

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

Применение численных методов в электротехнике

**Изучение свойств преобразований Фурье в среде
программирования «Wolfram Mathematica»**

Лабораторный практикум

Санкт-Петербург – 2018

УДК 519.6: 517.44: 621.3: 681.3

Автор:

Кизеветтер Дмитрий Владимирович

Применение численных методов в электротехнике. Изучение свойств преобразований Фурье в среде программирования "Wolfram Mathematica". Лабораторный практикум. 2018. – 63 С.

Пособие соответствует государственному образовательному стандарту подготовки бакалавров и предназначено для студентов ВУЗов очного и заочного форм обучения, а также открытого дистанционного обучения в рамках направления 13.03.02 «Электроэнергетика и электротехника», изучающих дисциплины Б1.В.ДВ.8.1 «Применение численных методов в электротехнике» и Б1.В.ОД.10 «Компьютерные технологии».

Пособие знакомит студентов с основными свойствами рядов и преобразований Фурье. Подробно изучаются свойства прямого и обратного дискретного преобразования Фурье (ДПФ), программные средства среды программирования "Wolfram Mathematica" (WM), необходимые для изучения свойств ДПФ, приводятся примеры программ для ввода данных из текстового файла и осуществления ДПФ с последующим изучением основных свойств преобразования. Подробно рассмотрен алгоритм быстрого преобразования Фурье (БПФ), а также программа в среде программирования WM для реализации алгоритма прямого и обратного БПФ.

Пособие может быть полезно студентам и аспирантам, изучающим программирование, а также инженерам и научным работникам, работающим в среде программирования "Wolfram Mathematica".

Табл. 7, ил. 23, библиограф. 6

Введение

Прямое и обратное преобразования Фурье (ПФ) широко используется в различных областях науки для решения как теоретических так и прикладных задач. Наиболее важными приложениями являются гармонический анализ и задачи частотной фильтрации сигналов. Изучение рядов Фурье, интеграла Фурье и преобразований Фурье входит в учебные программы по курсу «Математика», читаемого на всех технических факультетах университета. Классическим учебным пособием со строгим изложением основ математического анализа, дающим подробные сведения в данной области, является «Курс дифференциального и интегрального исчисления. Том 3» Г.М. Фихтенгольца [1], первое издание которого вышло в 1949 г. Для студентов первого и второго курсов ВУЗов можно порекомендовать сокращенный вариант этого издания – «Основы математического анализа. Учебник. Часть 2» [2]. Основные понятия о рядах Фурье и преобразованиях Фурье, а также описание их основных свойств дано также в математических справочниках, наиболее известными из которых являются «Справочник по математике для научных работников и инженеров» (Г. Корн и Т. Корн) [3], а также справочник по математике И.Н. Бронштейна и К.А. Семендяева [4].

Быстрое преобразование Фурье (БПФ) является одной из основных операций цифровой обработки сигналов (ЦОС). Подробное описание алгоритмов, их программная и аппаратная реализация приводятся во многих учебных пособиях, в частности, в пособиях [5–6]. Изучение свойств и применений преобразований Фурье, способствует приобретению следующих компетенций: ОПК-2 – «способность применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач», ОПК-3 – «способность использовать методы анализа и моделирования электрических цепей», ПК-2 – «способность обрабатывать результаты

экспериментов», в соответствии с учебным планом дисциплины Б1.В.ДВ.8.1 «Применение численных методов в электротехнике».

Для изучения свойств преобразований Фурье, алгоритма быстрого прямого и обратного преобразования Фурье удобно использовать среду программирования “Mathematica” фирмы “Wolfram Research, Inc” (далее в тексте – “Wolfram Mathematica”, сокращенно – “WM”). Данное пособие рассчитано на пользователей, знакомых с основами программирования в среде WM. В пособии обобщены основные сведения о рядах и преобразованиях Фурье, рассмотрены основные свойства ПФ и коэффициентов ДПФ, алгоритм прямого и обратного быстрого ПФ. Приведены примеры программ для изучения свойств коэффициентов ДПФ, для считывания данных из текстового файла, для простейшей частотной фильтрации сигналов. Подробно рассмотрен пример программной реализации алгоритма быстрого прямого и обратного преобразования Фурье.

Значительное внимание в пособии уделено специфике программирования в среде WM – нумерации элементов списков, выборке элементов списка и т.д. Рассмотрены также отличия в существующих определениях прямого и обратного преобразований Фурье, а также дискретного преобразования Фурье.

Пособие соответствует государственному образовательному стандарту подготовки бакалавров и предназначено для студентов ВУЗов очного и заочного форм обучения, а также открытого дистанционного обучения в рамках направления 13.03.02 «Электроэнергетика и электротехника», изучающих дисциплины Б1.В.ДВ.8.1 «Применение численных методов в электротехнике» и Б1.В.ОД.10 «Компьютерные технологии». Пособие может быть полезно студентам, аспирантам, инженерам и научным работникам, изучающим программирование в среде “Wolfram Mathematica”.

1. Ряды Фурье, интеграл Фурье, преобразования Фурье. Общие сведения

Ряды Фурье и интегралы Фурье используются для решения многих теоретических и прикладных задач. Разложение в ряд Фурье является частным случаем разложения в ряд по ортогональным функциям. В электротехнике и радиотехнике основными применениями преобразований Фурье является спектральный анализ сигналов, в частности, определение коэффициентов гармоник несинусоидальных сигналов и построение частотных фильтров.

Периодическую функцию $f(t)$ с периодом T в некоторых случаях целесообразно заменить приближением $s(t)$ в виде тригонометрической суммы:

$$s(t) = a_0/2 + a_1 \cos(\omega t) + a_2 \cos(2\omega t) + \dots + a_n \cos(n\omega t) + \dots + b_1 \sin(\omega t) + b_2 \sin(2\omega t) + \dots + b_n \sin(n\omega t) + \dots \quad (1.1)$$

где $\omega = 2\pi/T$, n – целое положительное число, a_n , b_n – коэффициенты ряда Фурье, определяемые по формулам Эйлера-Фурье:

$$a_n = \frac{2}{T} \int_0^T f(t) \cos(n\omega t) dt \quad (1.2)$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin(n\omega t) dt \quad (1.3)$$

Ряд Фурье может быть также записан в виде:

$$s(t) = \frac{a_0}{2} + h_1 \sin(\omega t + \varphi_1) + h_2 \sin(2\omega t + \varphi_2) + \dots + h_n \sin(n\omega t + \varphi_n) + \dots$$

где:

$$h_n = (a_n^2 + b_n^2)^{1/2} \quad (1.4)$$

$$\operatorname{tg}(\varphi_n) = a_n / b_n \quad (1.5)$$

В общем случае, если задан интервал разложения функции $-\pi < t < \pi$, то ряд Фурье, порожденный действительной функцией $f(t)$, для которой существует

интеграл $\int_{-\pi}^{+\pi} |f(t)| dt$ (более точные условия сформулированы в справочниках

[1-2]), это – бесконечный тригонометрический ряд:

$$\frac{1}{2} a_0 + \sum_{n=1}^{\infty} (a_n \cos(nt) + b_n \sin(nt)) \equiv \sum_{m=-\infty}^{+\infty} c_m \exp(mti) \quad (1.6)$$

где m – целое число, i – мнимая единица, коэффициенты которого также определяются по формулам Эйлера-Фурье:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt) dt \quad (1.7)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(nt) dt \quad (1.8)$$

$$c_n = \frac{1}{2} (a_n - ib_n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \exp(-nti) dt \quad (1.9)$$

Коэффициенты c_n в общем случае являются комплексными числами.

Если задан интервал разложения функции $-T/2 < t < T/2$, то ряд Фурье, порожденный действительной функцией $f(t)$, для которой существует

интеграл $\int_{-T/2}^{+T/2} |f(t)| dt$, это – также бесконечный тригонометрический ряд:

$$\frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos(n\omega t) + b_n \sin(n\omega t)) \equiv \sum_{m=-\infty}^{+\infty} c_m \exp(mt\omega i) \quad (1.10)$$

где:

$$a_n = \frac{2}{T} \int_{-T/2}^{+T/2} f(t) \cos(n\omega t) dt \quad (1.11)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{+T/2} f(t) \sin(n\omega t) dt \quad (1.12)$$

$$c_n = \frac{1}{2}(a_n - ib_n) = \frac{1}{T} \int_{-T/2}^{+T/2} f(t) \exp(-nt\omega i) dt \quad (1.13)$$

Если $T=2\pi$ ($\omega=2\pi/T=1$), то формулы (1.11) – (1.13) переходят в формулы (1.7) – (1.9).

Интеграл Фурье, порожденный действительной функцией $f(t)$, по определению есть

$$\frac{1}{\pi} \int_0^{+\infty} d\omega \int_{-\infty}^{+\infty} f(\tau) \cos(\omega(t - \tau)) d\tau \quad (1.14)$$

Равноценными определениями интеграла Фурье являются:

$$\int_{-\infty}^{+\infty} c(\nu) \exp(2\pi i \nu t) d\nu \quad (1.15)$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} c(\omega) \exp(i\omega t) d\omega \quad (1.16)$$

где $\nu=\omega/2\pi$, а функции $c(\nu)$ и $c(\omega)$ определяются следующими формулами:

$$c(\nu) = \int_{-\infty}^{+\infty} f(\tau) \exp(-2\pi i \nu \tau) d\tau \quad (1.17)$$

$$c(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(\tau) \exp(-i\omega\tau) d\tau \quad (1.18)$$

Функции $c(v)$ и $c(\omega)$ называются преобразованием Фурье функции $f(t)$. Так как функции (1.17) и (1.18) отличаются нормировочным множителем и масштабным коэффициентом, иногда эти функции обозначают различными символами, например, для (1.17), как и ранее – $c(v)$, а для (1.18) – $C(\omega)$. Тогда можно записать:

$$C(\omega) = \frac{1}{\sqrt{2\pi}} c\left(\frac{\omega}{2\pi}\right) \quad (1.19)$$

В некоторых случаях удобно использовать косинус- и синус- интегралы Фурье, которые определяются соответственно как

$$2 \int_0^{+\infty} c_c(v) \cos(2\pi vt) dv \equiv \sqrt{\frac{2}{\pi}} \int_0^{+\infty} C_c(\omega) \cos(\omega t) d\omega \quad (1.20)$$

$$2 \int_0^{+\infty} c_s(v) \sin(2\pi vt) dv \equiv \sqrt{\frac{2}{\pi}} \int_0^{+\infty} C_s(\omega) \sin(\omega t) d\omega \quad (1.21)$$

где:

$$c_c(v) = 2 \int_0^{+\infty} f(\tau) \cos(2\pi v\tau) d\tau \quad (1.22)$$

$$C_c(\omega) = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} f(\tau) \cos(\omega\tau) d\tau \equiv \frac{1}{\sqrt{2\pi}} c_c\left(\frac{\omega}{2\pi}\right) \quad (1.23)$$

$$c_s(v) = 2 \int_0^{+\infty} f(\tau) \sin(2\pi v\tau) d\tau \quad (1.24)$$

$$C_s(\omega) = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} f(\tau) \sin(\omega\tau) d\tau \equiv \frac{1}{\sqrt{2\pi}} c_s\left(\frac{\omega}{2\pi}\right) \quad (1.25)$$

Пусть задана функция $c(\omega)$ или $C(\omega)$. Тогда:

$$f(t) = \int_{-\infty}^{+\infty} c(\nu) \exp(2\pi i \nu t) d\nu \quad (1.26)$$

или

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} C(\omega) \exp(i\omega t) d\omega \quad (1.27)$$

В этом случае функция $f(t)$ называется обратным преобразованием Фурье функции $c(\nu)$ (или $C(\omega)$). Подробные сведения об условиях существования обратного преобразования также даны в учебных пособиях [1-2] и справочнике [3]. Следует отметить, что в научной и учебной литературе приводятся и другие определения преобразований Фурье, отличающиеся нормировкой или (и) знаком в показателе степени экспоненты ($\exp(\pm i\omega t)$). В частности, в учебных пособиях [1-2] прямое преобразование Фурье имеет знак «плюс», а обратное преобразование – «минус». В некоторых случаях нормировочный множитель $(2\pi)^{-1/2}$ для прямого преобразования Фурье не учитывают, а при выполнении обратного преобразования используют множитель $1/(2\pi)$.

Введем обозначение $\mathfrak{F}[f(t)]$ – прямое преобразование Фурье. Если преобразование существует, то оно обладает свойством линейности:

$$\mathfrak{F}[\alpha f_1(t) + \beta f_2(t)] = \alpha \mathfrak{F}[f_1(t)] + \beta \mathfrak{F}[f_2(t)] \quad (1.28)$$

К преобразованию Фурье можно применить теорему масштаба (или теорему подобия):

$$\mathfrak{F}[f(\alpha t)] = \frac{1}{\alpha} c\left(\frac{\nu}{\alpha}\right) \quad (1.29)$$

а также теорему сдвига:

$$\mathfrak{F}[f(t + \tau)] = c(\nu) \exp(2\pi i \nu \tau) \quad (1.30)$$

Для преобразования Фурье также справедливы теоремы: теорема Бореля о свертке, теорема о модуляции, теорема о дифференцировании, теорема Парсеваля. Все вышеуказанные свойства справедливы и для косинус- и синус- преобразований Фурье. Подробное описание свойств преобразований приведено в учебных пособиях и справочниках [1-6].

Преобразования Фурье для наиболее распространенных функций даны в таблице 1.1. Более полная таблица преобразований Фурье приводится, например, в справочнике [3]. Данные, представленные в таблице, могут быть полезны для проверки правильности работы компьютерных программ, в частности, в среде программирования “Wolfram Mathematica”. Так, в среде “Wolfram Mathematica” имеется возможность выполнять символьные операции, в том числе преобразования Фурье. Функция `FourierTransform[Expr, t, ω]` выполняет символьное преобразование Фурье функции *Expr*, заданной аналитически. В качестве примера на рис. 1.1. приведена копия экранной формы символьного преобразования Фурье функций $\exp(-t^2/2)$ и $\exp(-pt^2)$. В среде “Mathematica” указанные функции преобразования будут иметь вид: `FourierTransform[Exp[-t^2/2], t, ω]` и `FourierTransform[Exp[-p*t^2], t, ω]`.

