

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторной работы
“Определение индексов оси зоны кристаллических плоскостей”
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

Харьков
НТУ “ХПИ”
2011

Методические указания к выполнению лабораторной работы
“Определение индексов оси зоны кристаллических плоскостей” по курсу
“Электронная микроскопия и электронография” для студентов физико-
технического, механико-технологического и инженерно-физического фа-
культетов / Сост. А.Г. Багмут, И.А. Багмут, Г.П. Николайчук и др. –
Харьков: НТУ “ХПИ”, 2011. – 28 с.

Составители: А.Г. Багмут,
И.А. Багмут,
Г.П. Николайчук,
В.А. Жучков,
Е.Г. Борик

Рецензент А.Т. Пугачев

Кафедра теоретической и экспериментальной физики

I. ЦЕЛЬ РАБОТЫ

Определить направление оси кристаллографической зоны по электронограмме объекта с известной кристаллической структурой.

2. ОБЩИЕ ПОЛОЖЕНИЯ

Просвечивающая электронная микроскопия и электронография в совокупности образуют единый исследовательский комплекс, основанный на дифракции электронов. Электронно-микроскопические изображения кристаллических объектов необходимо дополнять индцированными картинками электронной дифракции. Они позволяют судить о фазовом составе и об ориентировке объекта в целом (по электронограмме), а также о фазовом составе и об ориентировке его выделенных локальных участков (по картинкам микро-дифракции). Определение направления оси кристаллографической зоны эквивалентно определению ориентировки изучаемого кристалла.

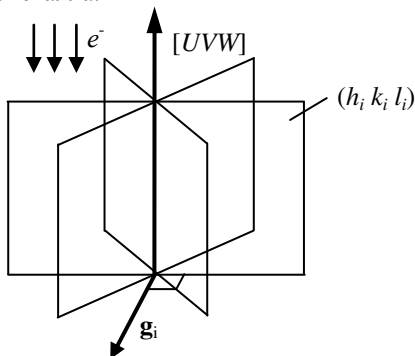


Рис. 1. К определению кристаллографической зоны

Кристаллографической зоной называется совокупность плоскостей, пересекающихся по одной линии (или параллельным линиям). Эта линия называется осью зоны (рис. 1). В кристаллической решетке направление оси зоны задается тремя индексами $[UVW]$. Плоскость принадлежит зоне, если направление оси зоны лежит в этой плоскости. Нормали к плоскостям зоны с осью $[UVW]$ задают пучок направлений, каждое из которых перпендикулярно $[UVW]$.

Ось зоны зададим вектором кристаллической решетки \mathbf{r}_{UVW} :

$$\mathbf{r}_{UVW} = U\mathbf{a} + V\mathbf{b} + W\mathbf{c}. \quad (1)$$

В выражении (1) векторы \mathbf{a} , \mathbf{b} , \mathbf{c} являются осевыми векторами кристаллической решетки. U , V , W есть координаты вектора решетки, выраженные в долях соответствующего периода по каждой оси.

Нормаль к i -ой плоскости с индексами Миллера $(h_i k_i l_i)$, которая

принадлежит зоне с осью $[UVW]$, представим вектором обратной решетки \mathbf{g}_i :

$$\mathbf{g}_i = h_i \mathbf{a}^* + k_i \mathbf{b}^* + l_i \mathbf{c}^*. \quad (2)$$

В выражении (2) векторы обратной решетки \mathbf{a}^* , \mathbf{b}^* , \mathbf{c}^* являются осевыми векторами обратной решетки:

$$\mathbf{a}^* = \frac{[\mathbf{b}, \mathbf{c}]}{\Omega}, \quad \mathbf{b}^* = \frac{[\mathbf{c}, \mathbf{a}]}{\Omega}, \quad \mathbf{c}^* = \frac{[\mathbf{a}, \mathbf{b}]}{\Omega}, \quad (3)$$

где объем элементарной ячейки $\Omega = (\mathbf{a}[\mathbf{b}, \mathbf{c}])$. Между осевыми векторами атомной и обратной решеток выполняются соотношения:

$$(\mathbf{a}, \mathbf{a}^*) = (\mathbf{b}, \mathbf{b}^*) = (\mathbf{c}, \mathbf{c}^*) = 1; \quad (4a)$$

$$(\mathbf{a}, \mathbf{b}^*) = (\mathbf{b}, \mathbf{c}^*) = (\mathbf{a}, \mathbf{c}^*) = 0. \quad (4b)$$

Поскольку нормаль к плоскости перпендикулярна любому лежащему в ней направлению, а \mathbf{r}_{UVW} лежит во всех плоскостях зоны, то скалярное произведение $(\mathbf{r}_{UVW}, \mathbf{g}_i) = 0$. Это соотношение в развернутом виде с учетом (4a) и (4b) записывается как:

$$h_i U + k_i V + l_i W = 0. \quad (5)$$

Соотношение (5) носит название “закон зоны”. Оно выполняется для любой плоскости зоны $[UVW]$.

Линия пересечения двух плоскостей с индексами Миллера $(h_1 \ k_1 \ l_1)$ и $(h_2 \ k_2 \ l_2)$ является осью зоны этих плоскостей и задается направлением $[UVW]$ в кристаллической решетке. Векторы обратной решетки $\mathbf{g}_1 = h_1 \mathbf{a}^* + k_1 \mathbf{b}^* + l_1 \mathbf{c}^*$ и $\mathbf{g}_2 = h_2 \mathbf{a}^* + k_2 \mathbf{b}^* + l_2 \mathbf{c}^*$ ориентированы перпендикулярно плоскостям $(h_1 \ k_1 \ l_1)$ и $(h_2 \ k_2 \ l_2)$ соответственно. Поэтому индексы $[UVW]$ оси зоны в кристаллической решетке можно найти, записав векторное произведение $[\mathbf{g}_1, \mathbf{g}_2]$:

$$[\mathbf{g}_1, \mathbf{g}_2] = \Omega^{-1} [\mathbf{a}(k_1 l_2 - k_2 l_1) + \mathbf{b}(l_1 h_2 - l_2 h_1) + \mathbf{c}(h_1 k_2 - h_2 k_1)]. \quad (6)$$

