|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
| NSTU | | |
| Кафедра вычислительных технологий | | |
|  | | |
| Лабораторная работа № 4 | | |
| по дисциплине «Численные методы» | | |
|  | | |
| **Дискретное преобразование Фурье** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-81 |
| Бригада: | 4 |
| Студенты: | Кайда Даниил |
|  | Парышков Дмитрий |
|  |  |
| Преподаватели: | Иткина Наталья Борисовна |
|  | Марков Сергей Игоревич |
|  | | |
| Новосибирск | | |
| 2020 | | |

1. **Цель работы**

Изучить и реализовать алгоритмы дискретного преобразования Фурье.

1. **Ход работы**

**Задание 1**

Реализовали алгоритмы прямого и обратного преобразования Фурье, а также быстрые алгоритмы прямого и обратного преобразования Фурье с использованием шаблонного класса complex<double>. Реализовали алгоритм построения отсчётов сигнала по формуле ниже, в который на вход подаются значения данной формулы:

.

Так же использована подпрограмма из ЛБР№3 для вывода числа в экспоненциальном виде.

Все сравнение двух чисел проводилось с точностью Eps = 0,001 (Функция sign, при вычисление значения и вывод данных 3-го подпункта всех заданий)

Их реализация:

void SignalCount(vector<complex<double>>& Z, vector<complex<double>>& DFT\_Data, vector<complex<double>>& IDFT\_Data,

int N, double A\_0, double A, double w, double fi, int key = 1)

{

Z.clear();

Z.resize(N);

DFT\_Data.clear();

DFT\_Data.resize(N);

IDFT\_Data.resize(N);

IDFT\_Data.clear();

if (key == 1) // Ключ 1 используется для всех заданий, кроме 6 (по умолчанию)

for (int j = 0; j < N; j++)

{

Z[j].\_Val[0] = A\_0 + A \* cos(2 \* PI \* j \* w / N + fi);

Z[j].\_Val[1] = 0;

}

else

for (int j = 0; j < N; j++)

{

Z[j].\_Val[0] = A\_0 + A \* cos(2 \* PI \* j / N + fi) + 0.01 \* A \* cos(2 \* PI \* j \* w / N + fi);

Z[j].\_Val[1] = 0;

}

}

void FFT(const vector<complex<double>>& Data, vector<complex<double>>& Result)

{

int N = Data.size(), M = N / 2;

Result.clear(); Result.resize(N);

complex<double> Exp, U, V;

for (int m = 0; m < M; m++)

{

U.\_Val[0] = 0.0; U.\_Val[1] = 0.0;

V.\_Val[0] = 0.0; V.\_Val[1] = 0.0;

for (int n = 0; n < M; n++)

{

Exp.\_Val[0] = cos(-2.0 \* PI \* m \* n / M);

Exp.\_Val[1] = sin(-2.0 \* PI \* m \* n / M);

U += Data[2 \* n] \* Exp;

V += Data[2 \* n + 1] \* Exp;

}

Exp.\_Val[0] = cos(-2.0 \* PI \* m / N);

Exp.\_Val[1] = sin(-2.0 \* PI \* m / N);

Result[m] = U + Exp \* V;

Result[m + M] = U - Exp \* V;

}

} // БПФ

void IFFT(const vector<complex<double>>& Data, vector<complex<double>>& Result)

{

int N = Data.size();

Result.clear(); Result.resize(N);

FFT(Data, Result);

complex<double> Val;

for (int i = 1; i <= N / 2; i++)

{

Val = Result[i];

Result[i] = Result[N - i] / double(N);

Result[N - i] = Val / double(N);

}

Result[0] /= double(N);

}//БОПФ

void DFT(const vector<complex<double>>& Data, vector<complex<double>>& Result) // ПФ

{

int N = Data.size();

Result.clear(); Result.resize(N);

for (int i = 0; i < N; i++)

{

Result[i].\_Val[0] = 0.0;

Result[i].\_Val[1] = 0.0;

for (int j = 0; j < N; j++)

{

complex<double> w(cos(-2.0 \* PI \* i \* j / N), sin(-2.0 \* PI \* i \* j / N));

Result[i] += Data[j] \* w;