Таблица 1.1. Преобразования Фурье

$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} C(\omega) \exp(i\omega t) d\omega$	$C(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(\tau) \exp(-i\omega\tau) d\tau$
$f(t)$	$C(\omega)$
$\frac{\sin(\alpha t)}{t} \quad (\alpha > 0)$	$\begin{matrix} (\frac{\pi}{2})^{1/2} & \omega < \alpha \\ 0 & \omega > \alpha \end{matrix}$
$\begin{matrix} \exp(i\alpha t) & p < t < \alpha \\ 0 & t < p, t > q \end{matrix}$	$\begin{matrix} \frac{i}{(2\pi)^{1/2}} \frac{\exp(iq(\alpha - \omega) - \exp(ip(\alpha - \omega)))}{\omega - \alpha} & \omega \neq \alpha \\ \frac{q - p}{(2\pi)^{1/2}} & \omega = \alpha \end{matrix}$
$\begin{matrix} \exp(-ct + i\alpha t) & t > 0 \\ 0 & t < 0 \end{matrix}$	$\frac{i}{(2\pi)^{1/2}(\alpha - \omega + ic)}$
$\begin{matrix} \exp(-pt^2) \\ \exp(-t^2/2) \end{matrix}$	$\begin{matrix} (2p)^{-1/2} \exp(-\omega^2/(4p)) \\ \exp(-\omega^2/2) \end{matrix}$
$\begin{matrix} \exp(i\alpha t^2) \\ \cos(\alpha t^2) \quad (\alpha > 0) \\ \sin(\alpha t^2) \end{matrix}$	$\begin{matrix} (2\alpha)^{-1/2} \exp(-i(\frac{\omega^2}{4\alpha} - \frac{\pi}{4})) \\ (2\alpha)^{-1/2} \cos(\frac{\omega^2}{4\alpha} - \frac{\pi}{4}) \\ (2\alpha)^{-1/2} \sin(-\frac{\omega^2}{4\alpha} + \frac{\pi}{4}) \end{matrix}$
$\begin{matrix} t ^{-s} & 0 < \text{Re}(s) < 1 \\ t ^{-1/4} \end{matrix}$	$\frac{2}{\pi} \Gamma(1-s) \sin(s\pi/2) \omega ^{s-1} \omega ^{1/2}$
$\frac{\exp(-\alpha t)}{ t ^{1/2}} \quad (\alpha > 0)$	$\frac{((\alpha^2 + \omega^2)^{1/2} + \alpha)^{1/2}}{(\alpha^2 + \omega^2)^{1/2}}$
$\begin{matrix} (\alpha^2 - t^2)^{-1/2} & t < \alpha \\ 0 & t > \alpha \end{matrix}$	$(\pi/2)^{1/2} J_0(\alpha\omega)$

$\frac{\sin(\beta(\alpha^2 + t^2)^{1/2})}{(\alpha^2 - t^2)^{1/2}}$	0	$ \omega > \beta$
	$(\pi/2)^{1/2} J_0(\alpha(\omega^2 - \beta^2)^{1/2})$	$ \omega < \beta$
$\frac{sh(\alpha t)}{sh(\pi t)} \quad -\pi < \alpha < \pi$	$\left(\frac{1}{2\pi}\right)^{1/2} \frac{\sin(\alpha)}{ch(\omega) + \cos(\alpha)}$	

В таблице: $\Gamma(x)$ – гамма-функция, $J_0(x)$ – функция Бесселя нулевого порядка, $sh(x)$, $ch(x)$ – гиперболические синус и косинус соответственно.

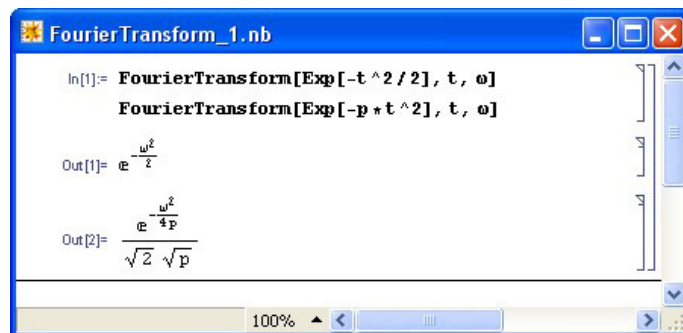


Рис. 1.1. Копия экранной формы активного окна после выполнения символьных преобразований Фурье

Как следует из рис. 1.1, полученный результат соответствует преобразованиям Фурье, приводимым в Табл. 1.1. В среде “Mathematica” используются также символьные синус- и косинус- преобразования Фурье: $\text{FourierSinTransform}[expr, t, \omega]$ и $\text{FourierCosTransform}[expr, t, \omega]$, а также обратные символьные преобразования Фурье: $\text{InverseFourierTransform}[expr, \omega, t]$, $\text{InverseFourierSinTransform}[expr, \omega, t]$, $\text{InverseFourierCosTransform}[expr, \omega, t]$. Пример выполнения обратного символьного преобразования Фурье приведен на рис. 1.2.

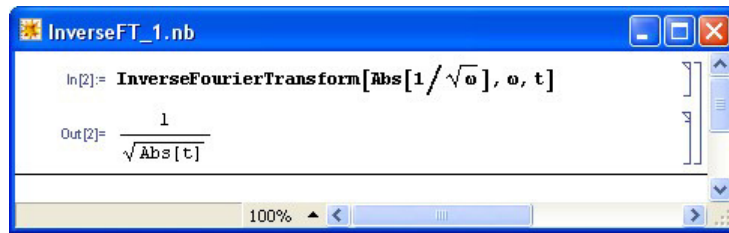


Рис. 1.2. Копия экранной формы активного окна после выполнения обратного символьного преобразования Фурье

Как видно из рис. 1.2, результат преобразования совпадает с табличной функцией (Табл. 1.1.).

В радиоэлектронные устройства для цифровой обработки сигналов (ЦОС), в частности, в устройства, выполняющие преобразование Фурье, исходный сигнал поступает либо изначально в цифровом виде, либо от аналого-цифрового преобразователя (АЦП) в виде последовательности чисел. То есть, исходный сигнал для ЦОС является дискретным. Формально в преобразовании (1.18) можно заменить интеграл на сумму, используя дискретные значения сигнала. Однако для выполнения преобразований Фурье дискретного набора чисел существуют специальные алгоритмы, позволяющие существенно ускорить выполнение расчетов. Обозначим исходную функцию, т.е. преобразуемый сигнал, как $f[k]$, где k – номер цифрового значения в последовательности. Для того, чтобы отличать непрерывную (или кусочно-непрерывную) функцию $f(t)$ от функции, заданной дискретно $f[k]$, будем использовать символы скобок «[]». Положим, что общее количество точек последовательности равно N . Отсчет номеров элементов k начинается с 0; последним номером является $k=N-1$. Преобразование Фурье функции, заданной дискретными значениями, имеет дискретные значения спектральной плотности $S[n]$, где n – номер элемента преобразования, максимальное значение которого также равно $N-1$. Такое преобразование называют дискретным преобразованием Фурье (ДПФ). По

аналогии с преобразованием (1.18) и выражением (1.6) коэффициенты преобразования Фурье $C[n]$ можно представить в виде:

$$C[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f[k] \left(\cos\left(\frac{2\pi i}{N-1} kn\right) + i \sin\left(\frac{2\pi i}{N-1} kn\right) \right) \quad (1.31)$$

или в виде:

$$C[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f[k] W_N^{kn} \quad (1.32)$$

где W_N^{kn} – так называемый поворотный множитель (или поворотный коэффициент), определяемый выражением:

$$W_N^{kn} \equiv \exp\left(\frac{2\pi i}{N-1} kn\right) \quad (1.33)$$

Классическое преобразование Фурье, рассмотренное выше (1.18), отличается от рассматриваемого ДПФ (1.32) нормировочным множителем и разными знаками в показателе степени экспоненты, кроме того, что в первом случае исходная функция является кусочно-непрерывной, а во втором случае – дискретной. В прямом преобразовании Фурье (1.18) показатель степени имеет знак «минус», а в ДПФ (1.32) – «плюс», в обратном преобразовании – «плюс» (1.27) и «минус» соответственно. Следует также учесть, что во многих учебных пособиях, как например [5-6], монографиях и научных статьях используется знак «минус» в показателе степени экспоненты. В среде программирования “Wolfram Mathematica” определением ДПФ являются выражения (1.32) и (1.33), т.е. при прямом дискретном преобразовании Фурье показатель степени экспоненты (1.33) – «плюс», а при обратном – «минус».

Коэффициенты дискретного преобразования Фурье имеют ряд свойств, характерных только ДПФ и используются для создания алгоритмов быстрого

преобразования Фурье (БПФ), построения спектральных фильтров и других применений. Поэтому изучению дискретного преобразования Фурье и его свойств посвящен отдельный параграф пособия, представленный ниже.

2. Основные свойства дискретного преобразования Фурье

Помимо базовых функций для изучения свойств преобразований Фурье, необходимы следующие функции среды программирования “Wolfram Mathematica”: “Table”, “Fourier”, “InverseFourier”, “Random”. Их назначение и описание приведены в таблице 2.1.

Таблица 2.1.

Функция	Запись функции в среде “Mathematica”	Описание функции
Table	Table[<i>expr</i> , { <i>i</i> _{max} }]	Генерирует список из <i>i</i> _{max} копий выражения <i>expr</i>
	Table[<i>expr</i> , { <i>i</i> , <i>i</i> _{max} }]	Генерирует список значений выражения <i>expr</i> для <i>i</i> от 1 до <i>i</i> _{max}
	Table[<i>expr</i> , { <i>i</i> , <i>i</i> _{min} , <i>i</i> _{max} }]	Генерирует список значений выражения <i>expr</i> для <i>i</i> от <i>i</i> _{min} до <i>i</i> _{max}
	Table[<i>expr</i> , { <i>i</i> , <i>i</i> _{min} , <i>i</i> _{max} , <i>di</i> }]	Генерирует список значений выражения <i>expr</i> для <i>i</i> от <i>i</i> _{min} до <i>i</i> _{max} с шагом <i>di</i> .
Fourier	Fourier[<i>list</i>]	Производит дискретное преобразование Фурье от списка комплексных значений <i>list</i> .
InverseFourier	InverseFourier[<i>list</i>]	Производит дискретное обратное преобразование Фурье от списка комплексных значений <i>list</i> .
Random	Random[]	Генерирует случайное значение действительной переменной (Real) с равномерным распределением от 0 до 1.
	Random[Integer]	Генерирует значения 0 или 1 с вероятностью 1/2.

	<code>Random[Complex]</code>	Генерирует значения от 0 до $1+i$ с равномерным распределением действительной и мнимой части.
	<code>Random[Type, {z_{min}, z_{max}}]</code>	Генерирует случайные значения с равномерным распределением для переменной типа <i>Type</i> (который может принимать значения <i>Integer</i> , <i>Real</i> или <i>Complex</i>) в диапазоне от z_{min} до z_{max} .

Исходные данные могут быть заданы в виде списка, элементы которого введены в строку программы и содержат необходимые значения (например, $data=\{0,0.25,0.5,0.75\}$ – список из четырех значений) или с помощью функции `Table` (например, $data=Table[(i-1)*0.25,\{i,1,4\}]$ – выражение, формирующее список, аналогичный предыдущему примеру). Следует учесть, что в среде “Mathematica” нумерация элементов списков начинается с единицы, а при классическом рассмотрении задачи дискретного преобразования Фурье – с нуля. Так например, при четном количестве точек N , точке симметрии дискретного спектра сигнала $C[j]$ соответствует значение $N/2$, а в “Mathematica” в списке $data$ – $N/2+1$. Основные свойства ДПФ обобщены в таблицах 2.2 и 2.3. Необходимо также учесть, что символ N в среде программирования “Mathematica” является служебным, поэтому его нельзя использовать в качестве переменной для обозначения количества точек. Далее в демонстрационных программах общее количество точек обозначено как Np .

Таблица 2.2.

N – четное число			
	Описание ДПФ	“Mathematica”	Примечание
1	$\text{Im}(C[0])=0$	$\text{Im}[ftd[[1]]]=0$	Мнимая часть нулевого коэффициента БПФ (первого элемента списка) равна 0
2	$\text{Re}(C[0])$	$\text{Re}[ftd[[1]]]$	Среднее значение по N точкам (без учета множителя $N^{1/2}$).
3	$N/2$	$N/2+1$	Точка симметрии (нечетной симметрии) спектра
4	$\text{Im}(C[N/2])=0$	$\text{Im}[ftd[[N/2+1]]]=0$	Мнимая часть значения спектра БПФ в точке симметрии равна нулю
5	$\text{Re}(C[N/2-k])=\text{Re}(C[N/2+n])$	$\text{Re}[ftd[[N/2+1-k]]]=\text{Re}[ftd[[N/2+1+n]]]$	Симметрия действительной части спектра БПФ относительно точки $N/2$
6	$\text{Im}(C[N/2-n])=-\text{Im}(C[N/2+n])$	$\text{Im}[ftd[[N/2+1-n]]]=-\text{Im}[ftd[[N/2+1+n]]]$	Нечетная симметрия мнимой части спектра БПФ относительно точки $N/2$

Таблица 2.3.

N – нечетное число			
	Описание ДПФ	“Mathematica”	Примечание
1	$\text{Im}(C[0])=0$	$\text{Im}[ftd[[1]]]=0$	Мнимая часть нулевого коэффициента БПФ (первого элемента списка) равна 0
2	$\text{Re}(C[0])$	$\text{Re}[ftd[[1]]]$	Среднее значение по N точкам (без учета

			множителя $N^{1/2}$).
3	$C[N/2]=C[N/2+1]$	$ftd[[N/2+1]]=ftd[[N/2+2]]$	Равенство значений спектра БПФ в точках $N/2$ и $N/2+1$
4	$\text{Im}(C[N/2])\neq 0$	$\text{Im}[ftd[[N/2+1]]]\neq 0$	Мнимая часть значения спектра БПФ в точке симметрии в общем случае не равна нулю
5	$\text{Re}(C[N/2-k])=\text{Re}(C[N/2+1+n])$	$\text{Re}[ftd[[N/2+1-n]]]=\text{Re}[ftd[[N/2+2+n]]]$	Симметрия действительной части спектра БПФ
6	$\text{Im}(C[N/2-n])=-\text{Im}(C[N/2+1+n])$	$\text{Im}[ftd[[N/2+1-n]]]=-\text{Im}[ftd[[N/2+2+n]]]$	Нечетная симметрия мнимой части спектра БПФ

Рассмотрим следующий пример (рис. 2.1). Выберем четное количество значений в исходном списке, т.е. зададим $Np=8$. Сформируем список исходных действительных (*Real*) значений *data*, используя функцию *Table* и генератор случайных чисел. Используя функцию дискретного преобразования Фурье *Fourier*, получим список комплексных значений дискретного Фурье спектра *ftd*.

```

test_1_01.nb *
In[1]:= Np = 8
data = Table[2 * Random[] - 1, {k, 0, Np - 1}]
ftd = Fourier[data]

Out[1]:= 8

Out[2]:= {-0.725475, 0.603942, -0.878068, -0.680047, 0.725985, 0.20467, 0.973586, 0.904091}

Out[3]:= {0.399051 + 0. i, -0.0173163 - 0.950875 i,
-0.0335904 + 0.206676 i, -1.00902 + 0.358442 i, -0.331148 + 0. i,
-1.00902 - 0.358442 i, -0.0335904 - 0.206676 i, -0.0173163 + 0.950875 i}

```

Рис. 2.1. Пример заполнения списка случайными числами и выполнения ДПФ

В результате выполнения программы получаем: список исходных значений – строка `Out[2]` и список значений дискретного преобразования Фурье от списка `data` – `Out[3]`. Первый элемент списка значений ДПФ (в данном примере – $0.399051+0.i$) является нулевым элементом ДПФ, характеризующим среднее значение (av) элементов исходного списка с учетом нормировки:

$$av=ftd[[1]]/Np^{(1/2)}. \quad (2.1)$$

Проверить правильность определения среднего значения можно суммированием всех элементов списка, поделив сумму на количество точек Np , используя, например, следующий код:

$$av=Sum[data[[j]],{j,1,Length[data]}/Np \quad (2.2)$$

В правильности свойств ДПФ, указанных в табл. 2.2 можно убедиться, проанализировав полученные значения спектра БПФ. Так, например, действительная часть значения первой точки спектра БПФ (второго элемента списка `ftd`) – -0.0173163 равна действительной части значения последней точки в спектре ДПФ (последнего элемента списка).