В выражении (6) индексами направлений $[UVW]$ являются коэффициенты при осевых векторах \mathbf{a} , \mathbf{b} и \mathbf{c} :

$$U = k_1 l_2 - k_2 l_1, \quad (7a)$$

$$V = l_1 h_2 - l_2 h_1, \quad (7b)$$

$$W = h_1 k_2 - h_2 k_1. \quad (7b)$$

Формулу (6) удобно записать в виде определителя:

$$\Omega^{-1} \begin{vmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \\ h_1 & k_1 & l_1 \\ h_2 & k_2 & l_2 \end{vmatrix} \quad (8)$$

Есть простой прием вычисления индексов направления оси зоны, являющейся линией пересечения двух плоскостей с индексами $(h_1k_1l_1)$ и $(h_2k_2l_2)$. В соответствии со схемой (9) индексы плоскостей запишем дважды в двух строках:

$$\begin{array}{ccccccc} h_1 & | & k_1 & l_1 & h_1 & k_1 & | & l_1 \\ h_2 & | & k_2 & l_2 & h_2 & k_2 & | & l_2 \end{array} \quad (9)$$

Далее отбросим крайние индексы слева и справа. Искомые индексы оси зоны получим как разности перекрестных произведений, указанных стрелками. В развернутом виде они выписаны в выражениях (7а), (7б) и (7в).

Дифракционные векторы \mathbf{g}_1 и \mathbf{g}_2 на электронограмме следует выбирать так, чтобы получить ось зоны направленной от центрального рефлекса к исследователю (т.е. антипараллельно направлению движения электронного луча e^- (рис. 1)). В этом случае второй из перемножаемых векторов должен быть повернут относительно первого на угол менее 180° против часовой стрелки.

Выбору дифракционных векторов \mathbf{g}_1 и \mathbf{g}_2 предшествует процедура индирования электронограммы (или картины микродифракции). Электронограмма является плоским сечением обратной решетки. При этом каждый дифракционный максимум (рефлекс) на ней соответствует узлу обратной решетки, положение которого задается дифракционным вектором \mathbf{g} . Введем следующие обозначения. R - расстояние от центра электронограммы до заданного рефлекса. L - расстояние между образцом и фотопластинкой (или экраном), называемое длиной дифракционной камеры. При анализе картин микродифракции под L подразумевают т. н. эффективную длину дифракционной камеры, учитывающую движение электронов в поле электромагнитных линз. λ - длина волны де Бройля электронов при установленном ускоряющем напряжении U . d - межплоскостное расстояние серии плоскостей кристаллической решетки, ответственных за появление рассматриваемого рефлекса. Между d и \mathbf{g} существует соотношение:

$$d = \frac{1}{|\mathbf{g}|}. \quad (10)$$

Из простых геометрических соображений следует, что

$$Rd = \lambda L. \quad (11)$$

Величина λL называется “постоянная прибора” и может быть определена при анализе электронограммы от эталонного объекта с известным значе-

нием d . Как правило, значение λ приводится в \AA , а значение L – в мм. Поэтому значение λL приводится в $\text{\AA}\cdot\text{мм}$.

В большинстве случаев исследователь имеет представление об изучаемом объекте. Поэтому будем считать, что состав, и следовательно, структура образца известны. В этом случае можно воспользоваться кристаллографическими таблицами *Joint Committee on Powder Diffraction Standards* (сокращенно *JCPDS*) и получить сведения о межплоскостных расстояниях и типе кристаллической решетки изучаемого объекта. Для каждого типа кристаллической решетки известны формулы, связывающие расстояния d и углы φ между плоскостями с параметрами решетки a , b и c , углами между главными осями элементарной ячейки α , β и γ и индексами Миллера h , k и l . Некоторые из этих формул приведены в приложении 1.

Дальнейшее изложение проведем на примере индирования электронограммы (точнее картины микро-дифракции) от кристаллического зародыша HfO_2 , растущего в аморфной пленке (рис. 2).

Исходные данные. Постоянная прибора $\lambda L = 30,77 \text{ \AA}\cdot\text{мм}$. Кристаллическая решетка зародыша соответствует ромбической модификации HfO_2 . Согласно таблице *JCPDS* (карта 21-0904) ее параметры имеют следующие значения: $a = 5,008 \text{ \AA}$, $b = 5,062 \text{ \AA}$ и $c = 5,223 \text{ \AA}$.

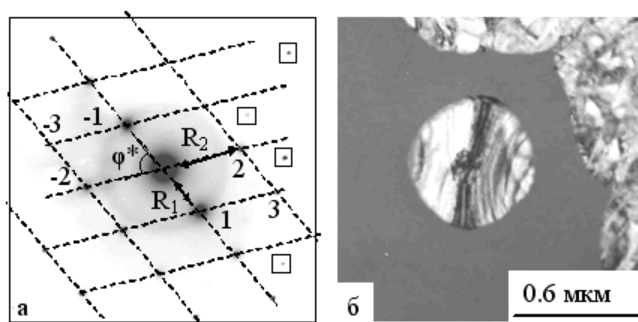


Рис. 2. Картина микродифракции (а) и электронномикроскопическое изображение (б) кристалла HfO_2 , растущего в аморфной матрице

1. Выделение на электронограмме рефлексов, принадлежащих одной кристаллографической зоне. Для выделения отражений, принадлежащих одной кристаллографической зоне с осью $[UVW]$, следует на изображении анализируемой электронограммы начертить узловую сетку, как

показано на рис. 2а. Рефлексы, принадлежащие зоне с осью $[UVW]$, должны располагаться в узлах этой сетки. Узловую сетку образуют две системы параллельных прямых, находящихся на одинаковом расстоянии одна от другой и проходящих через рефлексы электронограммы. Отражения, не совпадающие с точками пересечения прямых линий, не принадлежат данной кристаллографической зоне с осью $[UVW]$. На рис. 2а они находятся в центре нарисованных на электронограмме квадратиков. Их происхождение обусловлено наличием кристаллитов с иными ориентировками.

