

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА  
УКРАИНЫ  
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
“ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**к выполнению лабораторной работы**  
**“Определение относительного изменения плотности вещества при кристалли-**  
**зации аморфной пленки в электронном микроскопе”**  
**по курсу “Электронная микроскопия и электронография”**  
**для студентов физико-технического, механико-технологического и**  
**инженерно-физического факультетов**

Харьков 2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА  
УКРАИНЫ  
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
“ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
к выполнению лабораторной работы  
“Определение относительного изменения плотности вещества при кристаллизации  
аморфной пленки в электронном микроскопе”  
по курсу “Электронная микроскопия и электронография”  
для студентов физико-технического, механико-технологического и  
инженерно-физического факультетов

Утверждено  
редакционно-издательским  
советом университета,  
протокол № 1 от 23.06.2011 года

**Методические указания** к выполнению лабораторной работы “Определение относительного изменения плотности вещества при кристаллизации аморфной пленки в электронном микроскопе” по курсу “Электронная микроскопия и электронография” для студентов физико-технического, механико-технологического и инженерно-физического факультетов / Составители – А.Г. Багмут, И.А. Багмут, Г.П. Николайчук и др. – Харьков: НТУ “ХПИ”, 2011.– 36 с.

Составители:      А.Г. Багмут  
                              И.А. Багмут  
                              Г.П. Николайчук

Рецензент            С.В. Малыхин

Кафедра теоретической и экспериментальной физики

## 1. ЦЕЛЬ РАБОТЫ

Определить относительное изменение плотности вещества при кристаллизации аморфной пленки, вызванной ее нагревом в электронном микроскопе.

## 2. ОБЩИЕ ПОЛОЖЕНИЯ

Фазовый переход (фазовое превращение) – скачкообразное изменение физических свойств при непрерывном изменении внешних параметров. Различают фазовые переходы двух родов. При фазовом переходе I рода скачком меняются такие термодинамические характеристики вещества, как плотность, концентрация компонентов и др. Это широко распространенное в природе явление. К фазовым переходам I рода относятся: испарение и конденсация, плавление и затвердевание, сублимация и конденсация в твердую фазу. Некоторые структурные переходы в твердых телах также являются фазовыми превращениями I рода, например образование мартенсита в сплаве Fe – C. В чистых сверхпроводниках достаточно сильное магнитное поле вызывает фазовое превращение I рода из сверхпроводящего в нормальное состояние. При фазовом переходе I рода в единице массы вещества выделяется или поглощается определенное количество теплоты, называемое теплотой фазового перехода.

Для фазовых переходов II рода характерно отсутствие скачкообразного изменения плотности вещества, концентрации компонентов, теплоты перехода. К фазовым переходам II рода относятся: переход парамагнетик – ферромагнетик, сопровождаемый появлением макроскопического магнитного момента; переход парамагнетик – антиферромагнетик, сопровождаемый появлением антиферромагнитного упорядочения; переход  $^3\text{He}$  и  $^4\text{He}$  в сверхтекучее состояние и т. д.

Фазовый переход вещества из аморфного в кристаллическое состояние, т.е. кристаллизация, относится к фазовым переходам I рода. В результате кристаллизации происходит выделение теплоты фазового перехода (теплоты кристаллизации) и скачкообразное изменение плотности вещества на величину  $\Delta\rho$ , равную  $\rho_c - \rho_a$ .  $\rho_c$  и  $\rho_a$  есть плотность вещества в кристаллическом и аморфном состояниях соответ-

ственно. Относительное изменение плотности вещества при кристаллизации  $\eta$  определим как:

$$\eta = \frac{\Delta\rho}{\rho_a} = \frac{\rho_c - \rho_a}{\rho_a}. \quad (1)$$

Выделим в аморфном образце параллелепипед со сторонами  $X_a$ ,  $Y_a$  и  $Z_a$ , внутри которого сосредоточена масса вещества, равная  $m$  (рис. 1).

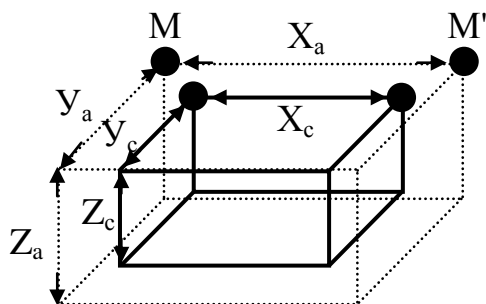


Рис. 1. К методике электронно-микроскопического определения относительного изменения плотности при кристаллизации вещества.

Тогда плотность аморфного вещества  $\rho_a = \frac{m}{X_a Y_a Z_a}$ . После кристаллизации образца эта же масса  $m$  будет расположена внутри параллелепипеда со сторонами  $X_c$ ,  $Y_c$  и  $Z_c$ . Плотность кристаллического вещества  $\rho_c = \frac{m}{X_c Y_c Z_c}$ . Подставив  $\rho_a$  и  $\rho_c$  в (1), получим:

$$\eta = \frac{X_a Y_a Z_a}{X_c Y_c Z_c} - 1. \quad (2)$$

В аморфном состоянии вещество является изотропным, поскольку отсутствует дальний порядок. В кристаллическом состоянии вещество анизотропное. Однако в случае мелкокристаллического вещества этой анизотропией можно пренебречь.

Поэтому с достаточной степенью точности можно считать, что  $\frac{X_a}{X_c} = \frac{Y_a}{Y_c} = \frac{Z_a}{Z_c}$ . С учетом последнего выражение (2) приобретает окончательный вид:

$$\eta = \left( \frac{X_a}{X_c} \right)^3 - 1. \quad (3)$$

Выражение (3) является ключевым при определении относительного измене-

ния плотности вещества методом электронной микроскопии. Для того, чтобы определить  $\eta$ , необходимо сделать две электронно-микроскопические фотографии одного и того же участка пленки до и после кристаллизации (рис. 2). После этого надо измерить расстояния  $X_a$  и  $X_c$  между фиксированными метками, жестко связанными с образцом. Если пленка получена методом импульсного лазерного осаждения, то в качестве меток удобно использовать затвердевшие микро-капли расплава (т.н. "брызговой эффект"). Микро-капли жестко связаны с пленкой. Поэтому изменение расстояния между ними в результате кристаллизации обусловлено не движением микро-капель относительно пленки, а изменением плотности самой пленки. На рис. 1 и на рис. 2 такие метки обозначены как  $M$  и  $M'$ . Величину  $\eta$  можно определить, подставив численные значения  $X_a$  и  $X_c$  в выражение (3).

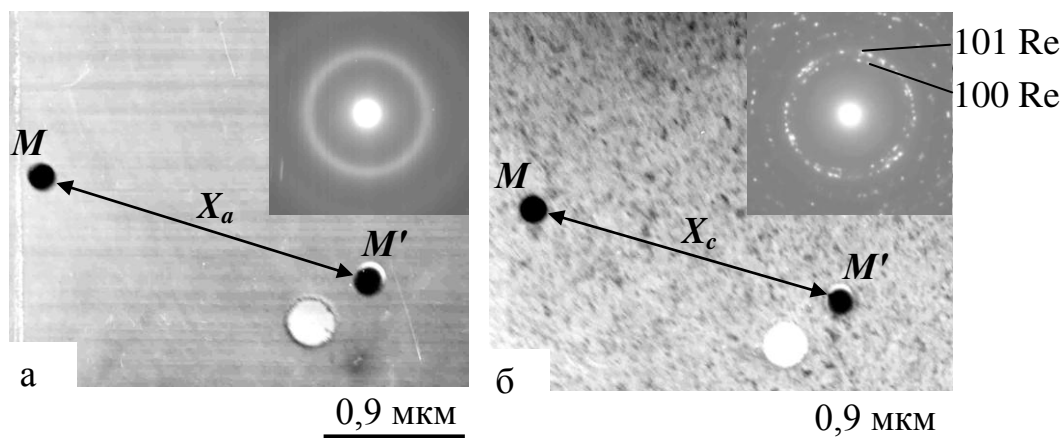


Рис. 2. Электронно-микроскопическое изображение пленки Re в исходном аморфном состоянии (а) и после ее кристаллизации электронным лучом (б). В правом верхнем углу каждого снимка приведены картины микродифракции, свидетельствующие о переходе Re из аморфного в кристаллическое состояние со структурой ГПУ.

Если на каждой из микрофотографий присутствует  $n$  меток, то число  $N$  возможных измерений расстояний между метками до и после кристаллизации (т.е.  $X_{ai}$  и  $X_{ci}$ ) составит  $\frac{n(n-1)}{2}$ . В этом случае по формуле (3) мы получим  $\frac{n(n-1)}{2}$  различ-

ных, но близких по значению величин  $\eta$ , подлежащих статистической обработке. Достоверность полученного результата растет с увеличением объема выборки (т.е. с ростом числа  $N$  измеренных значений  $\eta$ ). Увеличить объем выборки  $N$  можно посредством многократных повторных измерений расстояний между метками до и после кристаллизации пленки.

Значения  $X_a$  и  $X_c$  определяются по микрофотографиям как результат прямых измерений. Значения  $\eta$  вычисляются согласно соотношению (3) и являются косвенными измерениями. Кристаллизация данной аморфной пленки является необратимым процессом и условия проведения косвенных измерений являются невозможными. Вследствие этого обработка результатов должна проводиться так же, как и при прямых измерениях.

### Статистический анализ результатов измерений

Пусть в результате  $N$  измерений величина  $\eta$  последовательно принимает значения  $\eta_1, \eta_2, \dots, \eta_N$ . Весь интервал значений  $\eta$  от минимального значения  $\eta_{\min}$  до максимального значения  $\eta_{\max}$  разобьем на  $k$  одинаковых интервалов длиной  $\Delta\eta$  согласно соотношению:

$$\Delta\eta = \frac{\eta_{\max} - \eta_{\min}}{k}. \quad (4)$$

Тогда первый интервал  $\Delta\eta_1$  охватит значения  $\eta$  от  $\eta_{\min}$  до  $\eta_{\min} + \Delta\eta$ ; второй интервал  $\Delta\eta_2$  охватит значения от  $\eta_{\min} + \Delta\eta$  до  $\eta_{\min} + 2\Delta\eta$  и т.д. Последний интервал  $\Delta\eta_k$  охватит значения  $\eta$  от  $\eta_{\min} + (k-1)\Delta\eta$  до  $\eta_{\min} + k\Delta\eta$ .