При большом количестве точек ДПФ найти нужные значения в полученном списке может быть затруднительно. В этом случае целесообразно выводить на экран только заданный элемент списка. Так например, четвертый элемент списка можно вывести строкой «`ftd[[4]]`». В качестве примера на рис. 2.2 приведен результат выполнения программы для случая 128 точечного ДПФ. Для предотвращения вывода полного списка исходных данных, а также всего спектра ДПФ, строки в программе завершаются точкой с запятой («;»). В данном примере (рис. 2.2.) выводятся только номер точки ДПФ и два значения величины ДПФ с использованием функции «Print».

```

test_1_02.nb
Np = 128;
data = Table[2 * Random[] - 1, {k, 0, Np - 1}];
ftd = Fourier[data];
k = 2
Print["val. = ", ftd[[k + 1]], " and ", ftd[[Np - k + 1]]]

2

val. = 0.339921 - 0.442449 i and 0.339921 + 0.442449 i

```

Рис. 2.2. Пример вывода двух значений элементов списка ДПФ

При повторном запуске программы исходные данные изменяются вследствие использования генератора случайных чисел при формировании списка *data*, а все указанные выше свойства ДПФ – сохраняются.

Для прикладных целей наиболее актуальным является изучение свойств дискретного преобразования Фурье для гармонических сигналов. Традиционно свойства ДПФ изучают на примере восьмиточечного преобразования ($N=8$). В этом случае выражение для коэффициентов ДПФ

$$C[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f[k] \left(\cos\left(\frac{2\pi i}{N-1} kn\right) + i \sin\left(\frac{2\pi i}{N-1} kn\right) \right) \quad (2.3)$$

можно представить в виде:

$$C[n] = \frac{1}{2\sqrt{2}} \sum_{k=0}^7 f[k] \left(\cos\left(\frac{2\pi i}{7} kn\right) + i \sin\left(\frac{2\pi i}{7} kn\right) \right) \quad (2.4)$$

Действительная и мнимая части поворотного множителя W_N^{kn} определяются тригонометрическими функциями и имеют хорошо известные значения, приводимые в таблицах (табл. 2.4 и 2.5)

Табл. 2.4

k	$\text{Re}(W_N^{kn})$							
	n							
	0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1	1
1	1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	-1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$
2	1	0	-1	0	1	0	-1	0
3	1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	-1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$
4	1	-1	-1	-1	-1	-1	-1	-1
5	1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	-1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$
6	1	0	-1	0	1	0	-1	0
7	1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	-1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$

Табл. 2.5

k	$\text{Im}(W_N^{kn})$							
	n							
	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	$1/\sqrt{2}$	1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	-1	$-1/\sqrt{2}$
2	0	1	0	-1	0	1	0	-1
3	0	$1/\sqrt{2}$	-1	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	1	$-1/\sqrt{2}$
4	0	0	0	0	0	0	0	0
5	0	$-1/\sqrt{2}$	1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	-1	$1/\sqrt{2}$

6	0	-1	0	1	0	-1	0	1
7	0	$-1/\sqrt{2}$	-1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	1	$1/\sqrt{2}$

Рассмотрим более подробно ДПФ для $N=8$ от гармонических сигналов вида (при $m=1$ период по переменной k равен 8):

$$fc[k] = \cos\left(\frac{2\pi}{8} km\right), \quad (2.5)$$

$$fs[k] = \sin\left(\frac{2\pi}{8} km\right) \quad (2.6)$$

Аналогично предыдущему примеру сформируем список из восьми значений исходной функции с помощью функции “Table”. В выражениях (2.5), (2.6) список значений создается по переменной k , значения которой изменяются от 0 до $N-1$, а коэффициент m задает период функций fc и fs . Список из восьми значений спектра сигнала ДПФ получим с помощью функции “Fourier”. Для визуализации исходных данных и полученного спектра следует использовать функцию “ListPlot”. Целесообразно ввести дополнительный параметр в функции “ListPlot” для увеличения размера отображаемых на графике точек, например: “PlotStyle→PointSize[0.03]”. Копия экранной формы при выполнении программы приведена на рис. 2.3. В результате расчетов получаем: список значений исходной функции – Out[3], график исходной функции – Out[4], список значений ДПФ – Out[5], график действительной части значений ДПФ – Out[6], график мнимой части значений ДПФ – Out[7]. Заданная величина коэффициента m , равная 1, соответствует одному периоду функции fc для коэффициентов k от 0 до N (т.е. период равен $N+1$). Так как косинус является четной функцией, только действительная часть значений ДПФ может быть отлична от нуля. То есть, $\text{Im}(C[n])=0$ для любого значения n . Из полученных данных также следует,

что действительная часть коэффициентов $C[n]$ имеет ненулевое значение только для второго и восьмого элементов списка, что соответствует первому и седьмому коэффициенту ДПФ. Выполнив аналогичные расчеты для других значений m , можно обобщить полученные результаты:

$$\operatorname{Re}(C[n]) \neq 0, \text{ если } n=m \parallel n=N-m \quad (2.7)$$

$$\operatorname{Re}(\operatorname{ftd}[[n]]) \neq 0, \text{ если } n=m+1 \parallel n=N-m+1 \quad (2.8)$$

Выражение (2.7) соответствует классическому рассмотрению задачи дискретного преобразования Фурье, а выражение (2.8) – нумерации элементов в списках программной среды “Wolfram Mathematica”. В рассматриваемом примере значение $m=4$ соответствует $N/2$. В этом случае ненулевое значение имеет только один коэффициент ДПФ с номером 4, который является пятым элементом в списке *ftd* используемой программы. Выражения (2.7) и (2.8) справедливы, в частности, для случая $m=0$ (постоянное значение исходной функции) и случаев $4 < m \leq 8$. Отметим также, что в данном примере (рис. 2.3) величины $\pm 9.66741 \cdot 10^{-17}$ являются ошибкой вычислений, обусловленной операциями с плавающей запятой. Указанные значения следует расценивать как нулевые.

Для нечетной функции (2.6) действительная часть коэффициентов ДПФ $C[n]$ будет равна нулю, а отличным от нуля может быть только мнимая часть коэффициентов $C[n]$. Пример расчета и визуализации значений коэффициентов ДПФ для $m=1$ приведен на рис. 2.4. Как следует из результатов расчета, для нечетной функции *fs* характерны свойства, аналогичные (2.7), (2.8) четной функции:

$$\operatorname{Im}(C[n]) \neq 0, \text{ если } n=m \parallel n=N-m \quad (2.9)$$

$$\operatorname{Im}(\operatorname{ftd}[[n]]) \neq 0, \text{ если } n=m+1 \parallel n=N-m+1 \quad (2.10)$$

Из полученных данных следует, что $C[n=m] = -C[n=N-m]$. Это соответствует свойству №6 в таблице 2.2. Все остальные свойства, указанные в таблице 2.2

также выполняются. Необходимо отметить, что полученные значения коэффициентов БПФ четной функции f_c соответствуют значениям коэффициентов поворота в таблице 2.4, а нечетной – f_s – в таблице 2.5.

Если исходную функцию можно представить в виде суммы гармонических функций с некоторыми весовыми коэффициентами, то и коэффициенты ДПФ также можно представить в виде суммы гармоник с теми же коэффициентами, вследствие линейности преобразований Фурье.

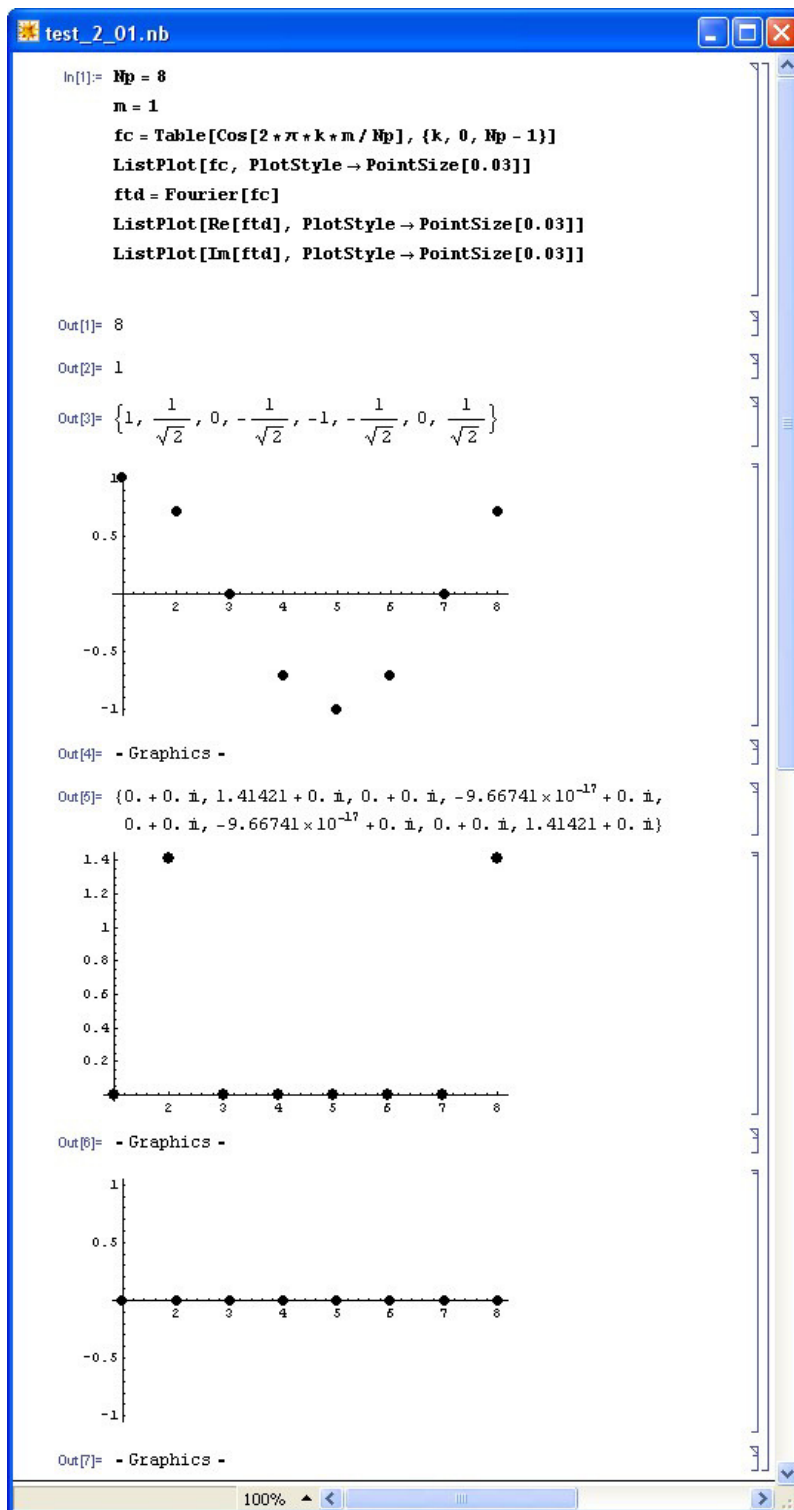


Рис. 2.3. Пример расчета и визуализации значений коэффициентов ДПФ для $m=1$ четной функции

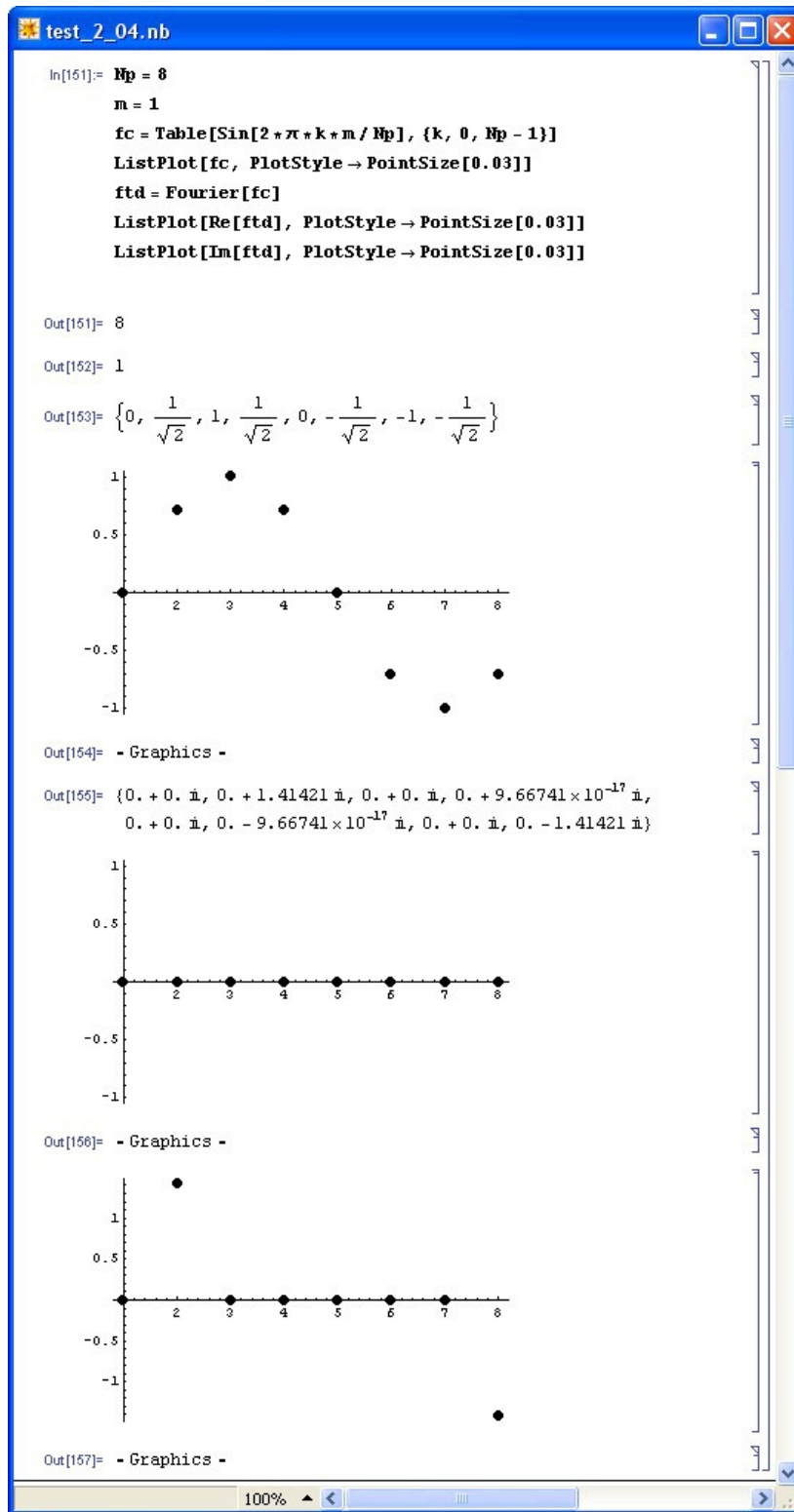


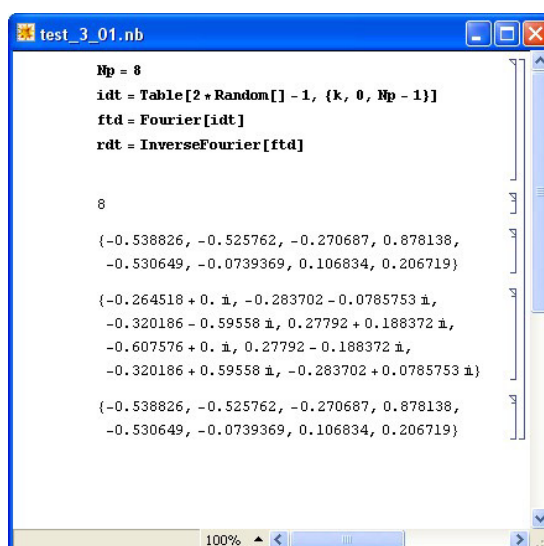
Рис. 2.4. Пример расчета и визуализации значений коэффициентов ДПФ для $m=1$ нечетной функции