2. Определение межплоскостных расстояний. Следует измерить расстояния R между центральным пятном и другими дифракционными пятнами на электронограмме, положение которых совпадает с узлами нанесенной сетки. В простейшем случае достаточно и двух измерений R_1 и R_2 , выполненных для рефлексов, не лежащих на одной прямой, проходящей через центр электронограммы. С целью повышения точности результата следует измерить расстояние между двумя крайними пятнами, расположенными на прямой, проходящей через ряд пятен и центральное пятно. Полученный результат надо разделить на число пятен, расположенных на этой прямой, исключая центральное пятно. В нашем примере $R_1 = 10,5$ мм и $R_2 = 14,8$ мм. В соответствии с (11):

$$d_1 = \frac{\lambda L}{R_1} = \frac{30.77}{10.5} \text{ \AA} = 2.93 \text{ \AA}, \quad (12a)$$

$$d_2 = \frac{\lambda L}{R_2} = \frac{30.77}{14.8} \text{ \AA} = 2.08 \text{ \AA}. \quad (12b)$$

Заметим, что метод микродифракции позволяет определять межплоскостные расстояния в \AA с точностью до второго знака после запятой.

3. Определение углов между дифракционными векторами. С помощью транспортира измерим угол φ^* между линиями, проведенными из центрального рефлекса (000) в места локализации двух произвольных отражений в узлах сетки (рис. 2а). В нашем случае это угол между векторами R_1 и R_2 , равный 63° . Полагаем, что точность измерения составляет $\pm 0,5^\circ$.

4. Индексирование электронограммы (определение индексов Миллера рефлексов электронограммы). Индексы, приписываемые рефлексам электронограммы, должны верно отражать расстояния и углы между плоскостями исследуемого объекта.

а) Сопоставим найденные значения d_1 и d_2 с данными таблиц *JCPDS*. Согласно карте 21-0904 межплоскостное расстояние $d_{(111)} =$

2,94500 Å и $d_{(121)} = 2,06900$ Å. Округлим эти числа до второго знака после запятой: $d_{(111)} = 2,94$ Å и $d_{(121)} = 2,07$ Å. В пределах точности метода микродифракции измеренные значения d_1 и d_2 практически совпадают с табличными значениями $d_{(111)}$ и $d_{(121)}$ соответственно.

Значения межплоскостных расстояний d для ромбической решетки вычисляются по формуле:

$$d = \frac{1}{\sqrt{\left(\frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2}\right)}} \quad (13)$$

б) Сравним измеренное значение угла φ^* с расчетным значением угла φ между плоскостями (111) и (121). Для нашего случая ромбической решетки используем формулу (10п) приложения 1:

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{\frac{h_1 \cdot h_2}{a^2} + \frac{k_1 \cdot k_2}{b^2} + \frac{l_1 \cdot l_2}{c^2}}{\sqrt{\left(\frac{h_1^2}{a^2} + \frac{k_1^2}{b^2} + \frac{l_1^2}{c^2}\right)} \cdot \sqrt{\left(\frac{h_2^2}{a^2} + \frac{k_2^2}{b^2} + \frac{l_2^2}{c^2}\right)}} \right) \quad (14)$$

Подставим в (14) табличные значения параметров кристаллической решетки ($a = 5,008$ Å, $b = 5,062$ Å и $c = 5,223$ Å), а также индексы Миллера ($h_1 = 1, k_1 = 1, l_1 = 1, h_2 = 1, k_2 = 2, l_2 = 1$). В результате получим значение $\varphi = 19,5^\circ$, которое не совпадает с измеренным значением $\varphi^* = 63^\circ$. Следовательно, неправомерно приписать рефлексам 1 и 2 соответственно индексы (111) и (121).

в) Среди ряда целочисленных (положительных и отрицательных) значений $h_1, k_1, l_1, h_2, k_2, l_2$ надо выбрать те, которые при подстановке в (13) и (14) дадут такие величины межплоскостных расстояний и углов, которые в пределах точности метода совпадут с измеренными значениями d_1, d_2 и φ^* . Они и будут индексами Миллера (h_1, k_1, l_1) и (h_2, k_2, l_2) соответственно для рефлексов 1 и 2 электронограммы на рис. 2а. Поставленная задача легко решается в системе *MatLab* с помощью специально разработанной файл-функции “*fimiller*”. Руководство к использованию и текст программы *fimiller* приведены в приложении 2.

В таблице 1 приведены четыре равноправных (на данном этапе) варианта индексов Миллера, которые в пределах точности эксперимента обеспечивают совпадение измеренных и рассчитанных углов и межплос-

костных расстояний.

Таблица 1 - Варианты индексов отражений 1 и 2 электронограммы рис. 2а

Вариант	$\varphi(^{\circ})$	$d_1 (\text{\AA})$	$(h_1 k_1 l_1)$	$d_2 (\text{\AA})$	$(h_2 k_2 l_2)$	$[U V W]$
1	62.84	2,942	(1 1 1)	2,073	(-1 2 1)	$[-1 -2 3]$
2	62.84	2,942	(1 1 -1)	2,073	(-1 2 -1)	$[1 2 3]$
3	62.84	2,942	(-1 1 1)	2,073	(1 2 1)	$[-1 2 -3]$
4	62.84	2,942	(1 -1 1)	2,073	(-1 -2 1)	$[1 -2 -3]$

В качестве примера индцирования электронограммы используем вариант 1. В соответствии с таблицей отражению 1 припишем индексы (1 1 1), а отражению 2 – индексы (-1 2 1). Радиусы – векторы рефлексов 1, 2 и 3 образуют вместе со следом прямого пучка параллелограмм. Индексы отражения 3 находим по правилу векторного сложения, а именно:

$$h_3 = h_1 + h_2, \quad k_3 = k_1 + k_2, \quad l_3 = l_1 + l_2. \quad (15)$$

В соответствии с (15) отражение 3 индцируется как (0 3 2). Соотношения, подобные (15), должны выполняться для индексов всех отражений заданной кристаллографической зоны.

Обратная решетка кристалла всегда имеет центр инверсии в узле 000. Поэтому на электроннодифракционной картине рефлекс, симметричный относительно центра 000 рефлексу $(h k l)$, имеет индексы $(-h -k -l)$. Следовательно, отражение -1 на электронограмме рис. 2а индцируется как (-1 -1 -1), отражение -2 индцируется как (1 -2 1), отражение -3 индцируется как (0 -3 -2). Электронограмма, индцированная в соответствии с первым вариантом таблицы, показана на рис. 3.