Определим частоты появлений измеренных значений  $\eta$  в интервале  $\Delta\eta_1$  как  $\Delta N_1$ ; в интервале  $\Delta\eta_2$  как  $\Delta N_2$  и т.д. В последнем интервале  $\Delta\eta_k$  частота появлений измеренных значений  $\eta$  составит  $\Delta N_k$ . Относительные частоты  $\nu_i$  появлений измеренных значений  $\eta$  в  $i$ -том интервале  $\Delta\eta_i$  определим как:

$$\nu_i = \frac{\Delta N_i}{N}. \quad (5)$$

Естественно, что сумма относительных частот должна равняться единице:

$$\sum_{i=1}^k \nu_i = 1. \quad (6)$$

Выражение (6) должно использоваться для проверки вычисленных значений всей

совокупности относительных частот.

Для отображения того факта, что при измерении  $\eta$ , являющейся постоянной величиной для рассматриваемого вещества пленки, разные ее значения реализуются с разными частотами, строится гистограмма относительных частот (рис. 3). По оси абсцисс отложены значения  $\eta$ , разбитые на интервалы шириной  $\Delta\eta$ . По оси ординат отложены значения относительных частот  $\nu$ .

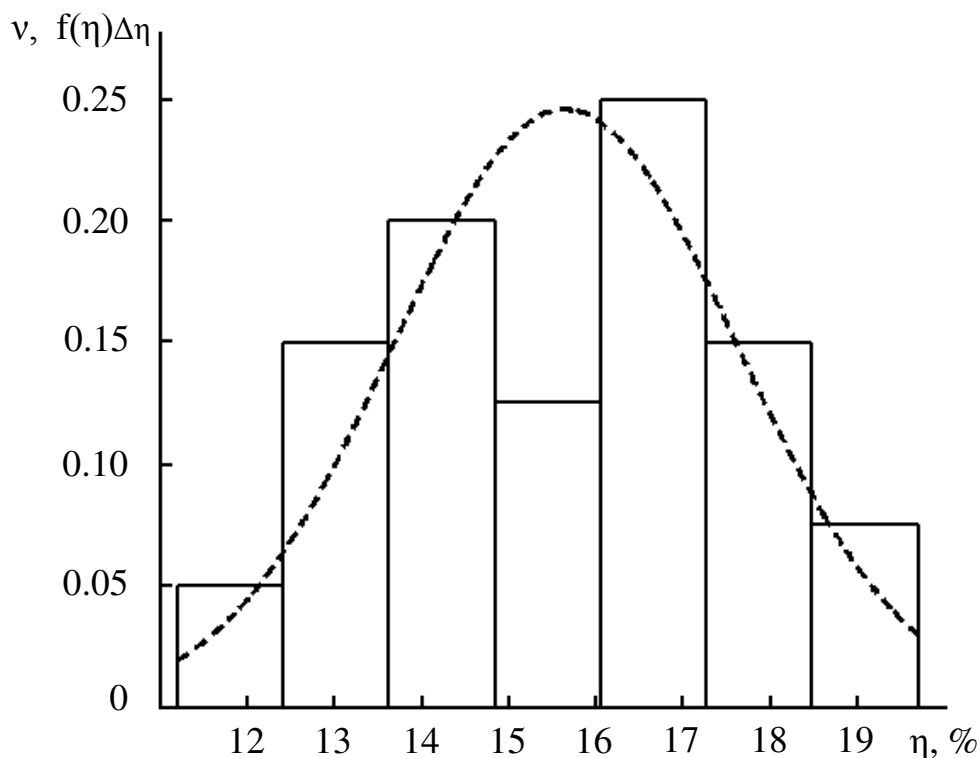


Рис. 3. Гистограмма относительных частот  $\nu$  и кривая распределения значений измеренной величины  $\eta$ , соответствующая распределению Гаусса (пунктирная кривая).

При неограниченном возрастании  $N$  и  $k$  ширина интервала  $\Delta\eta$  становится бесконечно малой (равной  $d\eta$ ). Вершины столбцов гистограммы образуют непрерывную линию, т.н. кривую распределения значений измеренной величины  $\eta$ .

Этой кривой отвечает функция распределения измеренной величины  $f(\eta)$  такая, что произведение  $f(\eta)d\eta$  есть вероятность того, что измеренное значение  $\eta$  попадет в интервал от  $\eta$  до  $\eta+d\eta$ . Функция  $f(\eta)$  является плотностью вероятности такого попадания. Она удовлетворяет условию:



$$\int_0^{\infty} f(\eta) d\eta = 1. \quad (7)$$

Во многих случаях аналитический вид  $f(\eta)$  соответствует т. н. “нормальному распределению”, или распределению Гаусса:

$$f(\eta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\eta - \mu)^2}{2\sigma^2}\right]. \quad (8)$$

В выражении (8)  $\mu$  есть математическое ожидание, а  $\sigma$  – среднее квадратическое отклонение нормального распределения.  $\sigma^2$  – есть дисперсия непрерывной случайной величины.

Поскольку число измерений ограничено ( $N \neq \infty$ ), то значения  $\mu$ ,  $\sigma^2$  и  $\eta$  остаются неизвестными. Их можно приблизительно оценить так называемой выборочной (исправленной) дисперсией  $s^2$ :

$$s^2 = \frac{\sum_{i=1}^N (\eta_i - \langle \eta \rangle)^2}{N - 1}. \quad (9)$$

В выражении (9)  $\langle \eta \rangle$  есть среднее арифметическое значение  $\eta$ :

$$\langle \eta \rangle = \frac{1}{N} \sum_{i=1}^N \eta_i. \quad (10)$$

С помощью  $s^2$  вводится понятие об исправленном среднем квадратичном отклонении  $s$ , характеризующем рассеяние результатов измерения величины  $\eta$ :

$$s = \sqrt{s^2}. \quad (11)$$

Погрешность  $\Delta\eta(\alpha)$ , ограничивающая доверительный интервал, устанавливается соотношением:

$$\Delta\eta(\alpha) = s \cdot t_\alpha(N), \quad (12)$$

где  $t_\alpha(N)$  есть коэффициент Стьюдента. Значения коэффициентов Стьюдента по заданному значению надежности  $\alpha$  и величине  $N$  определяют с помощью специальной таблицы, приведенной в приложении 1.

Результат измерений величины относительного изменения плотности при кристаллизации пленки определяют формулой:

$$\eta = \langle \eta \rangle \pm \Delta\eta(\alpha). \quad (13)$$

В соответствии с (13) истинное значение величины  $\eta$  попадает в доверительный

интервал ( $\langle \eta \rangle - \Delta\eta(\alpha)$ ;  $\langle \eta \rangle + \Delta\eta(\alpha)$ ) с надежностью, равной  $\alpha$ .

Рассмотрим следующий пример. Пусть в результате  $N = 40$  измерений величина  $\eta$  последовательно принимает значения 17.5, 16.9, 13.4, 16.6, 14.1, 19.1, 15.6, 19.3, 14.7, 17.0, 15.5, 11.2, 13.5, 17.0, 15.6, 13.2, 14.3, 16.2, 13.5, 17.8, 14.3, 19.7, 13.3, 16.1, 12.3, 14.1, 14.4, 16.6, 17.8, 16.3, 14.9, 17.5, 17.5, 16.0, 16.9, 16.6, 13.5, 17.4, 14.2, 14.4. Весь интервал значений  $\eta$  от минимального значения  $\eta_{\min} = 11.2$  до максимального значения  $\eta_{\max} = 19.7$  согласно (4) разбиваем на  $k = 7$  одинаковых интервалов длиной  $\Delta\eta = 1.2$ . Первый интервал охватывает значения  $\eta$  от 11.2 до 12.4; второй интервал охватывает значения от 12.4 до 13.6. Последний интервал  $\Delta\eta_k$  охватит значения  $\eta$  от 18.5 до 19.7. Соответствующая гистограмма относительных частот приведена на рис. 3.

В соответствии с (10) среднее арифметическое значение  $\langle \eta \rangle = 15.6$ . Согласно (9) выборочная (исправленная) дисперсия  $s^2 = 3.9$ , а согласно (11) исправленное среднее квадратичное отклонение  $s = 2.0$ . График функции  $f(\eta)\Delta\eta$ , соответствующий распределению Гаусса (8), приведен на рис. 3 (пунктирная линия). При построении графика сделано допущение, что  $\mu = \langle \eta \rangle$ ,  $\sigma = s$  и  $\sigma^2 = s^2$ .

Задав значение надежности  $\alpha = 0.95$  при  $N = 40$  по таблице из приложения 1 определяем численное значение коэффициента Стьюдента  $t_{0.95}(40) = 2.1$ . В соответствии с (12) погрешность, ограничивающая доверительный интервал  $\Delta\eta = 2.0 \times 2.1 = 4.2$ . Согласно (13) результат измерений представим в виде:  $\eta = 15.6 \pm 4.2$ . Следовательно, истинное значение величины  $\eta$  попадает в доверительный интервал (11.4; 19.8) с надежностью, равной 0.95.

В приложении 2 приведены некоторые функции системы MATLAB, облегчающие проведение статистической обработки результатов измерений.

### 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выбрать в поле зрения электронного микроскопа участок аморфной пленки, содержащей метки, подобные тем, что приведены на рис. 2а. Сфотографировать его на фотопластинку.

2. Не перемещая образец закристаллизовать этот участок пленки, используя

термический или электронно-лучевой нагрев. После остывания образца сфотографировать закристаллизованный участок пленки (подобный приведенному на рис. 2б) на фотопластинку.

3. С помощью фото-сканера провести оцифровку полученных изображений участка пленки до и после кристаллизации. Для этого обе фотопластинки на сканере следует расположить рядом и сканировать одновременно. При сканировании желательно использовать разрешение не менее 600 dpi. Тип изображения – серый, 8 бит. Изображение сохранить в виде файла с расширением “jpg”.

4. Запустить приложение MATLAB. Получить изображение объекта, записав в окне “Command Window” команды, содержащие функции “imread” и “imshow”. Перед этим необходимо убедиться, что требуемый файл находится в той папке, которая отображена в окне “Current Folder” (см. приложение 2).