Задание для самостоятельной работы

1. Для примера, рассмотренного выше (рис. 2.1), выведите несколько пар значений коэффициентов дискретного преобразования Фурье $C[n]$ для которых выполняется четная и нечетная симметрия относительно точки $N/2$.
2. Рассчитайте среднее значение исходной функции, сформированной в виде списка в соответствии с программой, приводимой на рис. 2.2. Оцените, на какую величину изменяется среднее значение при многократном запуске программы, вследствие использования генератора случайных чисел.
3. Выполните расчет и визуализацию в виде графиков значений коэффициентов ДПФ для $m=2, 3$ и 4 , аналогично примеру, приведенному на рис. 2.3 и 2.4.
4. Выполните расчет и визуализацию в виде графиков значений коэффициентов ДПФ для случая исходной функции, представляющей собой сумму двух функций вида (2.5) и (2.6) с различными значениями коэффициента m . Проанализируйте полученные данные.

3. Обратное дискретное преобразование Фурье

Для выполнения обратного дискретного преобразования Фурье в среде ‘Wolfram Mathematica’ используется функция ‘InverseFourier’ (см. табл. 2.1). Правильное вычисление коэффициентов обратного ДПФ, т.е. исходной функции, возможно только в случае, если коэффициенты ДПФ (спектра Фурье) соответствуют требованиям, указанным в Табл. 2.2 и 2.3. Простейшей проверкой выполнения прямого и обратного ДПФ может служить пример, приводимый на рис. 3.1. Как следует из полученных данных, восстановленный с помощью обратного ДПФ список значений соответствует списку значений исходной функции.



```
test_3_01.nb

Np = 8
idt = Table[2 * Random[] - 1, {k, 0, Np - 1}]
ftd = Fourier[idt]
rdt = InverseFourier[ftd]

8

{-0.538826, -0.525762, -0.270687, 0.878138,
-0.530649, -0.0739369, 0.106834, 0.206719}

{-0.264518 + 0. i, -0.283702 - 0.0785753 i,
-0.320186 - 0.59558 i, 0.27792 + 0.188372 i,
-0.607576 + 0. i, 0.27792 - 0.188372 i,
-0.320186 + 0.59558 i, -0.283702 + 0.0785753 i}

{-0.538826, -0.525762, -0.270687, 0.878138,
-0.530649, -0.0739369, 0.106834, 0.206719}
```

Рис. 3.1. Пример выполнения программы прямого и обратного ДПФ

Рассмотрим выполнение операций прямого и обратного дискретного преобразования Фурье на примере суммы двух гармонических функций. Пример выполнения программы представлен на рис. 3.2. В данной программе количество точек в ДПФ задано равным 128, коэффициенты $m1$, $k1$ и $m2$, $k2$ (для четной и нечетной функций) заданы различными. Исходной функцией для ДПФ является список значений ff , график которого выводится в процессе выполнения программы (рис. 3.2). Прямое ДПФ выполняется функцией ‘Fourier’. Для построения графика полученных действительных и мнимых

частей значений коэффициентов ДПФ используется функция “ListPlot” с явным описанием размера символов (точек) и диапазона чисел по осям графика (рис. 3.2, строки Out[6] и Out[8]). При автоматическом выборе диапазона по осям графика визуализация полученных данных может быть затруднена. Далее в программе выполняется обратное ДПФ спектра ftd , в результате которого формируется список значений yy восстановленной исходной функции. Построение графика дискретной функции yy также производится с помощью функции “ListPlot”:

$$\text{ListPlot}[yy, \text{PlotStyle} \rightarrow \text{PointSize}[0.03]] \quad (3.1)$$

Однако в некоторых случаях, попытка построения графика с использованием выражения (3.1) приводит к ошибке выполнения программы. Это обусловлено тем, что в результате выполнения обратного ДПФ значения восстановленной функции могут иметь ненулевую мнимую часть, вследствие чего построение графика с использованием функции “ListPlot” становится невозможным. Так как известно, что исходная функция должна содержать только действительные числа, целесообразно выводить график только действительной части дискретных величин списка yy . Например:

$$\text{ListPlot}[\text{Re}[yy], \text{PlotStyle} \rightarrow \text{PointSize}[0.03]] \quad (3.2)$$

Ненулевые значения коэффициентов $C[n]$ в полученном спектре ДПФ, соответствующие меньшей из частот ($m=2$, коэффициенты $C[2]$ и $C[126]$, в программе – $ftd[[3]]$ и $ftd[[127]]$) заменим на нулевые значения. Далее выполним обратное ДПФ. В результате преобразования получим функцию, в которой исключена меньшая из частот, то есть создадим частотный фильтр (предполагается, что исходная функция является зависимостью от времени). График функции, полученной в результате расчета, приведен на рис. 3.1, строка “Out[15]”. График подтверждает исключение меньшей из частот при использовании данного простейшего цифрового фильтра.

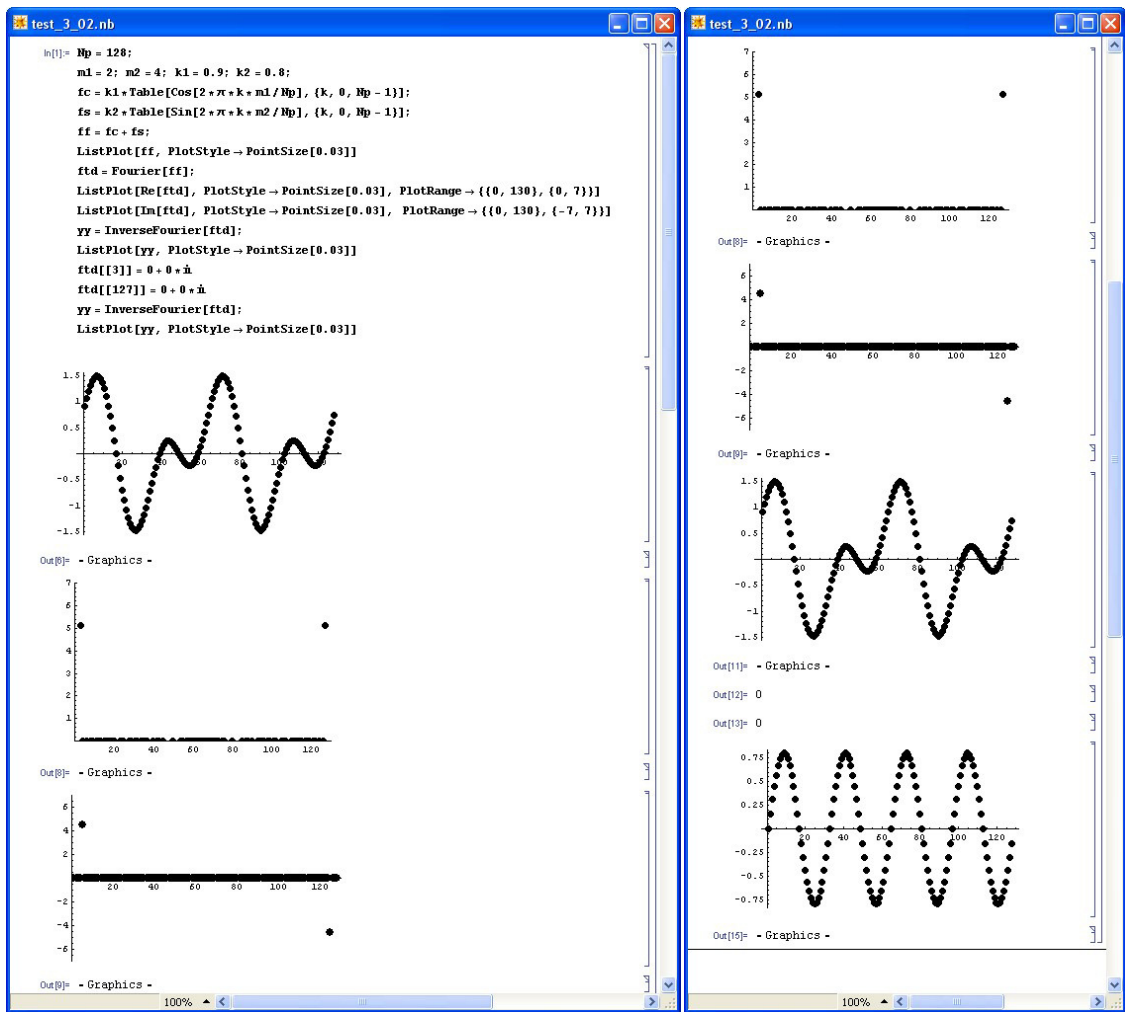


Рис. 3.2. Пример расчета и визуализации значений коэффициентов прямого и обратного ДПФ с фильтрацией одной из частот

Задание для самостоятельной работы

1. Задайте исходную функцию в виде суммы гармонического сигнала и случайных значений, генерируемых с помощью функции "Random[]", рассчитайте спектр ДПФ полученной функции. Оцените отношение сигнал-шум исходной функции, а также спектральную плотность сигнала и шума. Объясните полученный результат.

2. Задайте исходную функцию (сигнал) в виде суммы трех функций косинуса с различными коэффициентами m_1 , m_2 , m_3 , k_1 , k_2 , k_3 и случайными фазами в пределах от 0 до 2π . Выполните ДПФ, создайте цифровой фильтр, устраните одну из заданных частот, принимая во внимание наличие, в общем случае, как действительной, так и мнимой части коэффициентов ДПФ. Восстановите сигнал по обработанному спектру с использованием процедуры обратного ДПФ. Постройте график полученного сигнала.

4. Считывание данных из файла

Во многих случаях данные, поступающие с аналого-цифровых преобразователей, сохраняются в заданном файле. Для обработки таких данных их необходимо считать из файла в программную среду “Wolfram Mathematica”. Простейшим случаем является последовательная построчная запись чисел с плавающей запятой в текстовом файле. Пример записей в таком файле приведен на рис. 4.1:



Рис. 4.1. Пример записи исходных данных в текстовом файле

В среде “Mathematica 5” считывание данных из файла (D:\tmp\fff1.txt) можно осуществить следующим кодом:

```
str=OpenRead["D:\\tmp\\fff1.txt"]  
vADC= ReadList[str]  
Close[str]
```

(4.3)

Данные из файла «D:\tmp\fff1.txt» будут помещены в список *vADC*. Для контроля правильности считывания данных в конце программы выводятся два первых значения из списка *vADC*, соответствующие двум первым строкам в файле. Пример выполнения такой программы приведен на рис. 4.2. Числа в файле могут быть записаны в одну строку. Разделителем между числами может служить один или несколько пробелов (*Hex* код – 20, *Dec* – 32) или символ “Tab” (*Hex* код – 09).

```
Read_file_01.nb

In[1]:= str = OpenRead["D:\\tmp\\fff1.txt"]
vADC = ReadList[str]
Close[str]
vADC[[1]]
vADC[[2]]

Out[1]= InputStream[D:\\tmp\\fff1.txt, 5]
Out[2]= {1.2529, 0.6489, -0.0264, -0.1658, -0.9642, -1.}
Out[3]= D:\\tmp\\fff1.txt
Out[4]= 1.2529
Out[5]= 0.6489
```

Рис. 4.2. Пример программы для ввода данных из текстового файла

Однако в этом случае необходимо указать тип считываемых данных в качестве параметра функции `ReadList`:

$$vADC = \text{ReadList}[str, \text{Real}] \quad (4.4)$$

Рассмотрим пример считывания данных из файла и цифровой обработки сигнала. Используем для считывания из файла `D:\\tmp\\d1024.txt` программу, указанную выше (4.3). Построим график считанных данных с помощью функции `ListPlot`, выполним прямое ДПФ с использованием функции `Fourier` (рис. 4.3). Построим графики действительной и мнимой части всех коэффициентов спектра ДПФ (рис. 4.4.). Из полученных графиков следует, что действительные и мнимые части коэффициентов существенно отличаются от нуля только на начальном и конечном участке зависимости. Поэтому целесообразно построить эти графики в другом масштабе по оси абсцисс. Если известно, что сигнал представляет собой сумму полезного сигнала, синусоидальной (например, промышленной частоты) помехи и случайных шумов, то обработка такого сигнала упрощается. Исходя из полученных данных можно сделать предположение о том, что синусоидальной помехе соответствует четвертый элемент списка (мнимая

часть) коэффициентов ДПФ, что соответствует коэффициенту ДПФ с $n=3$. Исключим данную частоту из частотного спектра сигнала:

$$ftd[[4]]=0; ftd[[1022]]=0; \quad (4.5)$$

В программе это действие имеет комментарий «Фильтр 1». Произведем обратное ДПФ и выведем график восстановленного сигнала после первой фильтрации (рис. 4.5, Out[15]). Из графика видно, что устранение синусоидальной помехи дает возможность визуально определить форму полезного сигнала – меандр. Однако восстановленный сигнал имеет довольно большой уровень шума, соизмеримый с полезным сигналом. Так как шум имеет приблизительно одинаковую спектральную плотность во всем исследуемом диапазоне частот, а полезный сигнал является сравнительно низкочастотным, можно улучшить отношение сигнал-шум, применив фильтр низких частот. В данной программе он реализован следующим образом:

$$\begin{aligned} ftd[[513]]=0; \\ \text{For } [i=1, i<350, i++, ftd[[513-i]]=0; ftd[[513+i]]=0] \end{aligned} \quad (4.6)$$

Первая строка фрагмента (4.6) обнуляет действительную и мнимую часть 513-ого элемента списка *ftd*, соответствующего коэффициенту ДПФ в точке $N/2$. Вторая строка представляет собой цикл, обнуляющий 350 коэффициентов вправо и влево от 513-ого элемента списка *ftd*, т.е. высокочастотную часть спектра сигнала. Данный фильтр применен после фильтра №1, т.е. для ранее обработанного спектра *ftd*. В программе этот фильтр назван «Фильтр 2». Далее восстановим сигнал и выведем график получившейся зависимости (рис. 4.5, Out[20]). После второго фильтра сигнал имеет явно меньший уровень шума, чем после первого. Увеличим количество обнуляемых коэффициентов с 350 до 475. Фильтрацию осуществим с помощью фрагмента программы, аналогичной (4.6) – «Фильтр 3». Данный фрагмент можно использовать как после фильтра 2, так и заменив в цикле фильтра 2 количество обнуляемых коэффициентов. Результат обработки

сигнала с помощью фильтра 3 также приведен на рис. 4.5 (Out[24]). Сигнал, полученный после третьей фильтрации, имеет форму близкую к меандру и может использоваться для дальнейшей цифровой обработки или преобразования в двоичный код.