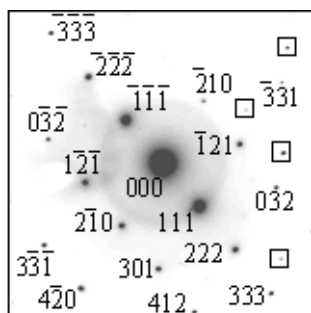


Рис. 3. Индцирование электронограммы, показанной на рис. 2 а (по первому варианту таблицы 1). Ось зоны $[-1 -2 3]$

5. Определение индексов направления оси зоны. Подставив значения $h_1 = 1$, $k_1 = 1$, $l_1 = 1$, $h_2 = -1$, $k_2 = 2$, и $l_2 = 1$ в (7) получим индексы направления оси зоны: $U = -1$, $V = -2$ и $W = 3$. Еще проще тот же результат можно получить, пользуясь схемой (9):

$$\begin{array}{c|ccc|c} 1 & 1 & 1 & 1 & 1 \\ \hline -1 & 2 & 1 & -1 & 2 \end{array} \rightarrow [-1 \ -2 \ 3]$$

Отметим, что найденная тройка индексов является одной из четырех возможных, приведенных в таблице 1 (седьмой столбец). По установленным индексам оси зоны можно судить о том, является ли данная плоскость обратной решеткой плоскостью симметрии этой решетки. Если да, то индентификация рефлексов можно считать законченным уже на этой стадии. Для ряда кристаллических решеток они приведены в таблице 2.

Таблица 2 - Плоскости обратной решетки, являющиеся плоскостями симметрии

№	Тип решетки	Плоскости симметрии обратной решетки
1	Кубическая	$\{100\}^*$ и $\{110\}^*$
2	Тетрагональная	$\{100\}^*$, $(001)^*$ и $\{110\}^*$
3	Гексагональная	$\{100\}^*$, $\{110\}^*$ и $(001)^*$
4	Ромбическая	$(100)^*$, $(010)^*$ и $(001)^*$

В соответствии с таблицей 2 в ромбической решетке направления $[-1 \ -2 \ 3]$, $[1 \ 2 \ 3]$, $[-1 \ 2 \ -3]$ и $[1 \ -2 \ -3]$ не являются осями симметрии четного (второго, четвертого или шестого) порядка. В этом случае однозначное индентификация рефлексов невозможно без привлечения дополнительных данных. Такими данными могут служить “побочные” отражения, связанные с пересечением сферой Эвальда узлов обратной решетки, расположенных над плоскостью основного ее сечения. В нашем случае такие данные отсутствуют.

Сформулируем ответ. Для предложенной электронограммы направление оси зоны может быть задано следующими индексами:

$$[-1 \ -2 \ 3], [1 \ 2 \ 3], [-1 \ 2 \ -3] \text{ или } [1 \ -2 \ -3].$$

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить точечную электронограмму (или картину микродифракции) от вещества с известной кристаллической структурой.
2. Выделить на электронограмме рефлексы, принадлежащие одной кристаллографической зоне.
3. Определить межплоскостные расстояния, соответствующие двум рефлексам рассматриваемой кристаллографической зоны.
4. Измерить угол между дифракционными векторами, задающими положение выбранных рефлексов на электронограмме.
5. Проиндцировать электронограмму.
6. Определить индексы направления оси зоны.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В каких случаях формируются точечные электронограммы?
2. Что такое кристаллографическая зона?
3. Как связаны осевые векторы обратной решетки с осевыми векторами кристаллической решетки?
4. Сформулируйте закон зоны.
5. Вычислите индексы направления оси зоны, являющейся линией пересечения двух плоскостей с индексами (3 1 5) и (1 0 2).
6. Как связаны дифракционный вектор и соответствующее ему межплоскостное расстояние?

СПИСОК ЛИТЕРАТУРЫ

1. Электронная микроскопия тонких кристаллов / Хирш П., Хови А., Николсон Р. и др. – М. : Мир, 1968. – 574 с.
2. Утевский Л.М. Дифракционная электронная микроскопия в металловедении / Л.М. Утевский. – М. : Металлургия, 1973. – 583 с.
3. Структура і фізичні властивості твердого тіла: Лабораторний практикум / Під ред. Л.С. Палатника. – К. : Вища шк., 1992. – 311 с.
4. Поршнев С.В. Matlab 7. Основы работы и программирования. / С.В. Поршнев – М. : БИНОМ, 2006. – 365 с.

ПРИЛОЖЕНИЯ

Приложение 1. Расчет межплоскостных расстояний $d(\text{\AA})$ и углов $\varphi(^{\circ})$

Кубическая система

$$d = \frac{1}{\sqrt{\left(\frac{h^2 + k^2 + l^2}{a^2}\right)}} \quad (1\text{п})$$

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{h_1 \cdot h_2 + k_1 \cdot k_2 + l_1 \cdot l_2}{\sqrt{(h_1^2 + k_1^2 + l_1^2) \cdot (h_2^2 + k_2^2 + l_2^2)}} \right) \quad (2\text{п})$$

Гексагональная система

$$d = \frac{1}{\sqrt{\left(\frac{4 \cdot (h^2 + h \cdot k + k^2)}{3 \cdot a^2} + \left(\frac{l}{c}\right)^2\right)}} \quad (3\text{п})$$

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{h_1 \cdot h_2 + k_1 \cdot k_2 + 0.5 \cdot (h_1 \cdot k_2 + h_2 \cdot k_1) + 0.75 \cdot \left(\frac{a}{c}\right)^2 \cdot l_1 \cdot l_2}{\sqrt{\left(h_1^2 + k_1^2 + h_1 \cdot k_1 + 0.75 \cdot \left(\frac{a}{c}\right)^2 \cdot l_1^2\right)}} \times \frac{1}{\sqrt{\left(h_2^2 + k_2^2 + h_2 \cdot k_2 + 0.75 \cdot \left(\frac{a}{c}\right)^2 \cdot l_2^2\right)}} \right) \quad (4\text{п})$$

Тетрагональная система.

$$d = \frac{1}{\sqrt{\left(\frac{h^2 + k^2}{a^2} + \left(\frac{l}{c}\right)^2\right)}} \quad (5\pi)$$