5. Вызвать файл функцию “fdensity”. Для этого в окне “Command window” следует набрать имя файл функции и нажать клавишу “Enter”.

6. Провести измерение расстояний между метками до и после кристаллизации пленки (не менее 8). Результаты измерений оформить в виде соотношения, подобного соотношению (13), а также в виде гистограммы, подобной приведенной на рис. 3.

#### 4. КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Что такое фазовый переход (фазовое превращение)?
2. Чем отличается фазовый переход I рода от фазового перехода II рода?
3. Приведите примеры фазовых переходов I и II рода.
4. Что такое плотность вещества и как она изменяется при фазовом превращении из аморфного в кристаллическое состояние?
5. Выведите формулу, связывающую расстояние между метками в пленке в аморфном и кристаллическом состояниях с относительным изменением плотности при кристаллизации пленки.
6. Если на микрофотографии пленки изображено  $n$  меток, то какое число линий между метками можно провести на фотоснимке?

7. Что такое относительная частота появления измеренных значений физической величины в заданном интервале и чему равна сумма всех относительных частот?
8. Что такое гистограмма относительных частот и как она строится?
9. Какому условию удовлетворяет функция плотности вероятности?
10. Запишите аналитический вид функции распределения Гаусса.
11. Что такое выборочная исправленная дисперсия и исправленное среднее квадратичное отклонение?
12. Как определить доверительный интервал по заданному уровню надежности и по числу измерений физической величины?

### СПИСОК ЛИТЕРАТУРЫ

1. А. С. № 1689794 СССР, МКИ G 01 N 9 / 00. Способ определения относительно-го изменения плотности / А.Г. Багмут, Г.П. Николайчук (СССР).- № 4713629 / 25; Заявл. 04.07.89; Опубл. 07.11.91, Бюл. № 41.
2. Хирш П., Хови А., Николсон Р., Пэшли Д., Уэлан М. Электронная микроскопия тонких кристаллов.- Москва: Мир, 1968.- 574 с.
3. Гмурман В.С. Теория вероятностей и математическая статистика.- Москва: Высшая школа, 1972.- 368 с.
4. Колупаев И.Н., Шипкова И.Г. Количественная обработка цифровых изображений микроструктур. (Система MatLab). Учебно-методическое пособие по курсу “Численные методы математики и математической физики”.- Харьков: НТУ “ХПИ”, 2006.- 82 с.
5. Багмут А.Г., Багмут И.А., Николайчук Г.П., Жучков В.А. Методические указания к выполнению лабораторной работы “Определение статистических характеристик частиц по микрофотографиям” по курсу “Электронная микроскопия и электронография” для студентов физико-технического, механико-технологического и инженерно-физического факультетов.- Харьков: НТУ “ХПИ”, 2011.- 36 с.

6. Мещеряков В.В. Задачи по статистике и регрессионному анализу.- Москва: Диалог Мифи, 2009.- 448 с.
7. Базакуца В.А., Лебедева М.В., Сук А.Ф. и др. Методические указания к лабораторным работам по разделу “Механика” курса физики.- Харьков: ХПИ, 1989.- 198 с.

## ПРИЛОЖЕНИЯ

### Приложение 1.

Таблица.

Значения коэффициентов Стьюдента  $t_{\alpha}(N)$  по заданным значениям надежности  $\alpha$  и числу измерений  $N$

N-1	$t_{\alpha}(N)$					
	$\alpha$					
	0.5	0.7	0.8	0.9	0.95	0.99
1	1.00	2.0	3.1	6.3	12.7	63.7
2	0.82	1.3	1.9	2.9	4.3	9.9
3	0.77	1.3	1.6	2.4	3.2	5.8
4	0.74	1.2	1.5	2.1	2.8	4.6
5	0.73	1.2	1.5	2.0	2.6	4.0
6	0.72	1.1	1.4	1.9	2.4	3.7
7	0.71	1.1	1.4	1.9	2.4	3.5
8	0.71	1.1	1.4	1.9	2.3	3.4
9	0.70	1.1	1.4	1.8	2.3	3.3
10	0.70	1.1	1.4	1.8	2.2	3.2
50	0.68	1.1	1.3	1.7	2.0	2.7
100	0.88	1.0	1.3	1.7	2.0	2.6
	0.67	1.0	1.3	1.6	2.0	2.6

### Приложение 2. Некоторые функции в системе MATLAB, применяемые при обработке результатов измерений и при работе с цифровыми изображениями

Наблюдаемые значения  $x_i$  в системе MATLAB можно представить в виде вектора-строки  $X=[x_1, x_2, \dots, x_i]$ . Рассмотрим некоторые статистические функции системы MATLAB и результаты, возвращаемые ими на примере вектора-строки  $X$ .

$X = [17.5, 16.9, 13.4, 16.6, 14.1, 19.1, 15.6, 19.3, 14.7, 17.0, 15.5, 11.2, 13.5, 17.0, 15.6, 13.2, 14.3, 16.2, 13.5, 17.8, 14.3, 19.7, 13.3, 16.1, 12.3, 14.1, 14.4, 16.6, 17.8, 16.3, 14.9, 17.5, 17.5, 16.0, 16.9, 16.6, 13.5, 17.4, 14.2, 14.4]$ .

**П.2.1. Определение числа элементов (длины) вектора X.** Для этого используется команда **“length (X)”**. Результатом применения команды является число 40.

**П.2.2. Нахождение минимального и максимального значения элементов вектора X.**

а). Минимальное значение находится посредством команды **“min (X)”**. Результатом ее применения является число 11.2000.

б). Максимальное значение находится посредством команды **“max (X)”**. Результатом ее применения является число 19.7000.

**П.2.3. Нахождение среднего арифметического значения из элементов вектора X (согласно соотношению (10)).** Для этого используется команда **“mean (X)”**. Результатом применения команды является число 15.6450.

**П.2.4. Нахождение исправленной дисперсии  $s^2$  из элементов вектора X (согласно соотношению (9)).** Для этого используется команда **“var (X)”**. Результатом применения команды является число 3.8815.

**П.2.5. Нахождение исправленного среднего квадратического отклонения  $s$  из элементов вектора X (согласно соотношению (11)).** Для этого используется команда или **“std (X)”**. Результатом применения этих команд является число 1.9702.

**П.2.6. Построение гистограммы частот элементов вектора-строки.** Команда **“hist (X, nbins)”** строит гистограмму частот элементов вектора-строки X. Число частичных интервалов (число “бинов”) задается величиной “nbins”. Если эта величина не указана, то число бинов по умолчанию будет равно 10 (команда **“hist (X)”**). В качестве примера построим гистограмму частот элементов вектора X, используя 7 частичных интервалов (nbins = 7). Для этого в командном окне следует набрать **“hist (X, 7)”**. Результат исполнения этой команды подобен гистограмме,

показанной на рис. 3. Отличие состоит в том, что по оси ординат рис. 3 отложены относительные частоты, которые в  $N = 40$  раз меньше частот элементов вектора  $X$ .

Определим вектор-строку  $\mathbf{x_i}$ , элементами которого являются значения середин частичных интервалов и вектор-строку  $\mathbf{n_i}$ , элементами которого являются суммы частот вариантов частичных интервалов. Для этого в командном окне следует набрать “[ $\mathbf{n_i}, \mathbf{x_i}$ ]=hist ( $\mathbf{X}, 7$ )”. Результат исполнения команды следующий:

```
ni =      2          6          8          5          10          6          3
xi = 11.8071  13.0214  14.2357  15.4500  16.6643  17.8786  19.0929
```

Он означает то, что в область первого частичного интервала, середине которого соответствует число 11.8071, попало 2 элемента вектора  $X$ . В область второго частичного интервала, середине которого соответствует число 13.0214, попало 6 элементов вектора, и т.д.

**П.2.7. Построение нормальной кривой функции плотности вероятности (соотношение 8)) по опытным данным, представленным элементами вектор-строки  $X$ .** Для этого целесообразно использовать стандартную MATLAB-функцию “**normpdf**” (normal probability density function). Она определяет элементы вектора  $Y$  с помощью команды  **$Y = \text{normpdf}(t, \mu, \sigma)$** . В этом выражении  $\mu$  есть выборочное среднее (10);  $\sigma$  есть исправленное среднее квадратичное отклонении  $s$  (11). В результате применения команды будут вычислены значения элементов вектора  $Y$  по формуле (8) в точках  $t$ , охватывающих интервал от минимального до максимального значения среди элементов вектора  $X$ . Вывод графика можно осуществить командой “**plot(t, Y, 'k', 'LineWidth', 1.5)**”. Параметры “**'k', 'LineWidth', 1.5**” задают цвет и ширину линии кривой, выводимой на экран компьютера. Итоговая форма записи следующая:

```
t=min(X):0.01:max(X);
Y=normpdf(t, mean(X), std(X));
plot(t, Y, 'k', 'LineWidth', 1.5)
```

В результате выполнения этих команд будет построена кривая плотностей относительных частот, подобная той, что приведена на рис. 3. Отличие состоит в том, что относительная частота (отложенная по оси ординат на рис. 3) превышает

значение плотности относительной частоты в число раз, равное численному значению ширины задаваемого интервала (бина).

**П.2.8. Совместное построение столбчатой диаграммы относительных частот элементов вектора X и функции распределения Гаусса.**

```
nbins = 7; [ni, xi]=hist (X, nbins);  
bar (xi, ni./length(X)); hold on;  
h=(max(X)-min(X))/nbins; t=min (X):0.01:max (X);  
Y = (normpdf(t, mean (X), std (X))).*h; plot (t, Y, 'k' ,'LineWidth', 1.5)
```

В результате выполнения приведенного набора команд будет построена столбчатая диаграмма относительных частот элементов вектора X и график функции распределения Гаусса, приведенные на рис. П.1. Вид диаграммы можно изменить, используя инструменты графического редактора.

**П.2.9. Создание массивов элементов.** Для создания массива **S**, состоящего из N элементов, которые распределены случайным образом, надо воспользоваться командой **S=rand(1, N)** (формирование вектор-строки) или командой **S=rand(N, 1)** (формирования вектор-столбца). Для создания массива **G**, состоящего из N элементов, которые распределены по нормальному закону, имеют среднее арифметическое значение **mu** и исправленное среднее квадратичное отклонение **sigma** используется команда **G=normrnd(mu, sigma, 1, N)** (формирование вектор-строки) или команда **G= normrnd(mu, sigma, N, 1)** (формирование вектор-столбца).