В некоторых случаях желательно иметь сигнал с нулевым средним значением, т.е. без постоянной составляющей. Для устранения постоянной составляющей данного сигнала («Фильтр 4») достаточно ввести в программу строку:

$$\text{ftd}[[1]]=0; \quad (4.7)$$

и далее выполнить обратное ДПФ. Результат выполнения программы также представлен на рис. 4.5 (Out[28]).

Если необходимо сохранить полученные данные, то запись в файл можно осуществить, используя следующий фрагмент программы:

```
str=OpenWrite["D:\\tmp\\out1024.txt"]
For[i=1, i<=1024, i++,Write[str,yy[[i]]]];
Close[str]
```

 (4.8)

Первая строка программы открывает поток *str* для записи в файл *D:\tmp\out1024.txt*, вторая – записывает числа из списка *yy* построчно с помощью цикла по переменной *i*, третья строка – закрывает поток *str*.

Необходимо отметить, что данный фрагмент программы записывает файл с пустой последней строкой (т.е. 1025-ой). Поэтому записываемый файл предназначен для считывания программами, отличными от рассмотренного выше примера, например, программой построения графиков “Origin”, приложениями Microsoft Office и др. При считывании данного файла с использованием фрагмента программы (3.3) в некоторых версиях программной среды “Wolfram Mathematica” количество считанных элементов списка может быть равно 1025.

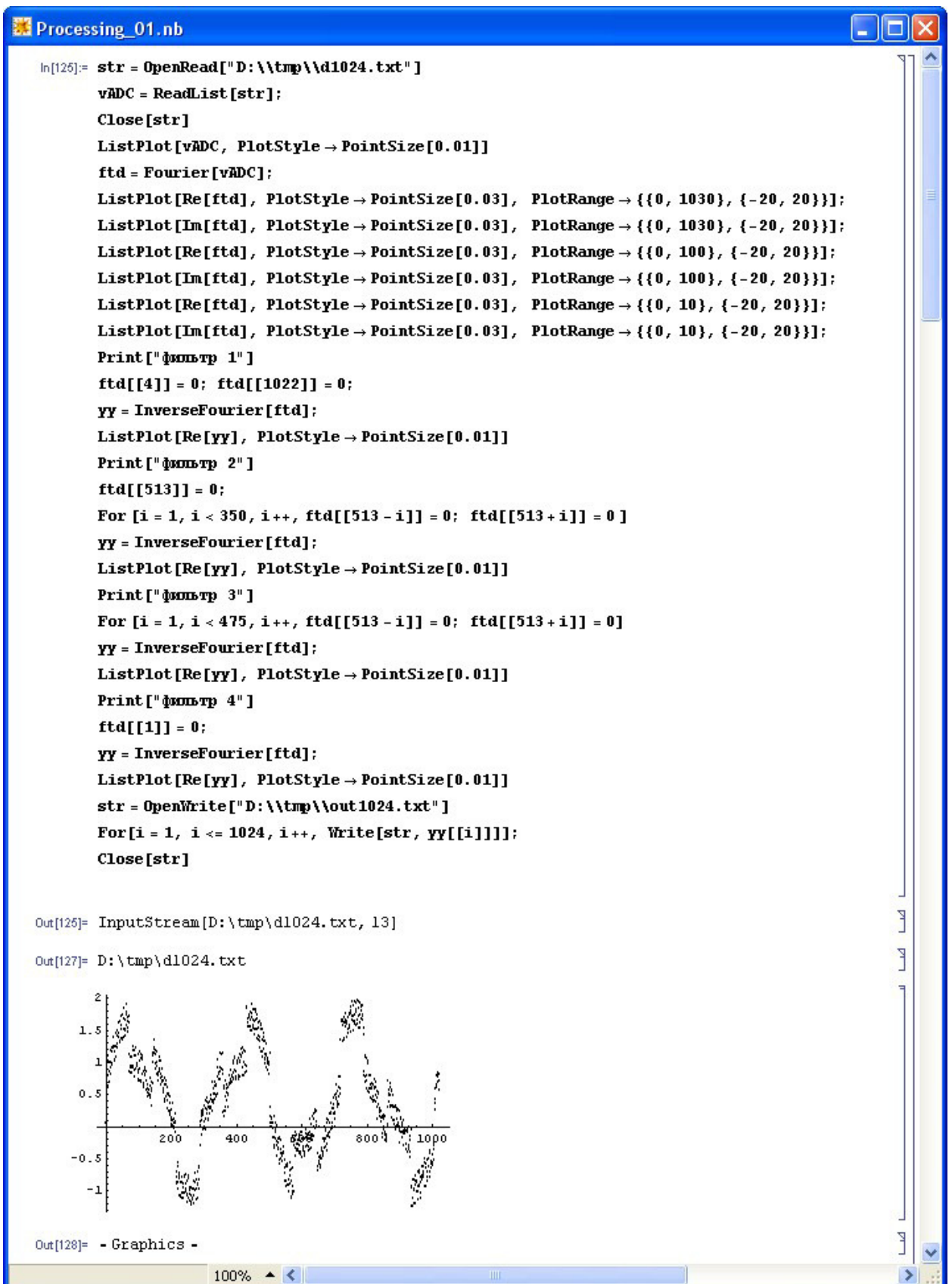


Рис. 4.3. Пример программы для считывания файла, выполнения дискретного преобразования Фурье, частотной фильтрации, обратного преобразования Фурье и записи полученных данных в файл с визуализацией сигналов и спектров

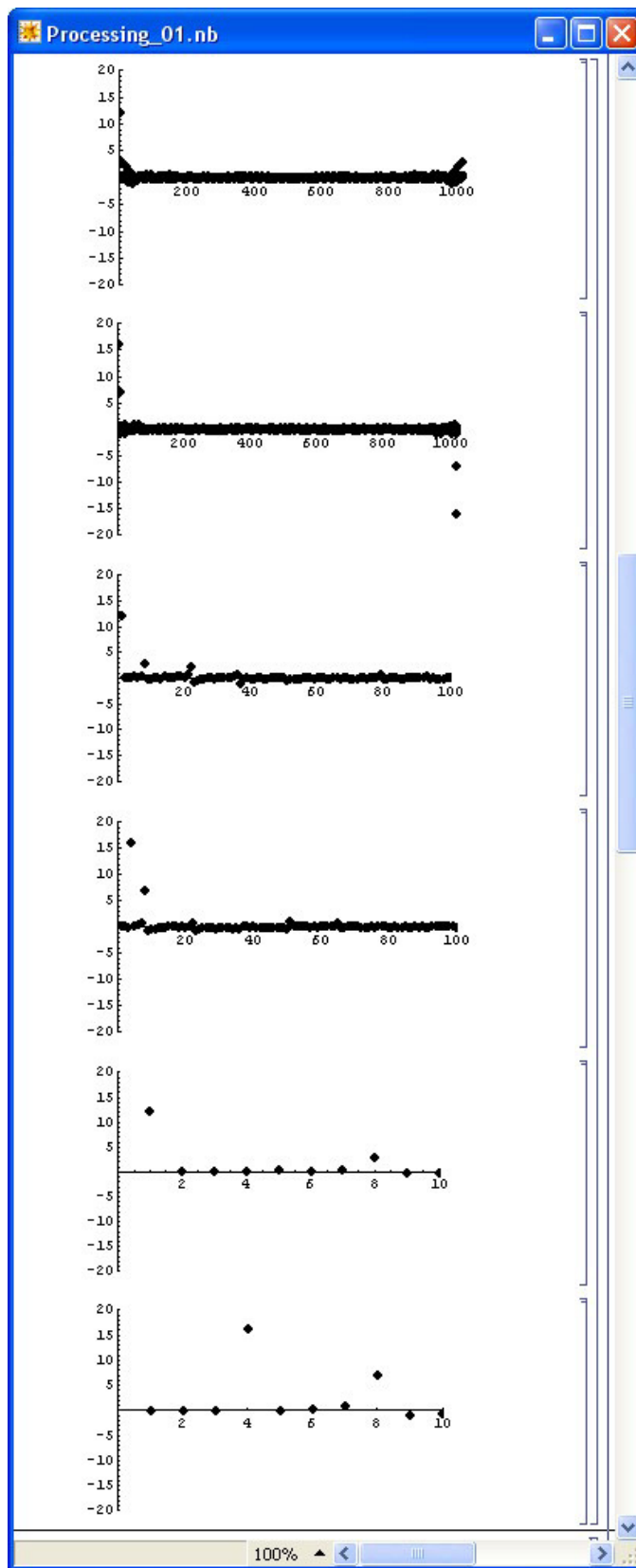


Рис. 4.4. Продолжение копии экранной формы к рис. 4.3. Действительная и мнимая части коэффициентов ДПФ, построенные в различном масштабе (по оси абсцисс – номер элемента в списке, начиная с 1, по оси ординат – значения коэффициентов)

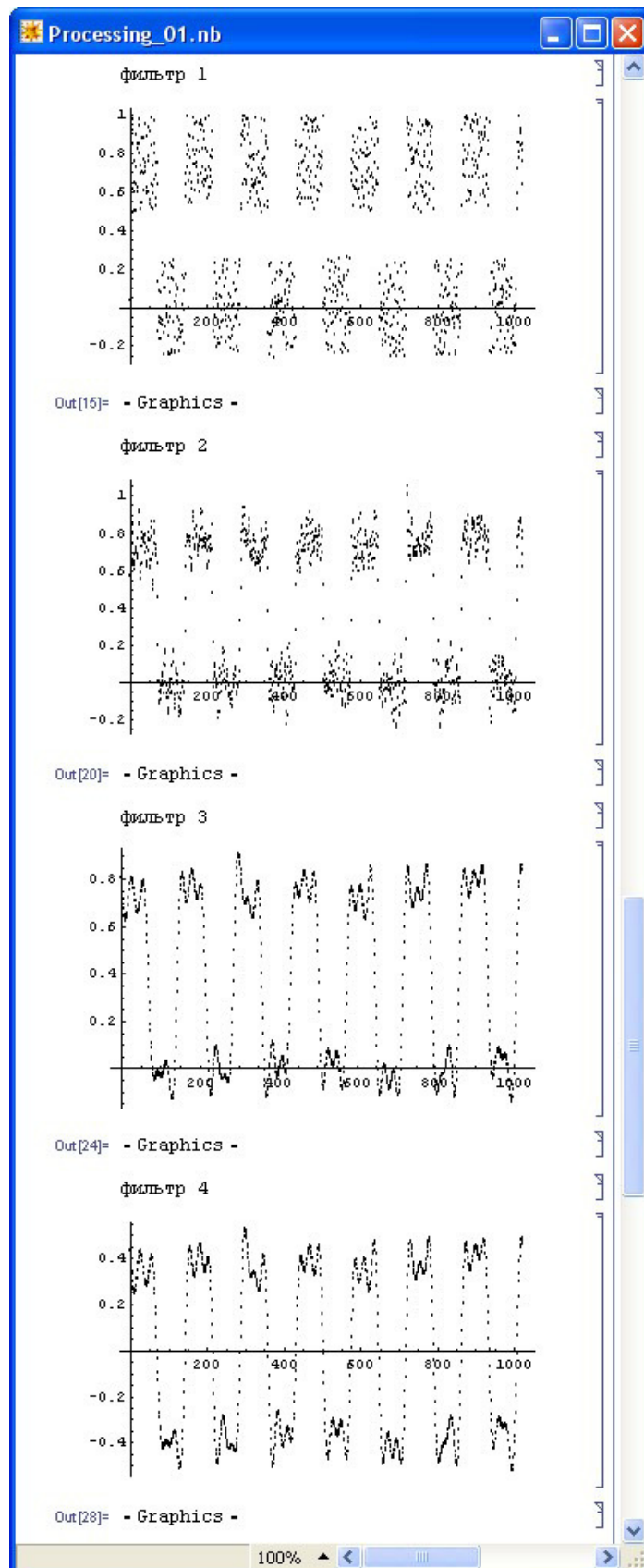


Рис. 4.5. Продолжение копии экранной формы к рис. 4.3. Восстановленный с помощью обратного ДПФ сигнал, после фильтров 1, 2, 3 и 4

Результат преобразования Фурье такого списка будут существенно отличаться от полученного выше. Возможны три способа устранения данного недостатка. Первый: удалить последнюю (пустую) строку из сохраненного файла вручную или программно. Второй – использовать для считывания функцию “RedList” с дополнительным параметром (*Real*) (см. 3.4). Целесообразно также контролировать количество считанных из файла в список элементов с помощью функции “Length”. Третий способ – использовать версию “Mathematica” не имеющую данной проблемы.

В рассмотренном примере фильтр низких частот является ступенчатым. В более общем случае частотные фильтры имеют более сложную частотную зависимость пропускания. Это позволяет улучшить фильтрацию сигналов в зависимости от поставленной задачи. Необходимо также отметить, что рассмотренный способ фильтрации является частным случаем широкого класса цифровых фильтров для обработки сигналов, подробно рассматриваемых в других учебных курсах. Подробное описание принципов построения частотных фильтров и их программная реализация, в частности, приведены в учебных пособиях [5-6].

Задание для самостоятельной работы

1. Напишите программу для считывания действительных чисел из текстового файла и выполнения дискретного преобразования Фурье. Количество чисел в файле должно быть не менее 16. Используя функцию “Length”, определите длину полученного списка. Проверьте выполнение основных свойств коэффициентов ДПФ, приводимых в Табл. 2.2, 2.3.
2. На основании полученных коэффициентов ДПФ, определите среднее значение элементов списка в исходном файле. Добавьте к исходном

данным синусоидальный сигнал с периодом, равным длине списка и с амплитудой, равной половине среднего значения элементов списка в исходном файле. Выполните ДПФ. Найдите коэффициенты ДПФ, соответствующие данному синусоидальному сигналу.