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{\frac{1}{a^2} (h_1 \cdot h_2 + k_1 \cdot k_2) + \left(\frac{1}{c}\right)^2 \cdot l_1 \cdot l_2}{\sqrt{\left(\frac{1}{a^2} \cdot (h_1^2 + k_1^2) + \left(\frac{l_1}{c}\right)^2\right) \cdot \left(\frac{1}{a^2} \cdot (h_2^2 + k_2^2) + \left(\frac{l_2}{c}\right)^2\right)}} \right) \quad (6\pi)$$

Ромбическая система.

$$d = \frac{1}{\sqrt{\left(\frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2}\right)}} \quad (7\pi)$$

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{\frac{h_1 \cdot h_2}{a^2} + \frac{k_1 \cdot k_2}{b^2} + \frac{l_1 \cdot l_2}{c^2}}{\sqrt{\left(\frac{h_1^2}{a^2} + \frac{k_1^2}{b^2} + \frac{l_1^2}{c^2}\right) \cdot \left(\frac{h_2^2}{a^2} + \frac{k_2^2}{b^2} + \frac{l_2^2}{c^2}\right)}} \right) \quad (8\pi)$$

Моноклинные система.

$$d = \frac{1}{\sqrt{\left(\frac{h^2}{a^2 \cdot (\sin(\beta))^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2 \cdot (\sin(\beta))^2} + \frac{-2 \cdot h \cdot l \cdot (\cos(\beta))}{a \cdot c \cdot (\sin(\beta))^2}\right)}} \quad ; \quad 9\pi)$$

$$\varphi = \frac{180}{\pi} \cdot \arccos \left(\frac{\frac{h_1 \cdot h_2}{a^2} + \frac{k_1 \cdot k_2 \cdot (\sin(\beta))^2}{b^2} + \frac{l_1 \cdot l_2}{c^2} + \frac{-(l_1 \cdot h_2 + h_1 \cdot l_2) \cdot \cos(\beta)}{a \cdot c}}{\sqrt{\left(\frac{h_1^2}{a^2} + \frac{k_1^2 \cdot (\sin(\beta))^2}{b^2} + \frac{l_1^2}{c^2} + \frac{-2 \cdot h_1 \cdot l_1 \cdot \cos(\beta)}{a \cdot c} \right)}} \times \frac{1}{\sqrt{\left(\frac{h_2^2}{a^2} + \frac{k_2^2 \cdot (\sin(\beta))^2}{b^2} + \frac{l_2^2}{c^2} + \frac{-2 \cdot h_2 \cdot l_2 \cdot \cos(\beta)}{a \cdot c} \right)}} \right) \cdot \quad (10\text{п})$$

Приложение 2. Файл-функция “*fimiller*”. Руководство пользователя

Предлагаемая программа составлена для работы в системе *MatLab*. Она облегчает процедуру индирования электронограммы от вещества с известными кристаллографическими параметрами. На экспериментальной электронограмме необходимо измерить угол φ между двумя дифракционными векторами, а также рассчитать соответствующие им межплоскостные расстояния d . После ввода в программу экспериментальных значений φ и d , а также табличных значений параметров кристаллической решетки, характеризующих образец, файл-функция “*fimiller*” предлагает варианты индексов Миллера ($h_1 k_1 l_1$) и ($h_2 k_2 l_2$) для двух выбранных рефлексов электронограммы.

Поясним работу программы на конкретном примере. Допустим, мы исследуем диоксид гафния (HfO_2). Для этого соединения, имеющего ромбическую решетку, параметры следующие: $a = 5,008$; $b = 5,062$ и $c = 5,223$ Å. Для двух выбранных на электронограмме отражений измеренные значения составляют: $\varphi = 63 \pm 0,3^\circ$, $d_1 = 2,93 \pm 0,01$ Å и $d_2 = 2,06 \pm 0,01$ Å. Необходимо решить, какие индексы следует приписать рассматриваемым рефлексам.

Задача решается следующим образом. Запускаем систему *MatLab*. В окне “*Command window*” набираем имя файл функции “*fimiller*”. В командном окне появляются строки.

Функция вычисления индексов Миллера

Введите $\langle I \rangle$ - для продолжения работы с текущими параметрами, $\langle 0 \rangle$ - для ввода новых параметров. Если ранее мы уже работали с этим веществом, то вводим “1”. Если нет - вводим “0”. Поскольку нам надо ввести новые параметры кристаллической решетки, то вводим “0”.

Задайте с помощью клавиатуры тип системы:

<1> - гексагональная система

<2> - кубическая система

<3> - моноклинная система

<4> - ромбическая система

<5> - тетрагональная система

Тип системы:

Исследуем вещество с ромбической кристаллической решеткой.

Вводим "4".

Введите (в ангстремах) значение параметра a =

Вводим табличное значение "5,008".

Введите(в ангстремах)значение параметра b =

Вводим табличное значение "5,052".

Введите(в ангстремах)значение параметра c =

Вводим табличное значение "5,223".

Введите значение угла между плоскостями (в градусах):

Вводим измеренное значение значения угла "63".

Введите максимальное отклонение значения угла между плоскостями (в градусах):

Вводим величину, равную погрешности измерения угла, в данном случае "0.3".

Введите минимальное возможное значение индексов Миллера:

В ходе выполнения программы осуществляется перебор чисел от максимального до минимального значения, из которых будут составлены пары индексов Миллера для рассматриваемых рефлексов. Если, например, ввести число "-2", то процесс начнется с рассмотрения индексов (-2 -2 -2). Вводим число "-2".

Введите максимальное возможное значение индексов Миллера:

Вводим число "2".

Задать ограничение для межплоскостных расстояний d : <1> - Да, <0> - Нет?

Эта команда дает возможность исключить из рассмотрения те индексы отражений, для которых $d_1 = 2,93 \text{ \AA} > d > d_2 = 2,06 \text{ \AA}$. Вводим "1".

Введите минимальное возможное значение для d :

Вводим "2,06".

Введите максимальное возможное значение для d :

Вводим "2,95".

Записать результаты расчета в файл: <1> - Да, <0> - Нет ?

При утвердительном ответе на вопрос команда создает файл с ре-

результатами расчета.

Вводим “1”.