Пример:

```
N=40; mu= 15.735; sigma= 2.2105;  
G=normrnd(mu, sigma, 1, N)
```

В результате выполнения этого набора команд будет создан вектор-строка **G**, для которого среднее арифметическое значение **mu = 15.735**, а исправленное среднее квадратичное отклонение **sigma = 2.2105**.



Для совместного построения столбчатой диаграммы и нормальной кривой по элементам созданного вектора-строки **G** можно использовать следующие команды:

```
N=40; mu= 15.735; sigma= 2.2105;  
G=normrnd (mu, sigma, 1, N);  
nbins = 7; [ni, xi]=hist (G, nbins);  
bar (xi, ni./length(G)); hold on;  
h=(max(G)-min(G))/nbins; t=min (G):0.01:max (G);  
Y = (normpdf(t, mean (G), std (G))).*h; plot (t, Y, 'k' , 'LineWidth', 1.5)
```

Результаты выполнения этих команд отражены на рис. П.1. Середины столбцов диаграммы соответствуют значениям  $x_i$  по оси абсцисс. Суммы относительных частот вариант частичных интервалов  $n_i$  отложены по оси ординат.

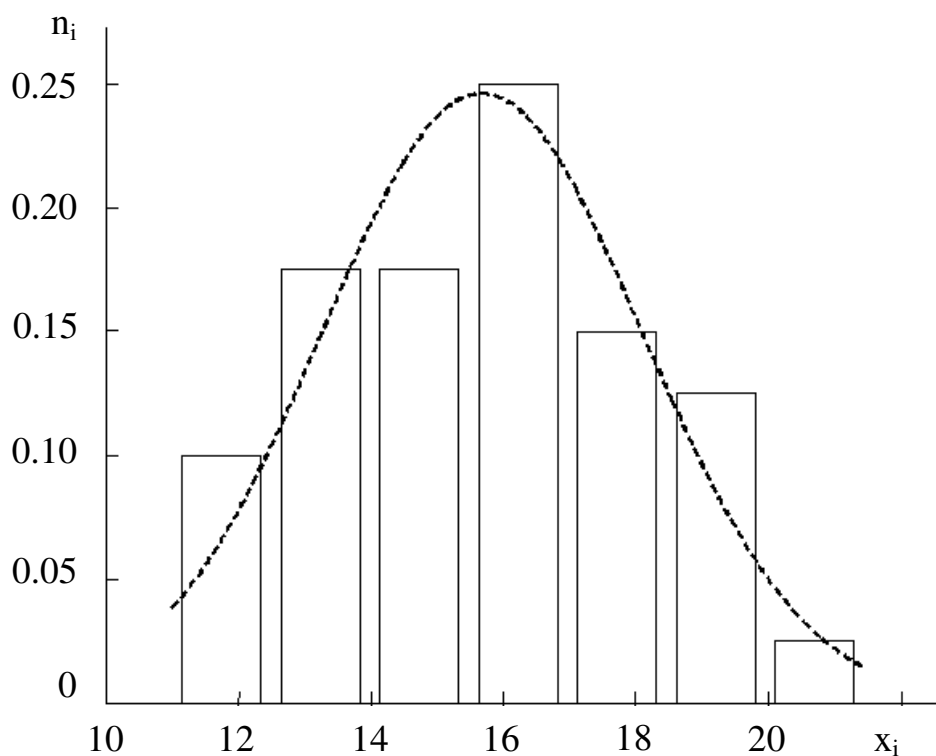


Рис. П.1. Столбчатая диаграмма и нормальная кривая, построенные по элементам вектора-строки **G**.

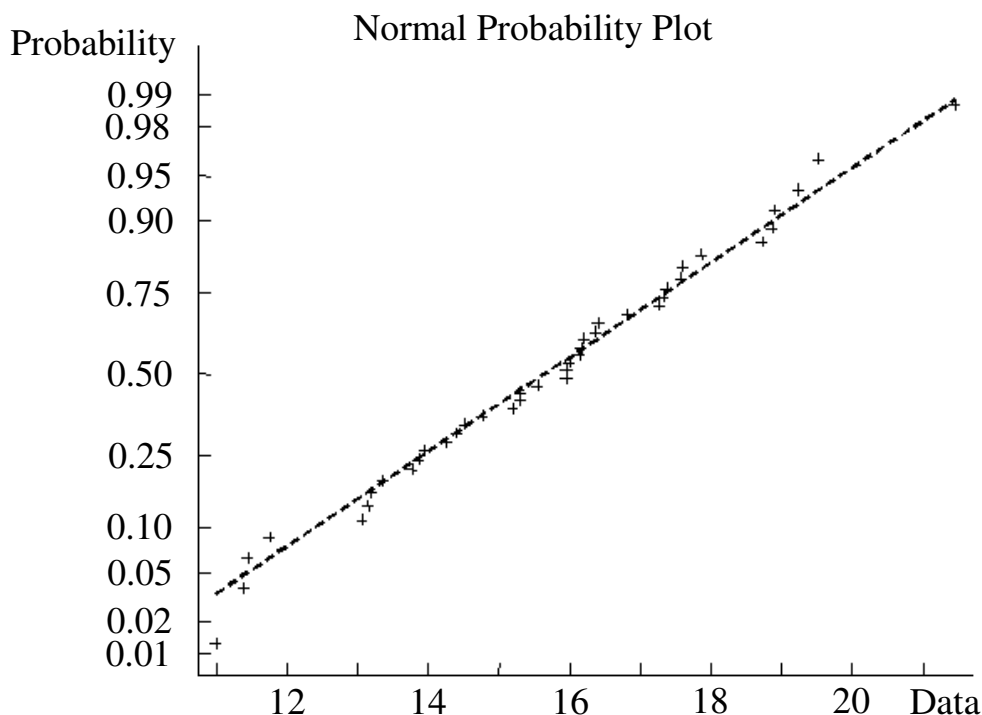


Рис. П.2. Результат использования функции `normplot`, примененный к элементам вектора-строки **G**.

**П.2.10. Проверка гипотезы о “нормальности” распределения элементов вектора **G**.** Визуальную оценку нормальности распределения можно провести, воспользовавшись высокоуровневой функцией `normplot(G)` или `probplot(G)`. Результат использования функции `normplot` показан на рисунке П.2. Если опытные данные, обозначенные маркерами “+” на графике рис. П.2 расположились вблизи сплошной части наклонной прямой, то, следовательно, распределение данных близко к нормальному закону.

**П.2.11. Формирование и вывод на экран цифрового изображения.** Цифровые изображения формируются посредством сканирования электронно-микроскопических фотографий, отпечатанных на стекле, фотопленке или фотобумаге с последующей их записью в виде файлов в память компьютера. Как правило, используются полутоновые изображения (grayscale), содержащие пиксели с рядом оттенков серого цвета. В памяти компьютера они хранятся в виде двумерных мас-

сиров-матриц. Работать с цифровыми изображениями в системе MATLAB позволяет пакет специализированных программ “Image Processing Toolbox”.

Для работы в окне MATLAB “Command Window” файл, к которому следует обратиться, должен находиться в той папке, которая отображена в окне “Current Folder”. Выбор требуемой папки осуществляется с помощью выпадающего меню “Обзор папок”, которое появляется после нажатия кнопки “Browse for folder”.

**Функция “*imread*”.** Для считывания графического объекта в командной строке MATLAB следует набрать команду `I = imread('filename.fmt')`, где `filename` есть имя файла, а `fmt` – его формат. Например, для считывания цифрового изображения, записанного в файле `para_dv_v.jpg`, Следует выполнить команду **“`I=imread('para_dv_v.jpg');`”**. В результате ее выполнения будет сформирован двумерный числовой массив `I`. Для предотвращения вывода числового массива на экран компьютера в конце команды следует ставить “;”.

**Функция “*imshow*”.** Эта функция позволяет получить изображение на экране компьютера. Выполнение команды **“`imshow(I)`”** приводит к выведению в отдельном окне изображения, записанного в файле `para_dv_v.jpg` (рис. П.3). Для того, чтобы уже выведенное на экран изображение не исчезало при выведении следующего изображения, перед командой “`imshow`” следует задавать команду “`figure`” (например **“`figure, imshow(J)`”**).

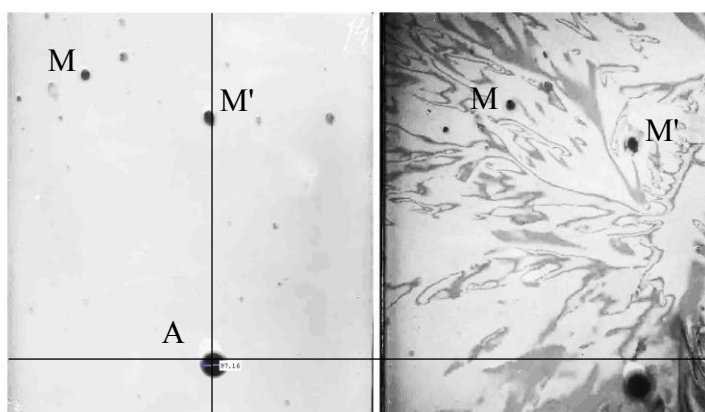
Каждому элементу массива `I` соответствует определенный пиксель изображения, который характеризуется яркостью. Как правило, используется шкала, где яркости пикселей изменяются от 0 (черный цвет) до 255 (белый цвет).

**П.2.12. Информация о цифровом изображении. Функция “*IMPIXELINFO*”.** Она позволяет получать первичную количественную информацию об объекте изображения. В результате выполнения команды **“*IMPIXELINFO*”** в нижнем левом углу открытого изображения появляется надпись *“Pixel info:(X, Y) Intensity”* (Если курсор находится в области рамки изображения). При перемещении курсора в области изображения в круглых скобках надписи отображаются текущие координаты курсора по горизонтали (X) и по оси вертикали (Y). Левый верхний угол изображения соответствует координатам (0, 0). Правее координат

отображается яркость пикселей (*Intensity*). Например, если курсор находится в области темной метки “А” (рис. П.3), то надпись отобразит: *Pixel info:(1704, 1007) 20*. В соседней области, примыкающей к метке “А”, надпись отобразит: *Pixel info:(1746, 1011) 62*. Следовательно, разница в яркости изображений метки и фона (т. е. “контраст”), составит 42 пикселя.