}

}

}

void IDFT(const vector<complex<double>>& Data, vector<complex<double>>& Result) // ОПФ

{

int N = Data.size();

Result.clear(); Result.resize(N);

for (int i = 0; i < N; i++)

{

Result[i].\_Val[0] = 0.0;

Result[i].\_Val[1] = 0.0;

for (int j = 0; j < N; j++)

{

complex<double> w(cos(-2.0 \* PI \* i \* j / N), -sin(-2.0 \* PI \* i \* j / N));

Result[i] += Data[j] \* w;

Result[i] /= double(N);

}

}

}

**Задание 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| N |  | Время выполнения | |
| FFT | DFT |
| 256 | 1 | 0.0178477 | 0.0583128 |
| 127 | 0.0136259 | 0.0466012 |
| 255 | 0.0176526 | 0.0736454 |
| 512 | 1 | 0.077166 | 0.17655 |
| 255 | 0.0616245 | 0.169674 |
| 511 | 0.0976529 | 0.215843 |

Вывод: как можно увидеть в таблице, алгоритм быстрого преобразования Фурье работает быстрее обычного алгоритма преобразования Фурье, в два и более раза, следовательно алгоритм быстрого преобразования Фурье, эффективнее.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9997e+00 | 128 | 0.6821e-14 | 128 | 0.5329e-16 |
| 255 | 0.9997e+00 | 128 | 0.1945e-12 | 128 | 0.1520e-14 |

Вывод: наблюдается два пика амплитуд = 128 при частоте m = 1 и m = 255. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 127 | 0.9997e+00 | 128 | -0.1959e-11 | 128 | -0.1530e-13 |
| 129 | 0.9997e+00 | 128 | 0.2151e-11 | 128 | 0.1681e-13 |

Вывод: наблюдается два пика амплитуд = 128 при частоте m = 127 и m = 129. Это означает, что сигнал описывается двумя высокочастотными векторами:

и

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9997e+00 | 128 | 0.4220e-11 | 128 | 0.3297e-13 |
| 255 | 0.9997e+00 | 128 | -0.4029e-11 | 128 | -0.3148e-13 |

Вывод: наблюдается два пика амплитуд = 128 при частоте m = 1 и m = 255. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9999e+00 | 256 | -0.2678e-14 | 256 | -0.1046e-16 |
| 511 | 0.9999e+00 | 256 | 0.5805e-11 | 256 | 0.2268e-13 |

Вывод: наблюдается два пика амплитуд = 256 при частоте m = 1 и m = 511. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 255 | 0.9999e+00 | 256 | -0.2289e-11 | 256 | -0.8943e-14 |
| 257 | 0.9999e+00 | 256 | 0.8105e-11 | 256 | 0.3166e-13 |

Вывод: наблюдается два пика амплитуд = 256 при частоте m = 255 и m = 257. Это означает, что сигнал описывается двумя высокочастотными векторами:

и

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9999e+00 | 256 | 0.1695e-10 | 256 | 0.6621e-13 |
| 511 | 0.9999e+00 | 256 | -0.1113e-10 | 256 | -0.4346e-13 |

Вывод: наблюдается два пика амплитуд = 256 при частоте m = 1 и m = 511. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

**Задание 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | -0.9999e+00 | -256 | 0.2678e-14 | 256 | 0.3142e+01 = π |
| 511 | -0.9999e+00 | -256 | -0.5805e-11 | 256 | -0.3142e+01 = -π |

Вывод: наблюдается два пика амплитуд = 256 при частоте m = 1 и m = 511. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

Фазовый спектр показывает начальный угол, на который была сдвинута соответствующая гармоника.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.1010e+03 | 25856 | 0.4441e-13 | 25856 | 0.1718e-17 |
| 511 | 0.1010e+03 | 25856 | 0.4441e-13 | 25856 | 0.2277e-13 |

Вывод: наблюдается два пика амплитуд = 25856 при частоте m = 1 и m = 511. Это означает, что сигнал описывается двумя низкочастотными векторами:

и

Как можно наблюдать, при отрицательных A, изменяется начальный угол = π, на который была сдвинута соответствующая гармоника, при положительном А, угол остаётся неизменным. Также числа А является амплитудой, т.е. при увеличении коэффициента А по модулю, амплитуда увеличивается на ту же величину.