3. Создайте список из 16 элементов y_0 , соответствующий синусоидальному сигналу с периодом, равным длине списка. Выполните для всех элементов списка преобразование в соответствии с формулой: $y_1[[i]] = y_0[[i]] + 2.0 * (y_0[[i]])^3$, соответствующее нелинейному фильтру. Выведите на экран графики исходной (y_0) и преобразованной (y_1) функций. Рассчитайте коэффициент третьей гармоники сигнала.

5. Изучение алгоритма быстрого преобразования Фурье

Ранее показано, что для осуществления дискретного преобразования Фурье (ДПФ) необходимо выполнить $N \times N$ операций вычисления экспоненты. С учетом свойств симметрии и периодичности поворачивающих множителей, количество таких операций, а также операций сложения и умножения, может быть уменьшено, соответственно скорость расчетов – увеличена. Рассмотрим классический алгоритм быстрого преобразования Фурье (БПФ), называемый алгоритмом БПФ с прореживанием по времени. Возникновение этого термина обусловлено тем, что в большинстве случаев исходная функция является функцией от времени, а основное применение алгоритма БПФ – обработка сигналов.

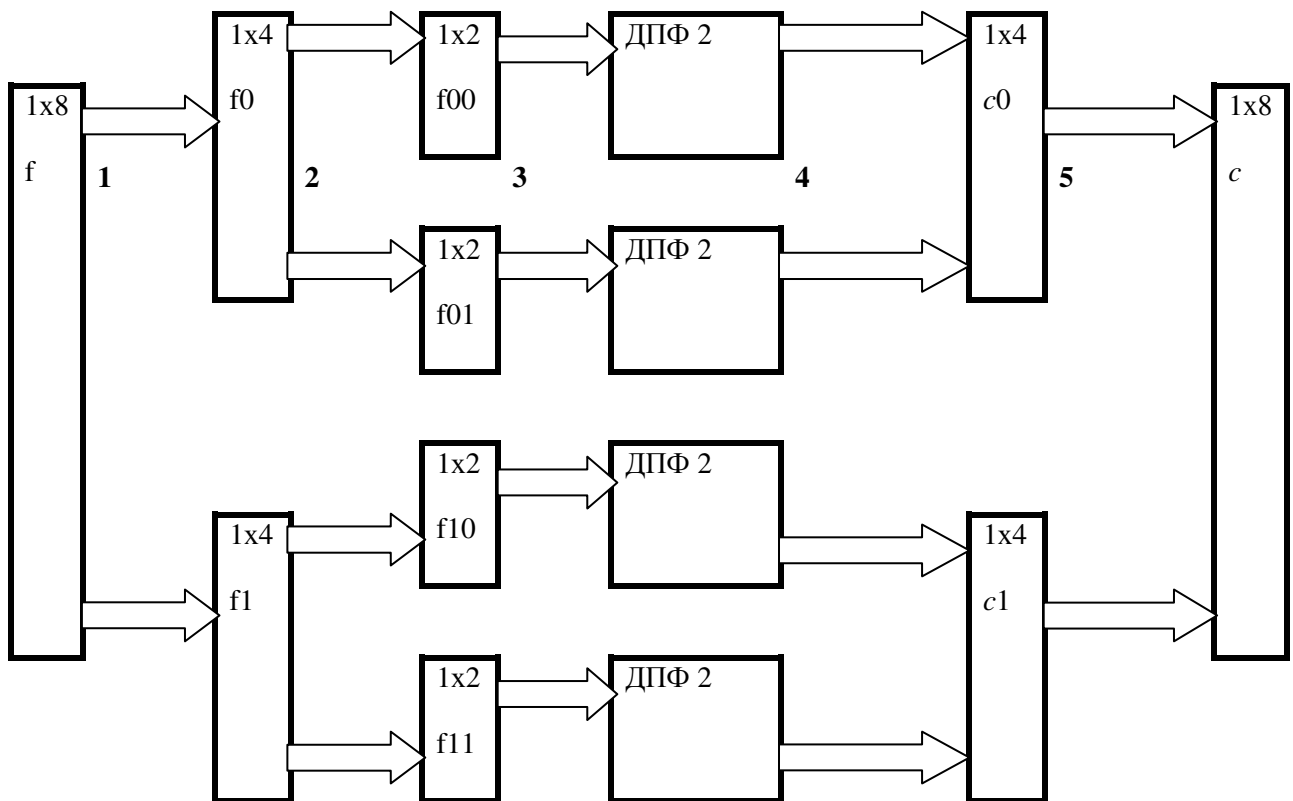


Рис. 5.1. Структурная схема алгоритма быстрого преобразования Фурье с прореживанием по времени по основанию 2 для $N=8$; операции 1-3 – прореживания, 4-5 – объединения, ДПФ 2 – двухточечные дискретные преобразования Фурье

Сущность данного алгоритма можно описать следующим образом (рис. 5.1). Исходная функция, заданная в виде дискретного набора f из N чисел (или исходный сигнал, содержащий N выборок) разделяется на две группы: одна из групп f_0 содержит четные элементы исходного набора, другая группа f_1 – нечетные:

$$f_0[k]=f[2k] \quad (5.1)$$

$$f_1[k]=f[2k+1] \quad (5.2)$$

Каждая из групп содержит $N/2$ элементов. Далее, каждая из этих двух групп делится на две подгруппы, образуя 4 подгруппы по $N/4$ элемента. Деление на подгруппы производится до достижения количества элементов в подгруппах, равного 2. Следовательно данный алгоритм применим для случая, когда количество точек N равно 2^m , где m - целое число. После разделения, для каждой подгруппы из подгрупп по 2 элемента производится вычисление ДПФ, также с использованием специального алгоритма (см. ниже). Далее, два полученных значения двухточечного ДПФ, также с применением специального алгоритма, объединяются попарно в подгруппы, содержащие по 4 элемента. Подгруппы по 4 элемента объединяются в подгруппы по 8 элементов и т.д. до достижения одной группы, состоящей из N элементов – искомого преобразования. На рис. 5.1 приведена структурная схема алгоритма для случая $N=8$.

Рассмотрим алгоритм более подробно. Обозначим коэффициенты преобразования ДПФ, т.е. спектральной плотности, как $c[n]$. Вычислим ДПФ от прореженной функции f :

$$c[n] = \sum_{k=0}^{N/2-1} f[2k]W_N^{2kn} + \sum_{k=0}^{N/2-1} f[2k+1]W_N^{(2k+1)n} \quad (5.3)$$

Выражение для коэффициентов c можно также записать в виде:

$$c[n] = \sum_{k=0}^{N/2-1} f[2k]W_N^{2kn} + W_N^n \sum_{k=0}^{N/2-1} f[2k+1]W_N^{2nk} \quad (5.4)$$

Принимая во внимание, что

$$W_N^{2kn} = W_{N/2}^{kn} \quad (5.5)$$

выражение (5.4) может быть представлено в виде:

$$c[n] = \sum_{k=0}^{N/2-1} f[2k]W_{N/2}^{kn} + W_N^n \sum_{k=0}^{N/2-1} f[2k+1]W_{N/2}^{nk} \quad (5.6)$$

То есть, в соответствии с обозначениями (5.1) и (5.2), выражение (5.6) является ДПФ двух подгрупп f_0, f_1 прореженной исходной функции. Тогда:

$$c[n] = c_0[n] + W_N^n c_1[n] \quad (5.7)$$

Из (5.7) следует, что зная значения элементов подгрупп $c_0[n]$ и $c_1[n]$, можно определить значение $c[n]$ полного спектра. Рассмотренное преобразование может быть произведено также для подгрупп $c_{00}, c_{01}, c_{10}, c_{11}$. В этом случае выражение (5.7) преобразуется к виду:

$$c_0[n] = c_{00}[n] + W_{N/2}^n c_{01}[n] \quad (5.8)$$

Однако при объединении двух подгрупп спектра ДПФ в одну группу возникает следующая проблема. Количество точек (элементов) в подгруппах в два раза меньше, чем в группе, в которую элементы этих подгрупп будут записаны. Поэтому определение значений элементов спектра $c[n]$ с $n > N/2$ (иногда называемой второй половиной спектра) требует специального рассмотрения.

Аналогично выражениям (5.3), (5.4), запишем выражение для $c[n+N/2]$ в виде:

$$c[n + N/2] = \sum_{k=0}^{N/2-1} f[2k] W_{N/2}^{(k+N/2)n} + W_N^{n+N/2} \sum_{k=0}^{N/2-1} f[2k+1] W_{N/2}^{(k+N/2)n} \quad (5.9)$$

Выполним преобразование поворотных множителей $W_{N/2}^{(k+N/2)n}$ и $W_N^{n+N/2}$:

$$W_{N/2}^{(k+N/2)n} = W_{N/2}^{kn} W_{N/2}^{nN/2} \quad (5.10)$$

$$W_N^{n+N/2} = W_N^n W_N^{N/2} \quad (5.11)$$

Так как для любого целого n :

$$W_{N/2}^{nN/2} = \exp\left(i \frac{2\pi n N/2}{N/2}\right) = 1, \quad (5.12)$$

$$W_N^{N/2} = \exp\left(i \frac{2\pi n N/2}{N}\right) = -1 \quad (5.13)$$

Из (5.10), (5.1), получаем:

$$W_{N/2}^{(k+N/2)n} = W_{N/2}^{kn} \quad (5.14)$$

$$W_N^{n+N/2} = -W_N^n \quad (5.15)$$

Тогда выражение (5.9) можно представить в виде:

$$c[n + N/2] = \sum_{k=0}^{N/2-1} f[2k] W_{N/2}^{kn} - W_N^n \sum_{k=0}^{N/2-1} f[2k+1] W_{N/2}^{kn} \quad (5.16)$$

С учетом введенных обозначений для подгрупп ДПФ, получаем:

$$c[n + N/2] = c0[n] - W_N^n c1[n] \quad (5.17)$$

Таким образом, используя выражения (5.8) и (5.17), объединив элементы подгрупп $c0$ и $c1$, можно получить искомую группу c , т.е. полный спектр

дискретного преобразования Фурье от исходной функции, заданной в виде набора f дискретных чисел.

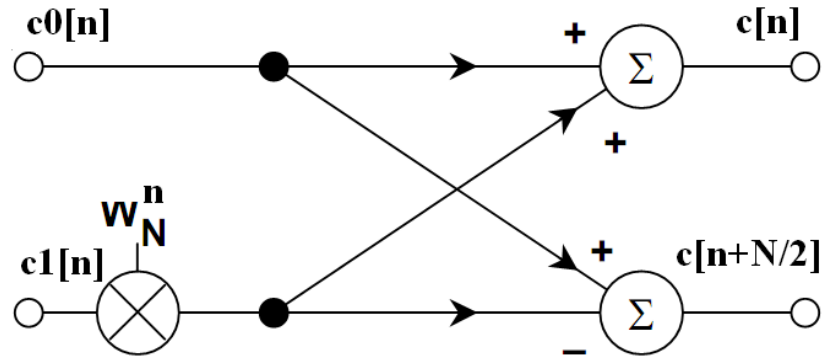


Рис. 5.2. Графическое изображение алгоритма объединения элементов подгрупп спектра («граф-бабочка»)

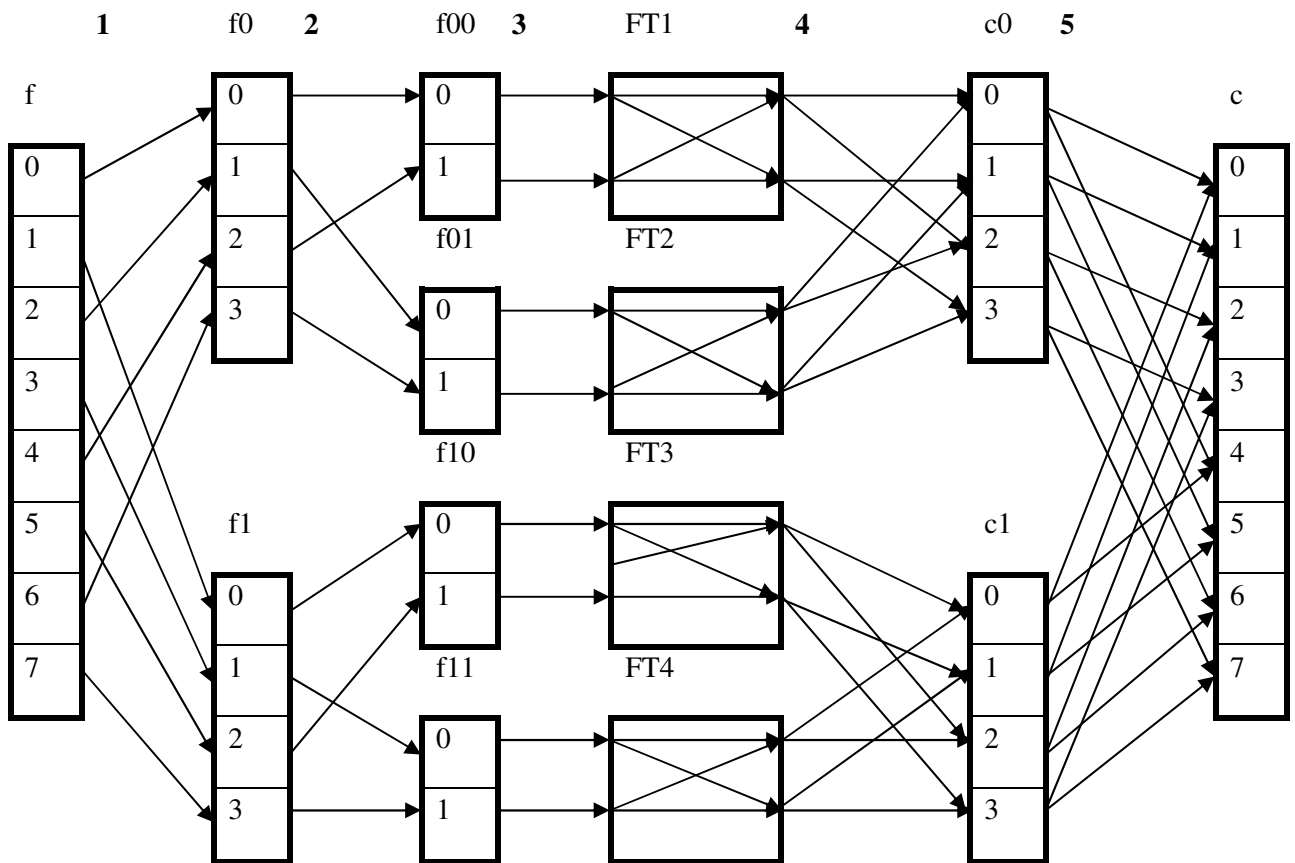


Рис. 5.3. Схематическое представление классического алгоритма быстрого преобразования Фурье для $N=8$ по основанию 2 с прореживанием по времени

Аналогично объединяем элементы подгрупп c_{00} , c_{01} в c_0 , c_{10} , c_{11} в c_1 . Если существуют подгруппы c_{000} , c_{001} и т.д., то они объединяются по аналогичному алгоритму. Графическое изображение алгоритма объединения элементов подгрупп спектра преобразований Фурье (в соответствии с формулами 5.7, 5.17) представлено на рис. 5.2. Общепринятое, но неофициальное название изображения данного алгоритма – «граф–бабочка».