**Функция “*imdistline*”.** Функция позволяет определить число пикселей вдоль прямой, соединяющей две выбранные точки на изображении. В результате выполнения команды “***h=imdistline***” на открытом изображении появляется прямая линия с двумя маркерами, подобная той, что изображена на рис. П.3. Число, пропечатываемое в белом прямоугольнике, соответствует числу пикселей, разделяющих два маркера. Устанавливая маркеры с помощью мыши на интересующие нас точки изображения, можно определить расстояние (в пикселях) между этими точками. Так, диаметр наиболее крупной метки на рис. П.3. равен 80 пикселям. Вывести это число на экран компьютера позволяет команда “***getDistance(h)***”.

Если кликнуть правой кнопкой мыши в тот момент, когда курсор находится на прямой линии или на одном из маркеров, то происходит выпадение контекстного меню, содержащего дополнительные опции. Более подробно о возможностях функции “*imdistline*” можно ознакомиться, используя справочную систему MATLAB.



Pixel info: (1704, 1007) 20

Рис. П.3. Изображение объекта, записанного в файле “para\_dv\_v.jpg”. Показано перекрестие курсора, наведенное на метку “А”

### Приложение 3.

#### **Файл-функция “fdensity”. Руководство пользователя**

Предлагаемая программа составлена для работы в системе MATLAB. Она позволяет по электронно-микроскопическим фотографиям пленки до и после кристаллизации измерить расстояния между выбранными метками, определить относительное изменение плотности вещества в результате кристаллизации пленки и провести статистическую обработку результатов измерений.

Программа “fdensity” для своей работы вызывает три вспомогательные функции: `colorcmp`, `scan_line` и `scan`. Листинги данных файл функций приведены ниже.

Поясним работу программы на конкретном примере. Допустим, мы исследуем относительное изменение плотности при кристаллизации пленки аморфного  $V_2O_3$ . Изображение участка пленки до и после кристаллизации было сфотографировано на двух фотопластинках размером 9x12 см при электронно-микроскопическом увеличении  $M_{\text{э}} = 25000$ . После оцифровывания этих изображений (с использованием фото-сканера) был сформирован один файл, содержащий оба изображения (до и после кристаллизации). Файл имеет имя и расширение “para\_dv\_v.jpg”.

Запускаем систему MATLAB. В окне “Command window” набираем имя файл функции “fdensity”. В командном окне появляются строки.

*Функция вычисления относительного изменения плотности.*

*Введите <1> - для обработки одного цифрового изображения*

*Введите <2> - для обработки двух цифровых изображений*

В том случае, когда изображение участка пленки до и после кристаллизации сфотографировано на двух фотопластинках, каждая из которых оцифрована и записана в памяти компьютера в виде отдельного файла, то следует вводить “2”. В случае одного файла (более удобный вариант, как в рассматриваемом примере), следует вводить “1”. Вводим “1”. В командном окне появляются строки.

*Введите имя файла (в одинарных кавычках) с цифровым изображением.*

Вводим: “para\_dv\_v.jpg” и нажимаем “Enter”.

В окне “Command window” следует команда:

*Введите <1> - для автоматического распознавания меток*

*Введите <2> - для распознавания меток в ручном режиме*

В том случае, когда изображение меток является контрастным, а метки выглядят резко очерченными, целесообразно использовать работу программы “fdensity” в автоматическом режиме. В противном случае более эффективным будет работа программы в ручном режиме.

**3.1. Работа программы “fdensity” в автоматическом режиме.** В этом случае следует ввести в окне “Command window” цифру “1”. После ввода единицы следует команда: *“Введите максимально допустимый горизонтальный размер объектов изображения (в пикселях). При введении 0, по **умолчанию, будет** задан размер, равный 200”*.

В автоматическом режиме программа “fdensity” определяет “центр масс” на изображении той метки, на которую наведен курсор. Вышеприведенная команда ограничивает допустимые размеры используемых меток. Если он не известен, то следует ввести “0”. Однако в разделе П.2.12 в качестве примера было установлено, что размер подобной метки равен 80 пикселям. Поэтому вводим “80”. На экране компьютера появляется команда:

*“Введите максимально допустимый вертикальный размер объектов изображения (в пикселях). При введении 0, по умолчанию, будет задан размер, равный 200”*.

Смысл этой команды ясен из предыдущего пояснения. Вводим “80”. На экране компьютера появляется команда:

*“Введите максимально допустимую разницу в яркости пикселей [0 - 255] между изображением метки и фоном. При введении 0, по умолчанию, будет задана разница, равная 20”*.

Вышеприведенная команда ограничивает допустимую разницу в яркости пикселей между изображением метки и фоном. Чем она больше, тем большая область на изображении будет восприниматься как одна метка. Если эта разница не известна, то следует ввести “0”. Но в разделе П.2.12 в качестве примера было уста-

новлено, что разница в яркости пикселей между изображением метки и фоном равна 42 пикселя. Поэтому вводим “42”. На экране компьютера появляется команда:

*“Задайте значение надежности,  $\alpha$  = ”.*

Значение надежности  $\alpha$  выбираем по таблице из приложения 1. Наиболее употребительный вариант соответствует  $\alpha = 0.95$ . Поэтому вводим “0.95”.

На экране компьютера появляется новое окно (Figure 1) подобное тому, что приведено на рис. П.3. Окно содержит скользящий курсор в виде перекрестия. Если навести центр перекрестия на выбранную метку на фотографии аморфной пленки и кликнуть левой клавишей мышки, то выбранная метка окрасится в красный цвет. После этого центр перекрестия следует расположить на другой метке этой же фотографии аморфной пленки и повторить описанную процедуру. Между двумя метками возникает соединяющая их линия, имеющая номер проведенного измерения (в данном случае номер 1). Описанная процедура повторяется для тех же двух меток на изображении пленки после кристаллизации. Линия, соединяющая окрашенные метки на изображении закристаллизованной пленки, имеет тот же номер 1. Для статистики следует иметь как можно большее число пар линий с одинаковым номером. Каждая пара линий дает одно значение относительного изменения плотности. В рассматриваемом примере число пар равно 8. Кликнув левой клавишей мышки вне поля фотографий Figure 1, на экране компьютера увидим сообщение:

$p1 = 6.2706 \%$	$p2 = 4.1079 \%$
$p3 = 5.5641 \%$	$p4 = 6.3367 \%$
$p5 = 4.8849 \%$	$p6 = 1.2032 \%$
$p7 = 7.0472 \%$	$p8 = 8.0256 \%$
$\langle p \rangle = 5.4300 \%$	Ответ: $p = 5.4300 \pm 4.6405 \%$

В рассмотренном примере  $p1$  есть относительное изменение плотности, определенное по паре линий с номером 1,  $p2$  – по паре линий с номером 2 и т. д.  $\langle p \rangle$  – есть среднее арифметическое значение. Ответ гласит, что истинное значение относительного изменения плотности  $\eta$  (в программе  $\eta$  переобозначено как  $p$ ) попадает

в доверительный интервал (0.7895; 10.0705) с надежностью, равной 0.95. Следует заметить, что ширина доверительного интервала уменьшается с уменьшением уровня надежности и с увеличением числа измерений.

Следующая команда: *“Записать результаты измерений в файл: <1> - Да, <0> - Нет ?”*.

Смысл команды ясен и не требует пояснений. Вводим “0”. Следующая команда:

*“Введите число интервалов (бинов) для построения гистограммы измерений. (при введении 0 число бинов определяется автоматически)”*.

В данном примере мы предоставим компьютеру выбрать число столбцов будущей гистограммы (т. е. число бинов). Вводим “0”.

Следующая команда:

*“Введите <1> - для построения нормальной кривой по результатам измерений. Введите <2> - для отображения гистограммы без нормальной кривой”*.

Вводим “1”. На экране компьютера появляется сообщение, отражающее статистические характеристики проведенных измерений, которые после округления до второго знака сведены в таблицу. П1 (третий столбец).

По результатам измерений в окне “Figure 2” приводится гистограмма частот и график функции, соответствующей распределению Гаусса (рис. П.4).

В окне “Figure 3” приводится результат использования функции “normplot”, примененный к элементам вектора **p**. (рис. П.5).

Табл. П.1. Результаты измерения относительного изменения плотности  $\eta$ .

№	Статистические характеристики измерений	Автоматический режим	Ручной режим
1	Минимальное значение	1.2 (%)	3.7 (%)
2	Максимальное значение	8.0 (%)	8.5 (%)
3	Среднее значение	5.4 (%)	6.8 (%)
4	Стандартное отклонение	2.0 (%)	1.5 (%)
5	Коэффициент вариации	36.1 (%)	22.4 (%)



6	Асимметрия	-0.9	-0.7
7	Экссесс	0.2	-0.4

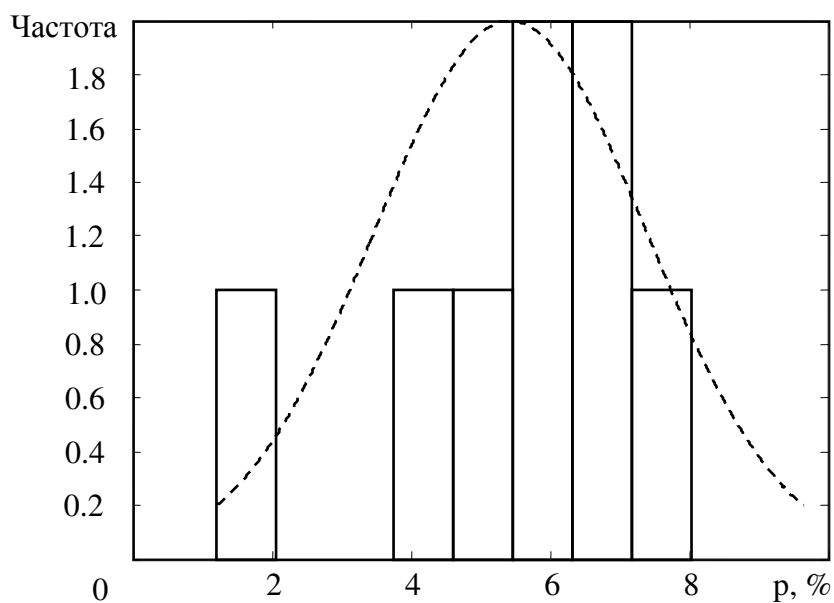


Рис. П.4. Гистограмма частот и нормальная кривая (пунктирная кривая), построенные в результате работы программы “fdensity” в автоматическом режиме.