**Задание 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | -0.9999e+00 | -256 | -0.3510e-13 | 256 | -0.3142e+01 = -π |
| 511 | -0.9999e+00 | -256 | -0.5780e-11 | 256 | -0.3142e+01 = -π |

Вывод: наблюдается два амплитудных пика частот при m = 1 и m = 511. При этом, можно видеть в столбце фазового спектра два отличных от нулю значения, равных -π. Это показывает на какой начальный угол была сдвинута соответствующая гармоника. Следовательно, величина смещения гармоники по фазе -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9999e+00 | 256 | -0.6482e-13 | 256 | -0.2532e-15 |
| 511 | 0.9999e+00 | 256 | 0.5850e-11 | 256 | 0.2285e-13 |

Вывод: наблюдается два амплитудных пика частот при m = 1 и m = 511. При этом, в столбце фазового спектра, фаза принимает значение ноль, с погрешностью e-13, что является истиной, ведь период функции cos - . Следовательно, величина смещения гармоники по фазе -

**Задание 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 0 | 0.2000e+01 | 512 | 0.0000e+00 | 512 | 0.0000e+00 |
| 1 | 0.2000e+01 | 256 | -0.5034e-13 | 256 | -0.1966e-15 |
| 511 | 0.2000e+01 | 256 | 0.5973e-11 | 256 | 0.2333e-13 |

Вывод: наблюдается два ‘старых’ амплитудных пика частот при m = 1 и m = 511, а также один ‘новый’ при m = 0 величиной равной 512. Это означает, что сигнал имеет постоянную составляющую:

Величина смещения сигнала равна 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 0 | 0.0000e+00 | -512 | 0.0000e+00 | 512 | 0.3142e+01 = π |
| 1 | -0.7530e-04 | 256 | 0.2445e-13 | 256 | 0.9552e-16 |
| 511 | -0.7530e-04 | 256 | 0.5669e-11 | 256 | 0.2214e-13 |

Вывод: наблюдается два ‘старых’ амплитудных пика частот при m = 1 и m = 511, а также один ‘новый’ при m = 0 величиной равной 512. Это означает, что сигнал имеет постоянную составляющую:

Величина смещения сигнала равна -1

**Задание 6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.1010e+01 | 258.56 | 0.1039e-13 | 258.56 | 0.4019e-16 |
| 511 | 0.1010e+01 | 258.56 | 0.5897e-11 | 258.56 | 0.2281e-13 |

Вывод: наблюдается два амплитудных пика частот при m = 1 и m = 511. Высокочастотных компонентов сигнала не наблюдается, т.к чисел m близких к N/2 - нет

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| m | *Re*(z) | *Re*() | *Im*() |  |  |
| 1 | 0.9899e+00 | 256 | -0.3293e-14 | 256 | -0.1286e-16 |
| 256 | -0.9900e+00 | 5.12 | 0.0000e+00 | 5.12 | 0.0000e+00 |
| 511 | 0.9899e+00 | 256 | 0.5812e-11 | 256 | 0.2270e-13 |

Вывод: наблюдается три амплитудных пика частот при m = 1, m = 256 (высокочастотная компонента сигнала) и m = 511.

Обнулим коэффициент под индексом 256 в дискретном преобразование и построим график.

Оранжевый график – результат фильтрации, синий график – исходный набор значений

Вывод: можем наблюдать режим фильтрации данных, благодаря устранению вручную высокочастотных компонент.

1. **Общий вывод**

В ходе лабораторной работы реализовали алгоритмы прямого и обратного преобразования Фурье, а также быстрые алгоритмы прямого и обратного преобразования Фурье с использованием шаблонного класса complex<double>.

Провели анализ амплитудного и фазового спектра, определили число частот для разных преобразований, величину смещения сигнала относительно оси абсцисс. Измерили время работы алгоритмов: быстрого и обычного преобразования Фурье. Сделали соответствующе мини-выводы по каждому пункту. Используя преобразование Фурье можно сжимать данные в несколько раз, а так же преобразование Фурье является мощным математическим аппаратом для анализа сигналов, в чём мы и убедились в данной лабораторной работе!