_coefficient(k, x)
Ak = fourier_coefficient_k(k, real(x));
Bk = fourier_coefficient_k(k, imag(x));
Ck = Ak + 1i * Bk;
End
```

3. Оцінка показників:

а. час обчислення.

```
disp(['Час обчислення: ', num2str(elapsed_time)]);
```

б. кількість операцій (множення, додавання).

```
% Підпрограма для обчислення кількості операцій
```

```
function [num_mult, num_add] = um_mult_add(k, x)
```

```
N = length(x);
```

```
num_mult = 4 * N;
```

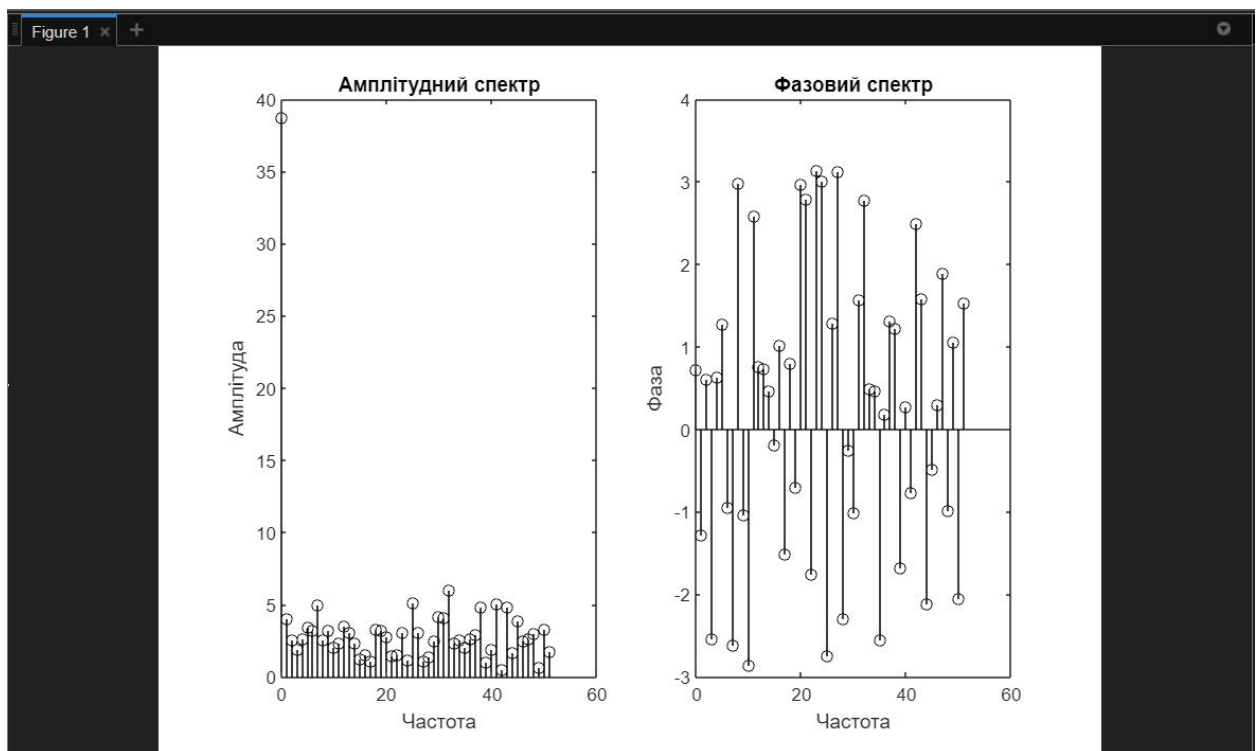
```
num_add = N - 1;
```

```
fprintf("Кількість операцій множення: %d\n", num_mult);
```

```
fprintf("Кількість операцій додавання: %d\n\n", num_add);
```

```
end
```

4. Побудова графіку функцій спектру амплітуд та фаз для N .



5. Головну програму для обчислення ДПФ з довільною вхідною множиною значень сигналу. Згенерування довільного вхідного вектора $(f_0, f_1, f_2, f_3, \dots, f_{N-1})$ ($N = 10 + n$, де n – номер студента у журналі) та обчислення п.1,2,4.

```
clear all
```

```
n = 2;
```

```

N = 50 + n;
% Генерація довільного вектора f
f = rand(N,1) + 1i * rand(N,1);
% Обчислення коефіцієнтів Фур'є та часу обчислення
tic;
for k = 0:N-1
C(k+1) = fourier_coefficient(k, f);
end
elapsed_time = toc;
% Обчислення спектру амплітуд та фаз та побудова графіків
amp_spectrum = abs(C);
phase_spectrum = angle(C);
freq_axis = 0:N-1;
subplot(1, 2, 1);
stem(freq_axis, amp_spectrum, 'black');
title('Амплітудний спектр');
xlabel('Частота');
ylabel('Амплітуда');
subplot(1, 2, 2);
stem(freq_axis, phase_spectrum, 'black');
title('Фазовий спектр');
xlabel('Частота');
ylabel('Фаза');
num=um_mult_add(k, f);
for k = 0:N-1
fprintf("C_%d = %.6f + %.6fi\n", k, real(C(k+1)), imag(C(k+1)));
end

```

```
Lab2.m | um_mult_add.m | fourier_coefficient.m | fourier_coefficient_k.m | Figure 1
Command Window
>> Lab2
Час обчислення: 0.023196
Кількість операцій множення: 208
Кількість операцій додавання: 51

C_0 = 29.261233 + 25.334428i
C_1 = 1.158858 + -3.828448i
C_2 = 2.101947 + 1.455650i
C_3 = -1.559446 + -1.066945i
C_4 = 2.134174 + 1.541206i
C_5 = 1.028968 + 3.301133i
C_6 = 1.874747 + -2.619978i
C_7 = -4.293206 + -2.492884i
C_8 = -2.489660 + 0.407002i
C_9 = 1.640549 + -2.785412i
C_10 = -1.990719 + -0.567264i
C_11 = -1.950862 + 1.215731i
C_12 = 2.505639 + 2.399597i
C_13 = 2.298265 + 2.055530i
C_14 = 2.081577 + 1.051798i
C_15 = 1.217039 + -0.242537i
C_16 = 0.806087 + 1.281676i
C_17 = 0.057343 + -1.087482i
C_18 = 2.284587 + 2.363673i
C_19 = 2.445844 + -2.080245i
C_20 = -2.715061 + 0.481349i
C_21 = -1.328230 + 0.482848i
C_22 = -0.287487 + -1.496540i
C_23 = -3.061700 + 0.012223i
C_24 = -1.147374 + 0.150859i
C_25 = -4.734222 + -1.992813i
C_26 = 0.875928 + 2.959969i
C_27 = -1.072562 + 0.027734i
```

Висновок: виконання лабораторної роботи №2 допомогло пригадати теоретичні відомості про дискретне перетворення Фур'є (ДПФ) та обернено дискретне перетворення Фур'є (ОДПФ). У ході виконання лабораторної роботи навчилася розробляти комп'ютерну програму мовою програмування Matlab для виконання ДПФ та ОДПФ.

Посилання на GitHub: <https://github.com/AlinaVitiuk/Mathematical-foundations-of-digital-signal-processing>