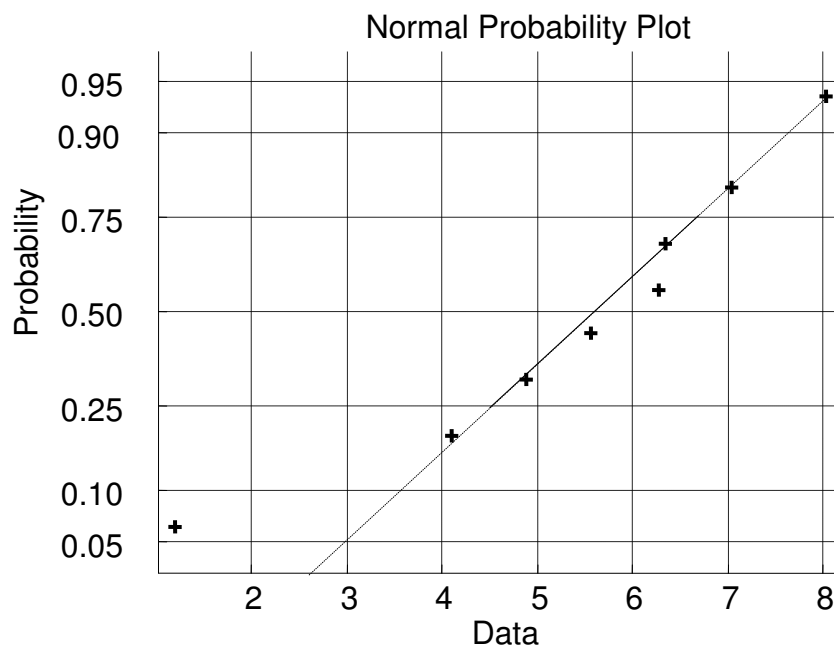


Рис. П.5. Результат использования функции normplot, примененный к элементам вектора **p**.

**3.2. Работа программы “fdensity” в ручном режиме.** После появления в окне “Command window” сообщения:

*Введите <1> - для автоматического распознавания меток*

*Введите <2> - для распознавания меток в ручном режиме*

Следует ввести в окне “Command window” цифру “2”. После ввода двойки следует команда:

*“Введите <1> - для отображения маркеров распознаваемых меток*

*Введите <2> - для распознавания меток без отображения их маркеров”.*

В первом случае место расположения перекрестия маркера на выбранной метке после щелчка левой клавишей мышки отмечается красным кружком. Во втором случае этого не происходит. В качестве примера вводим “1”. На экране компьютера появляется команда:

*“Задайте значение надежности,  $\alpha =$  ”.*

Значение надежности  $\alpha$  выбираем по таблице из приложения 1. Для разнообразия примеров зададим  $\alpha = 0.5$ . Вводим “0.5”.

На экране компьютера появляется окно (Figure 1) подобное тому, что приведено на рис. П.3. Далее следуют процедуры, описанные в предыдущем разделе. Проведя требуемое число пар линий (в нашем примере снова 8 пар) и кликнув левой клавишей мышки вне поля фотографий Figure 1, на экране компьютера появляется сообщение:

$$p1 = 3.7372 \%$$

$$p2 = 5.8870 \%$$

$$p3 = 8.5761 \%$$

$$p4 = 8.4501 \%$$

$$p5 = 5.7726 \%$$

$$p6 = 6.9799 \%$$

$$p7 = 7.8084 \%$$

$$p8 = 7.5270 \%$$

$$<p> = 6.8423 \%$$

$$\text{Ответ: } p = 6.8423 \pm 1.0876 \%$$

Ответ гласит, что истинное значение относительного изменения плотности попадает в доверительный интервал (5.7547; 7.9299) с надежностью, равной 0.5. Заметим, что ширина доверительного интервала стала меньше по сравнению с предыдущими измерениями, поскольку мы задали меньшее значение уровня надежности (при том же числе измерений).

После построения гистограммы (содержащей по нашему заданию 4 столбца) получим окончательный результат, приведенный в четвертом столбце табл. П1 (после округления до второго знака после запятой).

Сопоставим оба приведенных ответа. На рис. П.6а приведен результат измерения (округленный до второго знака после запятой) относительного изменения плотности  $\eta$ , выполненный в автоматическом режиме для  $V_2O_3$  с заданным уровнем надежности  $\alpha = 0.95$ . На рис. П.6б - тоже для измерений, выполненных в ручном режиме с уровнем надежности  $\alpha = 0.5$ . Непротиворечивость результатов очевидна: доверительный интервал, приведенный на рис. П.6а, охватывает доверительный интервал, приведенный на рис. П.6б.

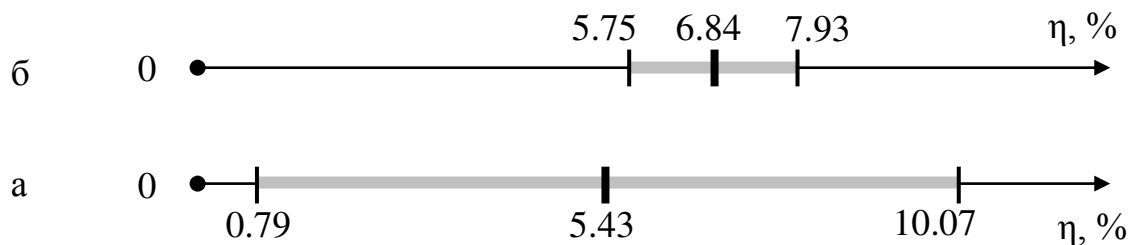


Рис. П.6. Результат измерения относительного изменения плотности  $\eta$ , выполненный в автоматическом режиме с уровнем надежности 0.95 (а) и в ручном режиме с уровнем надежности 0.5 (б).

#### Приложение 4. Листинг программы “fdensity”

```
function fdensity
OK = false;
while OK == false;
```

```

clc; clear all;

try
s = '      Функция вычисления относительного изменения плотности p.';
disp(s);
s1 = 'Введите <1> - для обработки одного цифрового изображения\n';
s2 = 'Введите <2> - для обработки двух цифровых изображений\n';
nimages = input([s1, s2]);
if nimages ~= 1 && nimages ~= 2
ok = false;
while ok == false
nimages = input([s1, s2]);
if nimages == 1 || nimages == 2
ok = true;
end;
end;
end;
if nimages == 1
s = 'Введите имя файла (в одинарных кавычках) с цифровым изображением: ';
fname1 = input(s);
elseif nimages == 2
s='Введите имя файла (в одинарных кавычках) с первым цифровым изображе-
нием: ';
fname1 = input(s);
s='Введите имя файла (в одинарных кавычках) со вторым цифровым изображе-
нием: ';
fname2 = input(s);
end
if nimages == 1
image1 = imread(fname1);
info1 = imfinfo(fname1);

```

```

w1 = info1.Width; h1 = info1.Height;
figure; hImg1 = imshow(image1);
ssize1 = sprintf('Размеры изображения (Ш x В): %d x %d, пикс.', w1, h1);
title(ssize1);
elseif nimages == 2
image1 = imread(fname1);
info1 = imfinfo(fname1);
w1 = info1.Width; h1 = info1.Height;
subplot(1,2,1); hImg1 = imshow(image1);
ssize1 = sprintf('Размеры изображения (Ш x В):\n %d x %d, пикс.', w1, h1);
title(ssize1);
image2 = imread(fname2);
info2 = imfinfo(fname2);
w2 = info2.Width; h2 = info2.Height;
subplot(1,2,2); hImg2 = imshow(image2);
ssize2 = sprintf('Размеры изображения (Ш x В):\n %d x %d, пикс.', w2, h2);
title(ssize2);
end
s1 = 'Введите <1> - для автоматического распознавания меток\n';
s2 = 'Введите <2> - для распознавания меток в ручном режиме\n';
scan_mode = input([s1, s2]);
if scan_mode ~= 1 && scan_mode ~= 2
ok = false;
while ok == false
scan_mode = input([s1, s2]);
if scan_mode == 1 || scan_mode == 2
ok = true;
end;
end;
end;

```

```

if scan_mode == 1
s1 = 'Введите максимально допустимый горизонтальный размер меток ';
s2 = 'изображения (в пикселях).\n';
s3 = 'При введении 0, по умолчанию, будет задан размер, равный 200): ';
lenx = input([s1,s2,s3]);
s1 = 'Введите максимально допустимый вертикальный размер меток ';
s2 = 'изображения (в пикселях).\n';
s3 = 'При введении 0, по умолчанию, будет задан размер, равный 200): ';
leny = input([s1,s2,s3]);
s1 = 'Введите максимально допустимую разницу в яркости пикселей ';
s2 = '[0 - 255] между изображением\пметки и фоном. При введении 0, ';
s3 = 'по умолчанию, будет задана разница, равная 20: ';
dcolor = input([s1,s2,s3]);
if lenx == 0
lenx = 200;
end;
if leny == 0
leny = 200;
end;
if dcolor == 0
dcolor = 20;
end;
elseif scan_mode == 2
s1='Введите <1> - для отображения маркеров распознаваемых меток\n';
s2='Введите <2> - для распознавания меток без отображения их маркеров\n';
marker_mode = input([s1, s2]);
if marker_mode ~= 1 && marker_mode ~= 2
ok = false;
while ok == false
marker_mode = input([s1, s2]);