После этого в окне “*Command window*” появляется результат работы программы.

result =

Индексы Миллера, ромбическая система

угол	(h1 k1 l1)	d1	(h2 k2 l2)	d2	СКИ1	СКИ2
62,84	-1 -1 -1	2,942	1 -2 -1	2,073	3	6
62,84	-1 -1 1	2,942	1 -2 1	2,073	3	6
62,84	-1 -2 -1	2,073	1 -1 -1	2,942	6	3
62,84	-1 -2 1	2,073	1 -1 1	2,942	6	3

Поясним результат. Первый столбец таблицы. Значение угла, вычисленное согласно соотношению (8п) приложения 2. Второй столбец – индексы, приписываемые одному из рефлексов. Третий столбец – межплоскостное расстояние соответствующее этому дифракционному вектору. Вычисляется согласно (7п) приложения 2. Четвертый столбец – индексы, приписываемые другому рефлексу. Пятый столбец – межплоскостное расстояние соответствующее этому дифракционному вектору. Шестой столбец (СКИ 1) – сумма квадратов индексов, приписываемых первому рефлексу. Седьмой столбец (СКИ 2) – сумма квадратов индексов, приписываемых второму рефлексу.

Согласно приведенной таблицы результатом работы файл функции “*fimiller*” является четыре равноправных варианта индирования двух выбранных рефлексов электронограммы.

Приложение 3. Листинг программы “*fimiller*”

1. function fimiller
2. fname = 'fimiller.ini'; [fid] = fopen(fname,'r'); if fid ~= -1
3. ini = fscanf(fid, '%g', [1 inf]); fclose(fid); stype = ini(1); switch stype
4. case 1
5. sstype = 'гексагональная'; a = ini(2); c = ini(3); a2 = a * a; a3 = a2 * a; ic2 = 1 / (c * c);
6. a2c075 = 0.75 * a2 * ic2; a234 = 4 / (3 * a2); case 2
7. sstype = 'кубическая'; a = ini(2); ia2 = 1 / (a * a); case 3; sstype =


```

'моноклинная';
8.  a = ini(2); b = ini(3); c = ini(4); beta = ini(5); beta = beta * (pi / 180);
9.  sin_beta = sin(beta); sin2_beta = sin_beta * sin_beta; cos_beta =
    cos(beta);
10. a2 = a * a; b2 = b * b; c2 = c * c; ia2 = 1 / a2; ib2 = 1 / b2; ic2 = 1 / c2;
11. cos_beta_ac = cos_beta / (a * c); cos_beta_ac_2 = 2 * cos_beta_ac;
12. sin2_beta_b2 = sin2_beta * ib2; ia2_sin2_beta = 1 / (a2 * sin2_beta);
13. ic2_sin2_beta = 1 / (c2 * sin2_beta);
14. cos_beta_ac_2_sin2_beta = cos_beta_ac_2 / sin2_beta;
15. case 4
16. sstype = 'ромбическая'; a = ini(2); b = ini(3); c = ini(4); a2 = a * a; b2
    = b * b;
17. c2 = c * c; ia2 = 1 / a2; ib2 = 1 / b2; ic2 = 1 / c2;
18. case 5
19. sstype = 'тетрагональная'; a = ini(2); c = ini(3); a2 = a * a; c2 = c * c;
    ia2 = 1 / a2;
20. ic2 = 1 / c2; end; end; param = 0;
21. s0 = '          Функция вычисления индексов Миллера';
22. disp([s0]); if fid ~= -1
23. s1 = 'Текущие параметры расчета\n';
24. s2 = sprintf(['Тип системы - ', sstype, '\n']);
25. switch sstype
26. case 1
27. s3 = sprintf('Значение параметра a = %3.4f ангстрем\n', a);
28. s4 = sprintf('Значение параметра c = %3.4f ангстрем\n', c);
29. s5 = "; s6 = "; case 2
30. s3 = sprintf('Значение параметра a = %3.4f ангстрем\n', a);
31. s4 = "; s5 = "; s6 = "; case 3;
32. s3 = sprintf('Значение параметра a = %3.4f ангстрем\n', a);
33. s4 = sprintf('Значение параметра b = %3.4f ангстрем\n', c);
34. s5 = sprintf('Значение параметра c = %3.4f ангстрем\n', a);
35. s6 = ...sprintf('Значение угла по данным кристаллографических
    таблиц=%3.4f град.\n', ... beta);
36. case 4
37. s3 = sprintf('Значение параметра a = %3.4f ангстрем\n', a);
38. s4 = sprintf('Значение параметра b = %3.4f ангстрем\n', c);
39. s5 = sprintf('Значение параметра c = %3.4f ангстрем\n', a);
40. s6 = ";
41. case 5

```

```

42. s3 = sprintf('Значение параметра a = %3.4f ангстрем\n', a);
43. s4 = sprintf('Значение параметра c = %3.4f ангстрем\n', c);
44. s5 = "; s6 = "; end;
45. s7 = ['Введите <1> - для продолжения работы с текущими пара-
    метрами,',... '<0> -
    для ввода новых параметров '];
46. param = input([s1, s2, s3, s4, s5, s6, s7]);
47. if param ~= 0 && param ~= 1
48. ok = false; while ok == false
49. param = input(['Введите <1> - для продолжения работы',...
    'с текущими параметрами, <0> - для ввода новых параметров ']);
50. if param == 0 || param == 1
51. ok = true;
52. end; end; end; end;
53. if param == 0
54. s2 = 'Задайте с помощью клавиатуры тип ситемы:\n';
55. s3 = '<1> - гексагональная система\n';
56. s4 = '<2> - кубическая система\n';
57. s5 = '<3> - моноклинная система\n';
58. s6 = '<4> - ромбическая система\n';
59. s7 = '<5> - тетрагональная система\n';
60. s8 = 'Тип системы: ';
61. stype = input([s2, s3, s4, s5, s6, s7, s8]);
62. if stype < 1 | stype > 5
63. ok = false;
64. while ok == false
65. stype = input(['Необходимо ввести число от 1 до 5.\n', s8]);
66. if stype >= 1 & stype <= 5
67. ok = true; end; end; end;
68. switch stype
69. case 1
70. sstype = 'гексагональная система';
71. a = input('Введите(в ангстремах)значение параметра a = ');
72. c = input('Введите(в ангстремах)значение параметра c = ');
73. a2 = a * a; a3 = a2 * a; ic2 = 1 / (c * c); a2c075 = 0.75 * a2 * ic2;
74. a234 = 4 / (3 * a2);
75. case 2
76. sstype = 'кубическая система';
77. a = input('Введите(в ангстремах)значение параметра a = ');