Рассмотрим более подробно алгоритм БПФ для случая $N=8$, т.е. восьмиточечного быстрого преобразования Фурье (рис. 5.2). Пусть исходная функция (сигнал) f задана восемью точками (отсчетами): $f[0]$, $f[1]$, ..., $f[7]$. В среде “Mathematica” введем список исходных значений функции f с помощью функции “Table”. Нумерация элементов списка начинается с единицы, поэтому список содержит элементы $f[[1]]$, $f[[2]]$, ..., $f[[N]]$. Разделим элементы исходной функции на две подгруппы f_0 , f_1 следующим образом: четные элементы – $f[0]$, $f[2]$, $f[4]$, $f[6]$ сохраним в виде подгруппы f_0 : $f_0[1]$, $f_0[2]$, $f_0[3]$, $f_0[4]$, а нечетные – $f[1]$, $f[3]$, $f[5]$, $f[7]$ – в виде подгруппы f_1 : $f_1[1]$, $f_1[2]$, $f_1[3]$, $f_1[4]$. При программировании в среде “Mathematica” перед заполнением элементов списков f_0 и f_1 , необходимо эти списки создать (или выполнить одновременно создание и заполнение). Таким образом, формируются два списка по 4 элемента в каждом. Фрагмент программы, выполняющий указанное разделение, с учетом нумерации списков с единицы, может быть записан в следующем виде:

$$\begin{aligned}
 & f = \text{Table}[\text{Cos}[2 * \pi * k / Np], \{k, 0, Np - 1\}] & (5.1) \\
 & \text{ListPlot}[f] \\
 & f_0 = \{f[[1]], f[[3]], f[[5]], f[[7]]\} \\
 & f_1 = \{f[[2]], f[[4]], f[[6]], f[[8]]\}
 \end{aligned}$$

Допустим и иной способ заполнения списков f_0 и f_1 :

$$\begin{aligned}
 & f = \text{Table}[\text{Cos}[2 * \pi * k / Np], \{k, 0, Np - 1\}] & (5.2) \\
 & \text{ListPlot}[f]
 \end{aligned}$$

```

f0=Table[0,{k,1,8}]
f0[[1]]=f[[1]]
f0[[2]]=f[[3]]
f0[[3]]=f[[5]]
f0[[4]]=f[[7]]
f1=Table[0,{k,1,8}]
f1[[1]]=f[[2]]
f1[[2]]=f[[4]]
f1[[3]]=f[[6]]
f1[[4]]=f[[8]]

```

Далее выполним разделение элементов двух подгрупп (списков) на четыре подгруппы. Подгруппу f_0 разделим на f_{00} и f_{01} , подгруппу f_1 – на f_{10} и f_{11} . В программе это будет соответствовать разделению списков f_0 и f_1 на списки f_{00} , f_{01} и f_{10} , f_{11} . Аналогично предыдущему разделению на две подгруппы, разделим четные и нечетные элементы подгрупп:

$$\begin{aligned}
f_{00}[0] &= f_0[0]; f_{00}[1] = f_0[2] & (5.3) \\
f_{01}[0] &= f_0[1]; f_{01}[1] = f_0[3] \\
f_{10}[0] &= f_1[0]; f_{10}[1] = f_1[2] \\
f_{11}[0] &= f_1[1]; f_{11}[1] = f_1[3]
\end{aligned}$$

В результате разделения получаем 4 подгруппы по 2 элемента в каждой. Программно разделение на четыре подгруппы может быть выполнено так:

$$\begin{aligned}
f_{00} &= \{f_0[[1]], f_0[[3]]\} & (5.4) \\
f_{01} &= \{f_0[[2]], f_0[[4]]\} \\
f_{10} &= \{f_1[[1]], f_1[[3]]\} \\
f_{11} &= \{f_1[[2]], f_1[[4]]\}
\end{aligned}$$

Дискретное преобразование Фурье для случая двух точек требует специального рассмотрения. Общее выражение для коэффициентов ДПФ

$$c[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f[k] \exp(i \frac{2\pi}{N} kn) \quad (5.5)$$

или

$$c[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f[k] W_N^{kn} \quad (5.6)$$

принимая во внимание, что $W_2^0 = 1$, $W_2^1 = -1$, выражение (5.6), с учетом $N=2$, может быть приведено к виду:

$$c[0] = (f[0] + f[1]) / 2^{1/2} \quad (5.7)$$

$$c[1] = (f[0] - f[1]) / 2^{1/2} \quad (5.8)$$

То есть, вычисление преобразования Фурье (без учета нормировочного коэффициента $2^{1/2}$) сводится к сумме и разности двух значений преобразуемой функции. В среде “Mathematica” данное преобразование может быть выполнено следующим образом:

$$c[[1]] = (f[[1]] + f[[2]]) / 2^{1/2} \quad (5.9)$$

$$c[[2]] = (f[[1]] - f[[2]]) / 2^{1/2} \quad (5.10)$$

В качестве проверки полученных формул, выполним преобразование Фурье с использованием встроенной функции (Fourier) среды “Mathematica” (список а) и с использованием выражений (5.9)-(5.10) (список с). После выполнения преобразований сравним полученные результаты (рис. 5.4). В данном примере исходная функция f задана в виде двух произвольных комплексных чисел. Как следует из результатов вычислений, полученные списки а и с совпадают (рис. 5.4). То есть, полученные выражения (5.9) и (5.10) являются правильными.

```

test_Fourier_N2.nb

In[1]:= f = {0.7 + 0.1 * i, 5.0 + 2 * i}
c = {0, 0}
c[[1]] = 1 / Sqrt[2] * (f[[1]] + f[[2]])
c[[2]] = 1 / Sqrt[2] * (f[[1]] - f[[2]])
Print["-----"]
c
a = Fourier[f]

Out[1]= {0.7 + 0.1 i, 5. + 2 i}

Out[2]= {0, 0}

Out[3]= 4.03051 + 1.48492 i

Out[4]= -3.04056 - 1.3435 i

-----

Out[6]= {4.03051 + 1.48492 i, -3.04056 - 1.3435 i}

Out[7]= {4.03051 + 1.48492 i, -3.04056 - 1.3435 i}

```

Рис. 5.4. Сравнение коэффициентов дискретного преобразования Фурье для $N=2$, полученных двумя различными способами

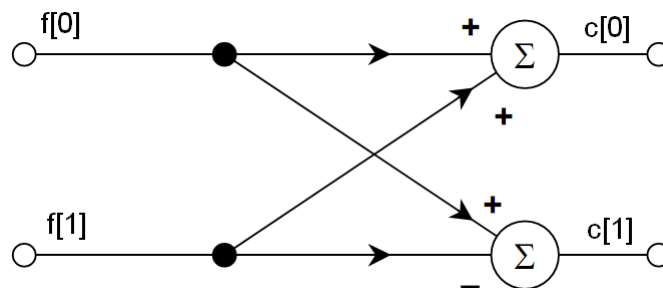


Рис. 5.5. Схематичное изображение алгоритма ДПФ для $N=2$ (в соответствии с формулами 5.7, 5.8, нормировочный множитель $1/2^{1/2}$ не учитывается)

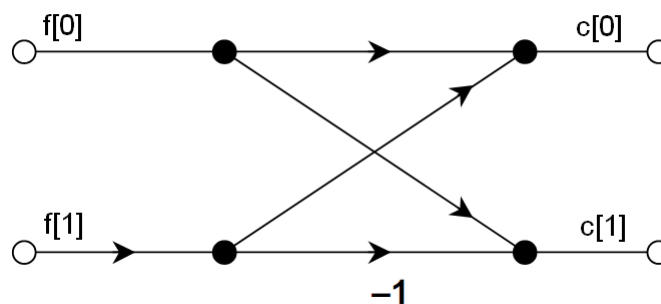


Рис. 5.6. Упрощенное схематичное изображение алгоритма ДПФ для $N=2$.

Схематичное изображение алгоритма выполнения дискретного преобразования Фурье для $N=2$ приведено на рис. 5.5. Упрощенный вид схемы алгоритма ДПФ представлен на рис. 5.6.

При рассмотрении алгоритма быстрого преобразования Фурье для $N=8$, два полученных коэффициента преобразования Фурье для $N=2$ целесообразно обозначить не как c , а как c_{00} , по аналогии с разделенной на подгруппы исходной функцией. Нормировку (\sqrt{N}) также целесообразно учесть однократно после вычисления значений всех элементов списка c , т.е. после выполнения всего рассматриваемого алгоритма БПФ. Тогда строки программы (5.9), (5.10) для первой подгруппы будут иметь вид:

$$c_{00}[[1]]=(f_{00}[[1]]+f_{00}[[2]]) \quad (5.11)$$

$$c_{00}[[2]]=(f_{00}[[1]]-f_{00}[[2]]) \quad (5.12)$$

Элементы остальных подгрупп – c_{01} , c_{10} , c_{11} вычисляются аналогично, в соответствии со схемой, приводимой на рис. 5.3. В результате ДПФ четырех подгрупп с двумя элементами в каждой подгруппе (f_{00} , f_{01} , f_{10} , f_{11}) получаем четыре подгруппы (c_{00} , c_{01} , c_{10} , c_{11}) также с $N=2$.

Далее необходимо объединить 4 подгруппы с двумя элементами в каждой в 2 подгруппы с четырьмя элементами в каждой, а далее – две полученные подгруппы в одну группу с восемью элементами. Это и будет искомая группа элементов c , соответствующая преобразованию Фурье от функции f . Такое объединение элементов производится на основании свойств дискретного преобразования Фурье.

Пример программы, реализующий классический алгоритм быстрого преобразования Фурье, приведен на рис. 5.7. Результат выполнения программы (начиная с 20-й строки) – на рис. 5.8. Назначение каждой строки программы дано в Табл. 5.1.

```

FFT_01.nb
In[1]:= Np = 8
W[Nn_, m_] := Exp[2 * π * i * m / Nn]
f = Table[Cos[2 * π * k / Np], {k, 0, Np - 1}]
ListPlot[f]
f0 = {f[[1]], f[[3]], f[[5]], f[[7]]}
f1 = {f[[2]], f[[4]], f[[6]], f[[8]]}
f00 = {f0[[1]], f0[[3]]}
f01 = {f0[[2]], f0[[4]]}
f10 = {f1[[1]], f1[[3]]}
f11 = {f1[[2]], f1[[4]]}
c00 = {0, 0}
c01 = {0, 0}
c10 = {0, 0}
c11 = {0, 0}
c00[[1]] = f00[[1]] + f00[[2]]
c00[[2]] = f00[[1]] - f00[[2]]
c01[[1]] = f01[[1]] + f01[[2]]
c01[[2]] = f01[[1]] - f01[[2]]
c10[[1]] = f10[[1]] + f10[[2]]
c10[[2]] = f10[[1]] - f10[[2]]
c11[[1]] = f11[[1]] + f11[[2]]
c11[[2]] = f11[[1]] - f11[[2]]
c0 = {0, 0, 0, 0}
c1 = {0, 0, 0, 0}
c0[[1]] = c00[[1]] + W[4, 0] * c01[[1]]
c0[[2]] = c00[[2]] + W[4, 1] * c01[[2]]
c0[[3]] = c00[[1]] - W[4, 0] * c01[[1]]
c0[[4]] = c00[[2]] - W[4, 1] * c01[[2]]
c1[[1]] = c10[[1]] + W[4, 0] * c11[[1]]
c1[[2]] = c10[[2]] + W[4, 1] * c11[[2]]
c1[[3]] = c10[[1]] - W[4, 0] * c11[[1]]
c1[[4]] = c10[[2]] - W[4, 1] * c11[[2]]
c = {0, 0, 0, 0, 0, 0, 0, 0}
c[[1]] = c0[[1]] + W[8, 0] * c1[[1]]
c[[2]] = c0[[2]] + W[8, 1] * c1[[2]]
c[[3]] = c0[[3]] + W[8, 2] * c1[[3]]
c[[4]] = c0[[4]] + W[8, 3] * c1[[4]]
c[[5]] = c0[[1]] - W[8, 0] * c1[[1]]
c[[6]] = c0[[2]] - W[8, 1] * c1[[2]]
c[[7]] = c0[[3]] - W[8, 2] * c1[[3]]
c[[8]] = c0[[4]] - W[8, 3] * c1[[4]]
normc = c / Sqrt[8]
Print["-----"]
N[normc]
a = Fourier[f]

```

Рис. 5.7. Пример реализации алгоритма быстрого преобразования Фурье

```

FFT_01.nb

Out[20]=  $\sqrt{2}$ 
Out[21]= 0
Out[22]=  $-\sqrt{2}$ 
Out[23]= {0, 0, 0, 0}
Out[24]= {0, 0, 0, 0}
Out[25]= 0
Out[26]= 2
Out[27]= 0
Out[28]= 2
Out[29]= 0
Out[30]=  $(1 - i) \sqrt{2}$ 
Out[31]= 0
Out[32]=  $(1 + i) \sqrt{2}$ 
Out[33]= {0, 0, 0, 0, 0, 0, 0, 0}
Out[34]= 0
Out[35]=  $2 + (1 - i) \sqrt{2} e^{\frac{i\pi}{4}}$ 
Out[36]= 0
Out[37]=  $2 + (1 + i) \sqrt{2} e^{\frac{3i\pi}{4}}$ 
Out[38]= 0
Out[39]=  $2 - (1 - i) \sqrt{2} e^{\frac{i\pi}{4}}$ 
Out[40]= 0
Out[41]=  $2 - (1 + i) \sqrt{2} e^{\frac{3i\pi}{4}}$ 
Out[42]=  $\left\{ 0, \frac{2 + (1 - i) \sqrt{2} e^{\frac{i\pi}{4}}}{2\sqrt{2}}, 0, \frac{2 + (1 + i) \sqrt{2} e^{\frac{3i\pi}{4}}}{2\sqrt{2}}, \right.$ 
 $\left. 0, \frac{2 - (1 - i) \sqrt{2} e^{\frac{i\pi}{4}}}{2\sqrt{2}}, 0, \frac{2 - (1 + i) \sqrt{2} e^{\frac{3i\pi}{4}}}{2\sqrt{2}} \right\}$ 
-----
Out[44]= {0., 1.41421 - 5.55112 × 10-17 i, 0., 0. + 5.55112 × 10-17 i,
0., 0. + 5.55112 × 10-17 i, 0., 1.41421 - 5.55112 × 10-17 i}
Out[46]= {0. + 0. i, 1.41421 + 0. i, 0. + 0. i, -9.66741 × 10-17 + 0. i,
0. + 0. i, -9.66741 × 10-17 + 0. i, 0. + 0. i, 1.41421 + 0. i}

```

Рис. 5.8. Результат выполнения программы, приводимой на рис. 5.7

Таблица 5.1.