```

```

if marker_mode == 1 || marker_mode == 2
    ok = true;
end;
end;
end;
end
s = 'Задайте значение надежности, a = ';
alpha = input(s);
npoints1 = 0;
npoints2 = 0;
nrc = get(0,'RecursionLimit') / 2;
left = true;
counter = 0;
ncoefs = 0;
while true
    [x,y,btn] = ginput(1);
    x = round(x);
    y = round(y);
    if nimages == 1
        w = w1; h = h1; image = image1;
    elseif nimages == 2
        if left == true
            w = w1; h = h1; image = image1;
        else
            w = w2; h = h2; image = image2;
        end
    end
    if x < 0 || x > w || y < 0 || y > h
        break;
    else

```

```

if scan_mode == 1
arxt = 0; aryt = 0;
color = image(y, x, 1);
arxt(1) = x; aryt(1) = y;
[arxt(2), aryt(2)] = scan_line(arxt(1), aryt(1), -1, 0, w, h, color,...
dcolor, image);
[arxt(3), aryt(3)] = scan_line(arxt(1), aryt(1), 1, 0, w, h, color,...
dcolor, image);
[arxt(4), aryt(4)] = scan_line(arxt(1), aryt(1), 0, -1, w, h, color,...
dcolor, image);
[arxt(5), aryt(5)] = scan_line(arxt(1), aryt(1), 0, 1, w, h, color,...
dcolor, image);
arxt(6) = arxt(2) + arxt(3) - arxt(1) - 1; aryt(6) = aryt(1);
[arxt(7), aryt(7)] = scan_line(arxt(6), aryt(6), 0, -1, w, h, color,...
dcolor, image);
[arxt(8), aryt(8)] = scan_line(arxt(6), aryt(6), 0, 1, w, h, color,...
dcolor, image);
aryt(9) = aryt(4) + aryt(5) - aryt(1) + 1; arxt(9) = arxt(1);
aryt(10) = aryt(7) + aryt(8) - aryt(6) + 1; arxt(10) = arxt(6);
[arxt(11), aryt(11)] = scan_line(arxt(10), aryt(10), -1, 0, w, h,...
color, dcolor, image);
[arxt(12), aryt(12)] = scan_line(arxt(9), aryt(9), 1, 0, w, h,...
color, dcolor, image);
[arxt(13), aryt(13)] = scan_line(arxt(1), aryt(1), -1, -1, w, h,...
color, dcolor, image);
[arxt(14), aryt(14)] = scan_line(arxt(1), aryt(1), 1, 1, w, h,...
color, dcolor, image);
[arxt(15), aryt(15)] = scan_line(arxt(1), aryt(1), -1, 1, w, h,...
color, dcolor, image);
[arxt(16), aryt(16)] = scan_line(arxt(1), aryt(1), 1, -1, w, h,...

```



```

color, dcolor, image);
[arxt(17), aryt(17)] = scan_line(arxt(6), aryt(6), -1, -1, w, h,...
color, dcolor, image);
[arxt(18), aryt(18)] = scan_line(arxt(6), aryt(6), 1, 1, w, h,...
color, dcolor, image);
[arxt(19), aryt(19)] = scan_line(arxt(6), aryt(6), -1, 1, w, h,...
color, dcolor, image);
[arxt(20), aryt(20)] = scan_line(arxt(6), aryt(6), 1, -1, w, h,...
color, dcolor, image);
[arxt(21), aryt(21)] = scan_line(arxt(10), aryt(10), -1, -1, w, h,...
color, dcolor, image);
[arxt(22), aryt(22)] = scan_line(arxt(10), aryt(10), 1, 1, w, h,...
color, dcolor, image);
[arxt(23), aryt(23)] = scan_line(arxt(10), aryt(10), -1, 1, w, h,...
color, dcolor, image);
[arxt(24), aryt(24)] = scan_line(arxt(10), aryt(10), 1, -1, w, h,...
color, dcolor, image);
[arxt(25), aryt(25)] = scan_line(arxt(9), aryt(9), -1, -1, w, h,...
color, dcolor, image);
[arxt(26), aryt(26)] = scan_line(arxt(9), aryt(9), 1, 1, w, h,...
color, dcolor, image);
[arxt(27), aryt(27)] = scan_line(arxt(9), aryt(9), -1, 1, w, h,...
color, dcolor, image);
[arxt(28), aryt(28)] = scan_line(arxt(9), aryt(9), 1, -1, w, h,...
color, dcolor, image);
x_accum = 0; y_accum = 0;
for i = 1:28
x_accum = x_accum + arxt(i);
y_accum = y_accum + aryt(i);
end

```

```

arxt(29) = round(x_accum / 28); aryt(29) = round(y_accum / 28);
[arxt(30), aryt(30)] = scan_line(arxt(29), aryt(29), -1, 0, w, h,...
color, dcolor, image);
[arxt(31), aryt(31)] = scan_line(arxt(29), aryt(29), 1, 0, w, h,...
color, dcolor, image);
[arxt(32), aryt(32)] = scan_line(arxt(29), aryt(29), 0, 1, w, h,...
color, dcolor, image);
[arxt(33), aryt(33)] = scan_line(arxt(29), aryt(29), 0, -1, w, h,...
color, dcolor, image);
[arxt(34), aryt(34)] = scan_line(arxt(29), aryt(29), -1, -1, w, h,...
color, dcolor, image);
[arxt(35), aryt(35)] = scan_line(arxt(29), aryt(29), 1, 1, w, h,...
color, dcolor, image);
[arxt(36), aryt(36)] = scan_line(arxt(29), aryt(29), -1, 1, w, h,...
color, dcolor, image);
[arxt(37), aryt(37)] = scan_line(arxt(29), aryt(29), 1, -1, w, h,...
color, dcolor, image);
arxt(38) = arxt(29); aryt(38) = aryt(10);
arxt(39) = arxt(9); aryt(39) = aryt(29);
arxt(40) = arxt(29); aryt(40) = aryt(6);
arxt(41) = arxt(10); aryt(41) = aryt(29);
[arxt(42), aryt(42)] = scan_line(arxt(38), aryt(38), -1, -1, w, h,...
color, dcolor, image);
[arxt(43), aryt(43)] = scan_line(arxt(38), aryt(38), 1, 1, w, h,...
color, dcolor, image);
[arxt(44), aryt(44)] = scan_line(arxt(38), aryt(38), -1, 1, w, h,...
color, dcolor, image);
[arxt(45), aryt(45)] = scan_line(arxt(38), aryt(38), 1, -1, w, h,...
color, dcolor, image);
[arxt(46), aryt(46)] = scan_line(arxt(39), aryt(39), -1, -1, w, h,...

```

```

color, dcolor, image);
[arxt(47), aryt(47)] = scan_line(arxt(39), aryt(39), 1, 1, w, h,...
color, dcolor, image);
[arxt(48), aryt(48)] = scan_line(arxt(39), aryt(39), -1, 1, w, h,...
color, dcolor, image);
[arxt(49), aryt(49)] = scan_line(arxt(39), aryt(39), 1, -1, w, h,...
color, dcolor, image);
[arxt(50), aryt(50)] = scan_line(arxt(40), aryt(40), -1, -1, w, h,...
color, dcolor, image);
[arxt(51), aryt(51)] = scan_line(arxt(40), aryt(40), 1, 1, w, h,...
color, dcolor, image);
[arxt(52), aryt(52)] = scan_line(arxt(40), aryt(40), -1, 1, w, h,...
color, dcolor, image);
[arxt(53), aryt(53)] = scan_line(arxt(40), aryt(40), 1, -1, w, h,...
color, dcolor, image);
[arxt(54), aryt(54)] = scan_line(arxt(41), aryt(41), -1, -1, w, h,...
color, dcolor, image);
[arxt(55), aryt(55)] = scan_line(arxt(41), aryt(41), 1, 1, w, h,...
color, dcolor, image);
[arxt(56), aryt(56)] = scan_line(arxt(41), aryt(41), -1, 1, w, h,...
color, dcolor, image);
[arxt(57), aryt(57)] = scan_line(arxt(41), aryt(41), 1, -1, w, h,...
color, dcolor, image);
nxt = 57;
[arxt, ix] = sort(arxt);aryt2 = 0;
for i = 1:nxt
    aryt2(i) = aryt(ix(i));
end
for i = 1:nxt
    aryt(i) = aryt2(i);

```

```

end
n = 1; n1 = 1; x = arxt(n1);
while n <= (nxt + 1)
if n == (nxt + 1) || arxt(n) ~= x
yt = 0;
for nk = n1:n-1
yt(nk-n1+1) = aryt(nk);
end
yt = sort(yt);
for nk = n1:n-1
aryt(nk) = yt(nk-n1+1);
end
if n <= nxt
n1 = n; x = arxt(n1);
end
end
n = n + 1;
end
min_x = min(arxt); max_x = max(arxt);
lx = max_x - min_x;
min_y = min(aryt); max_y = max(aryt);
ly = max_y - min_y;
if lx > lenx || ly > leny
s1 = 'Размеры сканируемого объекта превышают заданные максимально ';
s2 = 'допустимые. Выполните повторный щелчок мышью.';
message = [s1, s2];
hmb = msgbox(message,'fdensity','error','modal');
uiwait(hmb);
continue;
end

```

```

npixels = 0;%tStart=tic;
arx = 0;
ary = 0;
scan_image = zeros(h,w);
for n = 1:nxt
    npixels_t = 0;
    arx_t = 0;
    ary_t = 0;
    first = true;
    [npixels_t, scan_image, arx_t, ary_t] = scan(nrc, arxt(n), aryt(n),...
    image, scan_image, w, h, color, dcolor, npixels_t, arx_t, ary_t);
    while npixels_t > 0
        j = 1;
        for i = npixels+1:npixels+npixels_t;
            arx(i) = arx_t(j);
            ary(i) = ary_t(j);
            j = j + 1;
        end
        npixels = npixels + npixels_t;
        [maximum, i] = max(arx_t);
        x = arx_t(i) + 1;
        y = ary_t(i) + 1;
        npixels_t = 0;
        arx_t = 0;
        ary_t = 0;
        [npixels_t, scan_image, arx_t, ary_t] = scan(nrc, x, y, image,...
        scan_image, w, h, color, dcolor, npixels_t, arx_t, ary_t);
    end
end;
x_accum = 0; y_accum = 0;