```

```

78. ia2 = 1 / (a * a);
79. case 3;
80. sstype = 'моноклинная система';
81. a = input('Введите(в ангстремах)значение параметра a = ');
82. b = input('Введите(в ангстремах)значение параметра b = ');
83. c = input('Введите(в ангстремах)значение параметра c = ');
84. beta = input(['Введите значение угла по данным',...
    'кристаллографических таблиц (в градусах) = ']);
85. beta = beta * (pi / 180);
86. sin_beta = sin(beta);
87. sin2_beta = sin_beta * sin_beta;
88. cos_beta = cos(beta);
89. a2 = a * a; b2 = b * b; c2 = c * c; ia2 = 1 / a2; ib2 = 1 / b2; ic2 = 1 / c2;
90. cos_beta_ac = cos_beta / (a * c); cos_beta_ac_2 = 2 * cos_beta_ac;
91. sin2_beta_b2 = sin2_beta * ib2; ia2_sin2_beta = 1 / (a2 * sin2_beta);
92. ic2_sin2_beta = 1 / (c2 * sin2_beta);
93. cos_beta_ac_2_sin2_beta = cos_beta_ac_2 / sin2_beta;
94. case 4
95. sstype = 'ромбическая система';
96. a = input('Введите(в ангстремах)значение параметра a = ');
97. b = input('Введите(в ангстремах)значение параметра b = ');
98. c = input('Введите(в ангстремах)значение параметра c = ');
99. a2 = a * a; b2 = b * b; c2 = c * c; ia2 = 1 / a2; ib2 = 1 / b2; ic2 = 1 / c2;
100. case 5
101. sstype = 'тетрагональная система';
102. a = input('Введите(в ангстремах)значение параметра a = ');
103. c = input('Введите(в ангстремах)значение параметра c = ');
104. a2 = a * a; c2 = c * c; ia2 = 1 / a2; ic2 = 1 / c2; end; end;
105. phi_init = input('Введите значение угла между плоскостями (в гра-
    дусах): ');
106. dphi = input(['Введите максимальное отклонение значения угла',...
    'между плоскостями (в градусах): ']);
107. dphi = abs(dphi);
108. imin = input('Введите минимальное возможное значение индексов
    Миллера: ');
109. imax = input('Введите максимальное возможное значение индек-
    сов Миллера: ');
110. d_min_max = input(['Задать ограничение для межплоскостных',...
    'расстояний d : <1> - Да, <0> - Нет ? ']);

```

```

111.if d_min_max ~= 0 && d_min_max ~= 1
112.ok = false;
113.while ok == false
114.d_min_max = input(['Задать ограничение для межплоскостных',...
    'расстояний d : <1> - Да, <0> - Нет ? ']);
115.if d_min_max == 0 || d_min_max == 1
116.ok = true; end; end; end;
117.if (d_min_max == 1)
118.dmin = input('Введите минимальное возможное значение для d : ');
119.dmax = input('Введите максимальное возможное значение для d :
    ');
120.end;
121.fout = input('Записать результаты расчета в файл: <1> - Да, <0> -
    Нет ? ');
122.if fout ~= 0 && fout ~= 1
123.ok = false;
124.while ok == false
125.fout = input('Записать результаты расчета в файл: <1> - Да, <0> -
    Нет ? ');
126.if fout == 0 || fout == 1
127.ok = true; end; end; end;
128.h1_min = imin; h1_max = imax; k1_min = imin; k1_max = imax;
    l1_min = imin;
129.l1_max = imax;
130.h2_min = imin; h2_max = imax; k2_min = imin; k2_max = imax;
    l2_min = imin;
131.l2_max = imax;
132.phi_min = phi_init - dphi;
133.phi_max = phi_init + dphi;
134.RAD2GRAD = 180 / pi;
135.%erase data
136.data = zeros(1,1);
137.n = 0;
138.for h1=h1_min:h1_max
139.for k1=k1_min:k1_max
140.for l1=l1_min:l1_max
141.for h2=h2_min:h2_max
142.for k2=k2_min:k2_max
143.for l2=l2_min:l2_max

```

```

144.switch stype
145.case 1 % гексагональная система
146.numerator = h1 * h2 + k1 * k2 + 0.5 * (h1 * k2 + h2 * k1) + l1 * l2 *
    a2c075;
147.denominator = sqrt((h1 * h1 + k1 * k1 + h1 * k1 + l1 * l1 * a2c075)
    * ...
148.(h2 * h2 + k2 * k2 + h2 * k2 + l2 * l2 * a2c075));
149.case 2 % кубическая система
150.numerator = h1 * h2 + k1 * k2 + l1 * l2;
151.denominator = sqrt((h1 * h1 + k1 * k1 + l1 * l1) * (h2 * h2 + k2 * k2
    + l2 * l2));
152.case 3 % моноклинная система
153.numerator = h1 * h2 * ia2 + k1 * k2 * sin2_beta_b2 + l1 * l2 * ic2 - ...
    (l1 * h2 + l2 * h1) * cos_beta_ac;
154.denominator = sqrt((h1 * h1 * ia2 + k1 * k1 * sin2_beta_b2 + l1 * l1 *
    ic2 - ...
    h1 * l1 * cos_beta_ac_2) * (h2 * h2 * ia2 + k2 * k2 * sin2_beta_b2
    + ...
    l2 * l2 * ic2 - h2 * l2 * cos_beta_ac_2));
155.case 4 % ромбическая система
156.numerator = h1 * h2 * ia2 + k1 * k2 * ib2 + l1 * l2 * ic2;
157.denominator = sqrt((h1 * h1 * ia2 + k1 * k1 * ib2 + l1 * l1 * ic2) * ...
    (h2 * h2 * ia2 + k2 * k2 * ib2 + l2 * l2 * ic2));
158.case 5 % тетрагональная система
159.numerator = (h1 * h2 + k1 * k2) * ia2 + l1 * l2 * ic2;
160.denominator = sqrt((((h1 * h1 + k1 * k1) * ia2 + l1 * l1 * ic2) * ...
    ((h2 * h2 + k2 * k2) * ia2 + l2 * l2 * ic2)));
161.end;
162.if denominator == 0
163.phi = 0;
164.elseif numerator == 0 & denominator ~= 0
165.phi = sign(denominator) * (pi / 2);
166.else
167.phi = acos( numerator / denominator );
168.end;
169.phi = phi * RAD2GRAD;
170.if ((phi >= phi_min) & (phi <= phi_max))
171.n = n + 1;
172.data(n,1) = phi;

