

## ЛАБОРАТОРНА РОБОТА № 3

### Обчислення

#### дискретного перетворення Фур'є (ДПФ, DFT)

Завдання: Реалізувати пряме та обернене дискретне перетворення Фур'є та на мові програмування C++.

Нижче наведена програма мовою C призначена для обчислення прямого ДПФ або оберненого дискретного перетворення Фур'є ОДПФ дискретної в часі послідовності  $x(n)$ :

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}, k = 0, 1, \dots, N-1 \quad \text{ДПФ},$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W^{-nk} \quad \text{ОДПФ},$$

де  $W = e^{-\frac{2\pi i}{N}}$ , а  $n$  – довжина послідовності.

Вхідна послідовність  $x(n)$  повинна представлятися в комплексній формі (дійсна і уявна частини). Для послідовності дійсних чисел уявні частини покладаються рівними нулю. Основна функція `dftd.c` (програма 1) приведена в нижче, а функція, яка обчислює ДПФ або ОДПФ, – у програмі 2. Для читання вхідної послідовності даних і збереження перетворених даних потрібні дві функції: `read_data()` і `save_data()` (програма 3). Початкові дані зберігаються у файлі `coeff.dat`, а результат записується в файл `dftout.dat`.

#### Програма 1. Основна функція `dftd.c` для обчислення ДПФ

```
/*-----*/
/* Програма для прямого обчислення коефіцієнтів ДПФ*/
/* вона використовує 3 інші функції */
/* -----*/
#include "dspl.h"
#include "dft.h"
main()
{
    extern long npt;
    extern int inv;
    printf("виберіть тип перетворення \n");
    printf("n");
    printf("0 для прямого ДПФ\n");
    printf("1 для оберненого ДПФ\n");
    scanf("%d", &inv);
    read_data();
    dft();
    save_data ();
    exit();
}
#include "dft.c";
#include "rdata.c";
#include "sdata.c";
```

Програма 2. Функція на мові С для прямого обчислення ДПФ дискретної в часі послідовності. Функція записана в окремому файлі

```
/**/  
/* Функція для обчислення прямого ДПФ */  
/* дискретної в часі послідовності */  
/* */  
void dft()  
{  
    extern int inv;  
    extern long npt;  
    long k, n;  
    double WN, wk, cf s, XR[size], XI[size];  
    extern complex x[size];  
    WN=2*pi/npt;  
    if(inv==1)  
        WN=-WN;  
    for(k=0;k<npt;++k){  
        XR[k]=0.0; XI[k]=0.0;  
        wk=k*WN;  
        for(n=0;n<npt;++n){  
            c=cos(n*wk); s=sin(n*wk);  
            XR[k]=XR[k]+x[n+1].real*c+x[n+1].imag*s;  
            XI[k]=XI[k]-x[n+1].real*s+x[n+1].imag*c;  
        }  
        if(inv==1){/*розділити на N для ОДПФ*/  
            XR[k]=XR[k]/npt;  
            XI[k]=XI[k]/npt;  
        }  
    }  
    for(k=1;k<=npt;++k){/*зберегти перетворені дані в x*/  
        x[k].real=XR[k];  
        x[k].imag=XI[k];  
    }  
}
```

Програма 3. Функція для читання даних, функція для запису перетворених даних у файл на диску, заголовний файл, який містить постійні структурні визначення, і заголовний файл, який містить загальні оголошення та змінні

```
/*-----*/  
/*Функція для читання даних в комплексному форматі */  
/*      для ДПФ або ШПФ      */  
/*-----*/  
void read_data()  
{  
    extern long npt;  
    int n;  
    extern complex x[size];  
    for(n=0; n<size; ++n) {  
        x[n].real=0;  
        x[n].imag=0;  
    }  
    if ((in=fopen ("coeff.dat", "r")) ==NULL) {  
        printf ("неможливо відкрити файл coeff.dat\n");  
        exit (1);  
    }  
}
```

```

        fscanf (in, "%dI ", &npt);
        for (n=1; n<= npt; ++n) {
            fscanf (in,%If% If", &x[n].real,&x[n].imag);
        }
        fclose(in);
    }
void save_data() /*ім'я файлу sdata.c*/
{
    long k;
    int k1;
    extern long npt;
    extern complex x[size];
    if ((out=fopen ("dftout.dat", "w")) ==NULL) {
        printf ("неможливо відкрити файл dftout.dat\n");
        exit (1);
    }
    fprintf {out f "k\tXR(k)\t\tXI(k)\n");
    fprintf (out, "\n");
    for(k=1; k<= npt; ++k) {
        k1=k-1;
        fprintf (out,"%d,\t%f\t%f\n",k1,x[k].real,x[k].imag);
    }
    fclose (out);
}

```

```

/*Цей файл містить загальні оголошення та структури.*/
#include <stdio.h>
#include <math.h>
#include <dos.h>
#define size 600
#define pi 3.141592654
#define maxbits 30
typedef struct {
    double real;
    double imag;
    double modulus;
    double angle;
}complex;
/*ім'я файлу: dft.h*/
void dft();
void fft();
void real__data();
void save__data();
int inv;
long npt;
complex x[size];
FILE *in, *out, *fopen();

```

Задача 1.

Скористайтесь програмою прямого обчислення ДПФ і знайдіть коефіцієнти наступної восьмиточкової дискретної в часі послідовності:

$$x(n)=\{4,2,1,4,6,3,5,2\}$$