```

```

for i = 1:npixels
x_accum = x_accum + arx(i);
y_accum = y_accum + ary(i);
end
for i=1:npixels
line(arx(i), ary(i), 'Color', 'red', 'LineWidth', 1);
end
if left == true
npoints1 = npoints1 + 1;
arxc1(npoints1) = round(x_accum / npixels);
aryc1(npoints1) = round(y_accum / npixels);
else
npoints2 = npoints2 + 1;
arxc2(npoints2) = round(x_accum / npixels);
aryc2(npoints2) = round(y_accum / npixels);
end
elseif scan_mode == 2
if left == true
npoints1 = npoints1 + 1;
arxc1(npoints1) = x;
aryc1(npoints1) = y;
else
npoints2 = npoints2 + 1;
arxc2(npoints2) = x;
aryc2(npoints2) = y;
end
if marker_mode == 1
line(x, y, 'Color', 'red', 'Marker', 'o');
elseif marker_mode == 2
line(x, y, 'Color', 'red');

```

```

end
end
end
counter = counter + 1;
if counter == 2
if left == true
line([arxc1(npoints1-1) arxc1(npoints1)], [aryc1(npoints1-1)...
aryc1(npoints1)], 'Color', 'green', 'LineWidth', 1);
line(arxc1(npoints1-1), aryc1(npoints1-1), 'Color', 'green',...
'Marker', 'x');
line(arxc1(npoints1), aryc1(npoints1), 'Color', 'green',...
'Marker', 'x');
text((arxc1(npoints1-1) + arxc1(npoints1)) / 2, (aryc1(npoints1-1)...
+ aryc1(npoints1)) / 2, sprintf('%d', npoints1 / 2),...
'Color', 'green', 'FontSize', 12);
else
line([arxc2(npoints2-1) arxc2(npoints2)], [aryc2(npoints2-1)...
aryc2(npoints2)], 'Color', 'blue', 'LineWidth', 1);
line(arxc2(npoints2-1), aryc2(npoints2-1), 'Color', 'blue',...
'Marker', 'x');
line(arxc2(npoints2), aryc2(npoints2), 'Color', 'blue',...
'Marker', 'x');
text((arxc2(npoints2-1) + arxc2(npoints2)) / 2, (aryc2(npoints2-1)...
+ aryc2(npoints2)) / 2, sprintf('%d', npoints2 / 2),...
'Color', 'blue', 'FontSize', 12);
end
left = ~left;
counter = 0;
end
if mod((npoints1 + npoints2), 4) == 0

```

```

len1 = sqrt((arxc1(npoints1) - arxc1(npoints1-1))^2 +...
(aryc1(npoints1) - aryc1(npoints1-1))^2);
len2 = sqrt((arxc2(npoints2) - arxc2(npoints2-1))^2 +...
(aryc2(npoints2) - aryc2(npoints2-1))^2);
ncoefs = ncoefs + 1;
arcoefs(ncoefs) = ((len1 / len2)^3 - 1) * 100;
s = sprintf('p%d = %.4f %%', ncoefs, arcoefs(ncoefs)); disp(s);
end
end;
cmin = min(arcoefs);
cmax = max(arcoefs);
cmean = mean(arcoefs);
cstd = std(arcoefs, 1);
if cstd == 0 & cmean == 0
cvar = 0;
elseif cstd ~= 0 & cmean == 0
cvar = Inf * sign( cstd );
else
cvar = cstd / cmean * 100;
end;
h = (cmax - cmin) / ncoefs;
m1 = abs(cmean - cmin);
m2 = abs(cmax - cmean);
amax = max(m1, m2);
s = sprintf('<p> = %.4f %%', cmean); disp(s);
cstudent = tinvc((1+alpha)/2, ncoefs-1);%вычисление коэф. Стьюдента
dp = cstd * cstudent;
s = sprintf('Ответ: p = %.4f +- %.4f %%\n', cmean, dp); disp(s);
fout=input('Записать результаты измерений в файл: <1> - Да, <0> - Нет ? ');
if fout ~= 0 && fout ~= 1

```



```

ok = false;
while ok == false
fout=input('Записать результаты измерений в файл: <1> - Да, <0> - Нет ? ');
if fout == 0 || fout == 1
ok = true;
end;
end;
end;
if ((ncoefs > 0) & (fout == 1))
for i=1:ncoefs
data_out(i,1) = i;
data_out(i,2) = arcoefs(i);
end
[fid, message] = fopen('fdensity_out.txt','w');
fprintf(fid,'%4s %8s\n', '№', 'p');
fprintf(fid,'%4d %8.4f\n', data_out');
fclose(fid);
end;
s1='Введите число интервалов (бинов) для построения гистограммы ';
s2='измерений\n(при введении 0 число бинов определяется автоматически): ';
nbins = input([s1, s2]);
if nbins == 0
[F,x]=ecdf(arcoefs);
nbins=length(x)-1;
end;
s11 = 'Введите <1> - для построения нормальной кривой по ';
s12 = 'результатам измерений\n';
s1 = sprintf([s11, s12]);
s21 = 'Введите <2> - для отображения гистограммы без нормальной ';
s22 = 'кривой\n';

```

```

s2 = sprintf([s21,s22]);
norm = input([s1, s2]);
if norm ~= 1 && norm ~= 2
    ok = false;
while ok == false
    norm = input([s1, s2]);
if norm == 1 || norm == 2
    ok = true;
end;
end;
end;
figure;
bins = hist(arcoefs,nbins);
hist(arcoefs,nbins); hold on;
max_bins = max(bins);
kurt = kurtosis(arcoefs) - 3;
skew = skewness(arcoefs, 1);
if norm == 1
    x=(cmean - amax):(h/100):(cmean + amax);
    gauss = gaussmf(x,[cstd cmean]) * max_bins;
    plot(x,gauss,'red');
end;
xlabel('p, %');
ylabel('Частота');
figure;
normplot(arcoefs);
s1 = sprintf('Статистические характеристики измерений p:\n');
s2 = sprintf('1. Минимальное значение  = %.4f %%\n', cmin);
s3 = sprintf('2. Максимальное значение = %.4f %%\n', cmax);
s4 = sprintf('3. Среднее значение     = %.4f %%\n', cmean);

```

```

s5 = sprintf('4. Стандартное отклонение = %.4f %%\n', cstd);
s6 = sprintf('5. Коэффициент вариации = %.6f %%\n', cvar);
s7 = sprintf('6. Асимметрия = %.6f\n', skew);
s8 = sprintf('7. Эксцесс = %.6f\n', kurt);
disp([s1, s2, s3, s4, s5, s6, s7, s8]);
OK = true;
catch
s1 = 'Введены некорректные данные, либо все измерения р равны, либо ';
s2 = 'число измерений р меньше двух. Повторите ввод.';
message = [s1, s2];
hmb = msgbox(message,'fdensity','error','modal');
uiwait(hmb);
OK = false;
end
end

```

#### **Приложение 5. Листинг файл функции “colorcmp”**

```

function equal = colorcmp(color1, color2, dcolor)
if color1 > color2
dcolor2 = color1 - color2;
else
dcolor2 = color2 - color1;
end;
if dcolor2 <= dcolor
equal = true;
else
equal = false;
end
end

```

### Приложение 6. Листинг файл функции “scan\_line”

```
function [x, y] = scan_line(x, y, dx, dy, w, h, color, dcolor, image)
ok = true;
while ok == true && x > 0 && x <= w && y > 0 && y <= h
if colorcmp(color, image(y, x, 1), dcolor) == true
x = x + dx; y = y + dy;
else
ok = false;
end
end
x = x - dx; y = y - dy;
end
```

### Приложение 7. Листинг файл функции “scan”

```
function [npixels, scan_image, arx, ary] = scan(nrc, x, y, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary)
if npixels >= nrc || x <= 0 || x > w || y <= 0 || y > h...
|| scan_image(y,x) == 1
return;
end
scan_image(y,x) = 1;
if colorcmp(color, image(y, x, 1), dcolor) == false
return;
end
npixels = npixels + 1;
arx(npixels) = x;
ary(npixels) = y;
dx = -1; dy = -1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image, ...
scan_image, w, h, color, dcolor, npixels, arx, ary);
```

```

dy = 0;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dy = 1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dx = 0; dy = -1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dy = 1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dx = 1; dy = -1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dy = 0;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
dy = 1;
[npixels, scan_image, arx, ary] = scan(nrc, x+dx, y+dy, image,...
scan_image, w, h, color, dcolor, npixels, arx, ary);
end

```

## СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ.....	4
2. ОБЩИЕ ПОЛОЖЕНИЯ.....	4
3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	11
4. КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ.....	12
СПИСОК ЛИТЕРАТУРЫ.....	14
ПРИЛОЖЕНИЯ.....	14
Приложение 1. Таблица значений коэффициентов Стьюдента .....	14
Приложение 2. Некоторые функции в системе MATLAB, применяемые при обработке результатов измерений и при работе с цифровыми изображениями.....	25
Приложение 3 Файл-функция “fdensity”. Руководство пользователя.....	28
3.1. Работа программы “fdensity” в автоматическом режиме.....	28
3.2. Работа программы “fdensity” в ручном режиме.....	32
Приложение 4 Листинг программы “fdensity” .....	32
Приложение 5. Листинг файл функции “colorcmp” .....	
Приложение 6. Листинг файл функции “scan_line” .....	
Приложение 7. Листинг файл функции “scan” .....	

## Навчальне видання

**Методичні вказівки** до виконання лабораторної роботи за темою “**Определение статистических характеристик частиц по микрофотографиям**”  
по курсу “Електронна мікроскопія та електроннографія”  
для студентів фізико-технічного, механіко-технологічного та інженерно-фізичного  
факультетів

Російською мовою

Укладачі      **Багмут** Олександр Григорович  
                    **Багмут** Іван Олександрович  
                    **Ніколайчук** Григорій Павлович

Роботу до видання рекомендувала      К.Т. Лемешевська  
Відповідальний за випуск                  О.Г. Багмут

В авторській редакції

План 2011 р., поз. 101

Підп. до друку . . . р.	Формат 60x84 1/16.	Папір офсет. №2
Riso-друк.	Гарнітура Таймс.	Ум. друк. арк. , .
Облік.-вид. арк. 1,8		
Наклад 50 прим.	Зам. №	Ціна договірна

---

Видавничий центр НТУ “ХПІ”

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.

61002, Харків, вул. Фрунзе, 21

---

Друкарня НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21