№	Строка программы	Назначение
1	$Np=8$	Задаёт количество точек преобразования, равное 8
2	$W[Nn_m_]:=Exp[2*\pi*i*m/Nn]$	Задаёт определение поворотного множителя
3	$f=Table[Cos[2*\pi*k/Np],\{k,0,Np-1\}]$	Формирует список f из Np элементов и заполняет его в соответствии с указанной формулой
4	ListPlot[f]	Выводит график функции f , заданной в виде списка
5	$f0=\{f[[1]],f[[3]],f[[5]],f[[7]]\}$	Формирует список f0 из нечетных элементов списка f
6	$f1=\{f[[2]],f[[4]],f[[6]],f[[8]]\}$	Формирует список f1 из четных элементов списка f
7	$f00=\{f0[[1]],f0[[3]]\}$	Формирует список f00 из нечетных элементов списка f0
8	$f01=\{f0[[2]],f0[[4]]\}$	Формирует список f01 из четных элементов списка f00
9	$f10=\{f1[[1]],f1[[3]]\}$	Формирует список f10 из нечетных элементов списка f1
10	$f11=\{f1[[2]],f1[[4]]\}$	Формирует список f11 из четных элементов списка f1
11	$c00=\{0,0\}$	Создает список c00 из двух нулевых элементов
12	$c01=\{0,0\}$	Создает список c01 из двух нулевых элементов
13	$c10=\{0,0\}$	Создает список c10 из двух нулевых элементов

		элементов
14	$c11=\{0,0\}$	Создает список c11 из двух нулевых элементов
15	$c00[[1]]=f00[[1]]+f00[[2]]$	Рассчитывает значение ДПФ (N=2) для нечетного элемента списка c00
16	$c00[[2]]=f00[[1]]-f00[[2]]$	Рассчитывает значение ДПФ (N=2) для четного элемента списка c00
17	$c01[[1]]=f01[[1]]+f01[[2]]$	Рассчитывает значение ДПФ (N=2) для нечетного элемента списка c01
18	$c01[[2]]=f01[[1]]-f01[[2]]$	Рассчитывает значение ДПФ (N=2) для четного элемента списка c01
19	$c10[[1]]=f10[[1]]+f10[[2]]$	Рассчитывает значение ДПФ (N=2) для нечетного элемента списка c10
20	$c10[[2]]=f10[[1]]-f10[[2]]$	Рассчитывает значение ДПФ (N=2) для четного элемента списка c10
21	$c11[[1]]=f11[[1]]+f11[[2]]$	Рассчитывает значение ДПФ (N=2) для нечетного элемента списка c11
22	$c11[[2]]=f11[[1]]-f11[[2]]$	Рассчитывает значение ДПФ (N=2) для четного элемента списка c11
23	$c0=\{0,0,0,0\}$	Создает список c0 из четырех нулевых элементов
24	$c1=\{0,0,0,0\}$	Создает список c1 из четырех нулевых элементов
25	$c0[[1]]=c00[[1]]+W[4,0]*c01[[1]]$	Рассчитывает значение элемента списка c0[[1]] в соответствии с рассмотренным алгоритмом
26	$c0[[2]]=c00[[2]]+W[4,1]*c01[[2]]$	Рассчитывает значение элемента списка c0[[2]] в соответствии с рассмотренным

		алгоритмом
27	$c0[[3]] = c00[[1]] - W[4,0] * c01[[1]]$	Рассчитывает значение элемента списка c0[[3]] в соответствии с рассмотренным алгоритмом
28	$c0[[4]] = c00[[2]] - W[4,1] * c01[[2]]$	Рассчитывает значение элемента списка c0[[4]] в соответствии с рассмотренным алгоритмом
29	$c1[[1]] = c10[[1]] + W[4,0] * c11[[1]]$	Рассчитывает значение элемента списка c1[[1]] в соответствии с рассмотренным алгоритмом
30	$c1[[2]] = c10[[2]] + W[4,1] * c11[[2]]$	Рассчитывает значение элемента списка c1[[2]] в соответствии с рассмотренным алгоритмом
31	$c1[[3]] = c10[[1]] - W[4,0] * c11[[1]]$	Рассчитывает значение элемента списка c1[[3]] в соответствии с рассмотренным алгоритмом
32	$c1[[4]] = c10[[2]] - W[4,1] * c11[[2]]$	Рассчитывает значение элемента списка c1[[4]] в соответствии с рассмотренным алгоритмом
33	$c = \{0,0,0,0,0,0,0,0\}$	Создает список с из восьми нулевых элементов
34	$c[[1]] = c0[[1]] + W[8,0] * c1[[1]]$	Рассчитывает значение элемента списка c[[1]] в соответствии с рассмотренным алгоритмом
35	$c[[2]] = c0[[2]] + W[8,1] * c1[[2]]$	Рассчитывает значение элемента списка c[[2]] в соответствии с рассмотренным алгоритмом
36	$c[[3]] = c0[[3]] + W[8,2] * c1[[3]]$	Рассчитывает значение элемента списка c[[3]] в соответствии с рассмотренным алгоритмом

37	$c[[4]]=c0[[4]]+W[8,3]*c1[[4]]$	Рассчитывает значение элемента списка c[[4]] в соответствии с рассмотренным алгоритмом
38	$c[[5]]=c0[[1]]-W[8,0]*c1[[1]]$	Рассчитывает значение элемента списка c[[5]] в соответствии с рассмотренным алгоритмом
39	$c[[6]]=c0[[2]]-W[8,1]*c1[[2]]$	Рассчитывает значение элемента списка c[[6]] в соответствии с рассмотренным алгоритмом
40	$c[[7]]=c0[[3]]-W[8,2]*c1[[3]]$	Рассчитывает значение элемента списка c[[7]] в соответствии с рассмотренным алгоритмом
41	$c[[8]]=c0[[4]]-W[8,3]*c1[[4]]$	Рассчитывает значение элемента списка c[[8]] в соответствии с рассмотренным алгоритмом
42	$normc=c/Sqrt[8]$	Создает список normc из нормированных значений элементов списка c
43	Print["-----"]	Выводит на экран разделительную линию
44	N[normc]	Выводит значения всех элементов списка normc , соответствующего результату вычислений с использованием рассмотренного алгоритма быстрого преобразования Фурье
45	a=Fourier[f]	Выводит значения всех элементов списка a , соответствующего преобразованию Фурье, выполненного с использованием функции Fourier программной среды Mathematica

Алгоритм быстрого обратного преобразования Фурье аналогичен алгоритму прямого БПФ. Как и при выполнении прямого преобразования, при выполнении обратного преобразования производится разделение четных и нечетных элементов списка значений спектральной плотности сигнала C . Данная операция называется прореживание по частоте (или прореживание по спектру). Из одного списка C с количеством элементов N (где N может быть представлено как 2^m , $m=2, 3, 4, \dots$ – целое число) формируются два списка $C0$ и $C1$ с количеством элементов $N/2$. Из списков $C0$, $C1$ формируются списки $C00$, $C01$, $C10$, $C11$ с количеством элементов $N/4$ и т.д. до достижения количества элементов в списках равного 2. Далее, по аналогии с алгоритмом БПФ, производится двухточечное преобразование Фурье с последующим последовательным объединением списков с увеличением количества элементов в списке в 2, 4, 8 и т.д. раз, до достижения количества элементов в списке равного N . В результате обратного БПФ список C преобразуется в список f , соответствующий восстановленному сигналу.

Формально для реализации алгоритма обратного БПФ можно воспользоваться программой, приводимой на рис. 5.7 и в Табл. 5.1, внося небольшие изменения. Например, достаточно заменить сформированный список f в рассмотренном ранее примере, на список, заполненный нулями (5.13) и задать два любых ненулевых значения в соответствии с правилами (Табл. 2.2, 2.3) – (5.14, 5.15):

$$f = \text{Table}[0, \{k, 0, Np-1\}] \quad (5.13)$$

$$f[[2]] = 1 \quad (5.14)$$

$$f[[8]] = 1, \quad (5.15)$$

а функцию $\text{Fourier}[f]$ в конце программы (в Табл. 5.1 – строка 45) заменить на $\text{InverseFourier}[f]$. Если список f рассматривать как список, соответствующий спектру сигнала, а C – как список, соответствующий восстановленному сигналу, то рассмотренный ранее алгоритм и его программная реализация в

этом случае будет выполнять обратное быстрое преобразование Фурье. Однако для того, чтобы сохранить принятые в пособии обозначения сигнала (f) и его спектра (c), целесообразно в программе обратного БПФ заменить условные обозначения. Копия экранной формы программы, реализующей указанный пример, приведена на рис. 5.9, результат выполнения программы – на рис. 5.10. Из данных, приводимых на рис. 5.10 следует, что результаты выполнения обратного БПФ с использованием описанного выше алгоритма и с применением функции `InverseFourier` программой среды “Mathematica” практически совпадают. Отличия полученных значений восстановленного сигнала пренебрежимо мало (менее 10^{-16}) и обусловлено накоплением ошибки в процессе расчета при операциях с плавающей запятой.

```

Inverse_FFT_01.nb
In[304]:= Np = 8
W[Nn_, m_] := Exp[2 * π * i * m / Nn]
c = Table[0, {k, 0, Np - 1}]
c[[2]] = 1
c[[8]] = 1
c0 = {c[[1]], c[[3]], c[[5]], c[[7]]}
c1 = {c[[2]], c[[4]], c[[6]], c[[8]]}
c00 = {c0[[1]], c0[[3]]}
c01 = {c0[[2]], c0[[4]]}
c10 = {c1[[1]], c1[[3]]}
c11 = {c1[[2]], c1[[4]]}
f00 = {0, 0}
f01 = {0, 0}
f10 = {0, 0}
f11 = {0, 0}
f00[[1]] = c00[[1]] + c00[[2]]
f00[[2]] = c00[[1]] - c00[[2]]
f01[[1]] = c01[[1]] + c01[[2]]
f01[[2]] = c01[[1]] - c01[[2]]
f10[[1]] = c10[[1]] + c10[[2]]
f10[[2]] = c10[[1]] - c10[[2]]
f11[[1]] = c11[[1]] + c11[[2]]
f11[[2]] = c11[[1]] - c11[[2]]
f0 = {0, 0, 0, 0}
f1 = {0, 0, 0, 0}
f0[[1]] = f00[[1]] + W[4, 0] * f01[[1]]
f0[[2]] = f00[[2]] + W[4, 1] * f01[[2]]
f0[[3]] = f00[[1]] - W[4, 0] * f01[[1]]
f0[[4]] = f00[[2]] - W[4, 1] * f01[[2]]
f1[[1]] = f10[[1]] + W[4, 0] * f11[[1]]
f1[[2]] = f10[[2]] + W[4, 1] * f11[[2]]
f1[[3]] = f10[[1]] - W[4, 0] * f11[[1]]
f1[[4]] = f10[[2]] - W[4, 1] * f11[[2]]
f = {0, 0, 0, 0, 0, 0, 0, 0}
f[[1]] = f0[[1]] + W[8, 0] * f1[[1]]
f[[2]] = f0[[2]] + W[8, 1] * f1[[2]]
f[[3]] = f0[[3]] + W[8, 2] * f1[[3]]
f[[4]] = f0[[4]] + W[8, 3] * f1[[4]]
f[[5]] = f0[[1]] - W[8, 0] * f1[[1]]
f[[6]] = f0[[2]] - W[8, 1] * f1[[2]]
f[[7]] = f0[[3]] - W[8, 2] * f1[[3]]
f[[8]] = f0[[4]] - W[8, 3] * f1[[4]]
normf = f / Sqrt[8]
Print["-----"]
N[normf]
a = InverseFourier[c]

```

Рис. 5.9. Пример реализации алгоритма обратного быстрого преобразования Фурье

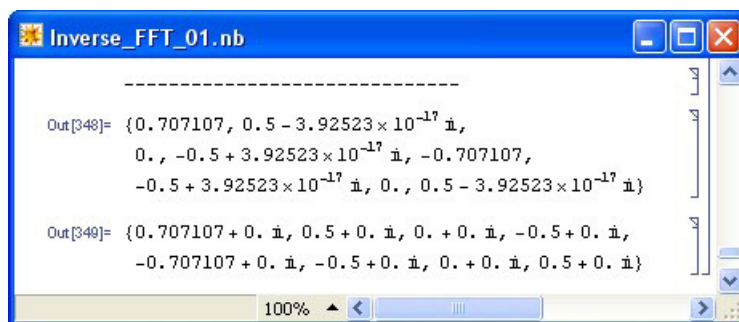


Рис. 5.10. Результат выполнения обратного БПФ для примера, представленного на рис. 5.8

Задание для самостоятельной работы и контрольные вопросы

1. В чем основное преимущество быстрого преобразования Фурье по сравнению с дискретным преобразованием Фурье?
2. Начертите схему алгоритма 16-ти точечного БПФ с прореживанием по времени по основанию 2.
3. Определите значения коэффициентов поворота $\text{Re}(W_N^{kn})$ и $\text{Im}(W_N^{kn})$ при $N=16$ для всех возможных значениях k (аналогично значениям, приводимым в таблицах 2.3 и 2.4) при $n=4$ и $n=5$.
4. Какие из коэффициентов поворота $\text{Re}(W_{16}^{kn})$ и $\text{Im}(W_{16}^{kn})$ имеют нулевые значения?
5. Напишите программу для реализации рассмотренного алгоритма быстрого преобразования Фурье для случая 16-ти точечного преобразования ($N=16$).
6. Напишите программу для реализации алгоритма обратного быстрого преобразования Фурье для случая 16-ти точечного преобразования ($N=16$).

Библиографический список

1. Фихтенгольц, Г.М. Курс дифференциального и интегрального исчисления. Т. 3 / Г.М. Фихтенгольц. – М.: Физматлит, 2002. – 728 С.
2. Фихтенгольц, Г.М. Основы математического анализа. Учебник. Ч. 2 / Г.М. Фихтенгольц. – СПб: Лань, 2008 – 464 С.
3. Корн, Г. Справочник по математике (для научных работников и инженеров) / Г. Корн, Т. Корн. – М.: Наука, 1977. – 832 С.
4. Бронштейн, И.Н. Справочник по математике. / И.Н. Бронштейн, К.А. Семендяев. – М.: Наука, 1964. – 608 С.
5. Сергиенко, А.Б. Цифровая обработка сигналов: учеб. пособие. / А.Б. Сергиенко. – СПб.: БХВ–Петербург, 2011. – 768 С.
6. Куприянов, М.С. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования. / М.С. Куприянов, Б.Д. Матюшкин. – СПб.: Политехника, 2002. – 592 С.

Оглавление

	Введение	3
1	Ряды Фурье, интеграл Фурье, преобразования Фурье. Общие сведения	5
2	Основные свойства дискретного преобразования Фурье	16
3	Обратное дискретное преобразование Фурье	29
4	Считывание данных из файла	33
5	Изучение алгоритма быстрого преобразования Фурье	42
12	Библиографический список	62