```

```

173.data(n,2) = h1; data(n,3) = k1; data(n,4) = l1;
174.data(n,5) = h2; data(n,6) = k2; data(n,7) = l2;
175.end; end; end; end; end; end; end;
176.%-----remove duplicates
177.index_remove = zeros(1,n);
178.for i=1:n
179.if (index_remove(i) == 0)
180.for j=i+1:n
181.if (index_remove(j) == 0)
182.[first, second] = extra_line(data(i,2),data(i,3),data(i,4),data(i,5),data(i,6),...
    data(i,7), data(j,2),data(j,3),data(j,4),data(j,5),data(j,6),data(j,7));
183.if second == true
184.index_remove(j) = 1;
185.end;
186.if first == true
187.index_remove(i) = 1;
188.break; end; end; end; end; end; end;
189.% erase data_out
190.data_out = zeros(1,1);
191.n_out = 0;
192.for i=1:n
193.if (index_remove(i) == 0)
194.n_out = n_out + 1;
195.data_out(n_out,1) = data(i,1); data_out(n_out,2) = data(i,2);
196.data_out(n_out,3) = data(i,3); data_out(n_out,4) = data(i,4);
197.data_out(n_out,5) = data(i,5); data_out(n_out,6) = data(i,6);
198.data_out(n_out,7) = data(i,7); end; end;
199.for i=1:n_out
200.sum = 0;
201.for j=5:7
202.sum = sum + data_out(i,j) * data_out(i,j);
203.end;
204.data_out(i,9) = data_out(i,1);
205.data_out(i,1) = sum;
206.end;
207.[temp, ix] = sort(data_out, 1);
208.temp = data_out;
209.for i = 1:n_out

```

```

210.data_out(i,:) = temp(ix(i,:));
211.tmp = data_out(i,1);
212.data_out(i,1) = data_out(i,9);
213.data_out(i,9) = tmp;
214.end
215.for i=1:n_out
216.sum = 0;
217.for j=2:4
218.sum = sum + data_out(i,j) * data_out(i,j);
219.end;
220.data_out(i,8) = data_out(i,1);
221.data_out(i,1) = sum;
222.end;
223.[temp, ix] = sort(data_out, 1);
224.temp = data_out;
225.for i = 1:n_out
226.data_out(i,:) = temp(ix(i,:));
227.tmp = data_out(i,1);
228.data_out(i,1) = data_out(i,8);
229.data_out(i,8) = tmp;
230.end;
231.for i = 1:n_out
232.switch stype
233.case 1 % гексагональная система
234.h = data_out(i,2); k = data_out(i,3); l = data_out(i,4);
235.d1 = 1 / sqrt((h * h + h * k + k * k) * a234 + l * l * ic2);
236.h = data_out(i,5); k = data_out(i,6); l = data_out(i,7);
237.d2 = 1 / sqrt((h * h + h * k + k * k) * a234 + l * l * ic2);
238.case 2 % кубическая система
239.h = data_out(i,2); k = data_out(i,3); l = data_out(i,4);
240.d1 = 1 / sqrt((h * h + k * k + l * l) * ia2);
241.h = data_out(i,5); k = data_out(i,6); l = data_out(i,7);
242.d2 = 1 / sqrt((h * h + k * k + l * l) * ia2);
243.case 3 % моноклинная система
244.h = data_out(i,2); k = data_out(i,3); l = data_out(i,4);
245.d1 = 1 / sqrt((h * h * ia2_sin2_beta + k * k * ib2 + l * l *
    ic2_sin2_beta - ...
    h * l * cos_beta_ac_2_sin2_beta));
246.h = data_out(i,5); k = data_out(i,6); l = data_out(i,7);

```

```

247.d2 = 1 / sqrt((h * h * ia2_sin2_beta + k * k * ib2 + l * l *
    ic2_sin2_beta - ...
    h * l * cos_beta_ac_2_sin2_beta));
248.case 4 % ромбическая система
249.h = data_out(i,2); k = data_out(i,3); l = data_out(i,4);
250.d1 = 1 / sqrt(h * h * ia2 + k * k * ib2 + l * l * ic2);
251.h = data_out(i,5); k = data_out(i,6); l = data_out(i,7);
252.d2 = 1 / sqrt(h * h * ia2 + k * k * ib2 + l * l * ic2);
253.case 5 % тетрагональная система
254.h = data_out(i,2); k = data_out(i,3); l = data_out(i,4);
255.d1 = 1 / sqrt((h * h + k * k) * ia2 + l * l * ic2);
256.h = data_out(i,5); k = data_out(i,6); l = data_out(i,7);
257.d2 = 1 / sqrt((h * h + k * k) * ia2 + l * l * ic2);
258.end;
259.data_out(i,10) = data_out(i,9); data_out(i,9) = data_out(i,8);
260.data_out(i,8) = data_out(i,7); data_out(i,7) = data_out(i,6);
261.data_out(i,6) = data_out(i,5); data_out(i,5) = d1;
262.data_out(i,11) = data_out(i,10); data_out(i,10) = data_out(i,9);
263.data_out(i,9) = d2;
264.end; kk = 1; m = 0;
265.if d_min_max == 1
266.temp = data_out;
267.for i = 1:n_out
268.if ((temp(i,5) >= dmin) & (temp(i,5) <= dmax)) &...
    ((temp(i,9) >= dmin) & (temp(i,9) <= dmax))
269.data_out(kk,:) = temp(i,:);
270.kk = kk + 1;
271.end; end;
272.m = size(data_out);
273.n_out = m(2);
274.end;
275.%-----output data
276.%-----console
277.if n_out > 0
278.s = '-----
    ';
279.s1 = sprintf('%55s', ['Индексы Миллера, ', sstype, ' система']);
280.s2 = sprintf('%8s \t %4s %4s %4s %8s\t %4s %4s %4s %8s\t %4s
    %4s', 'угол', ...

```



```

    '(h1', 'k1', 'l1)', 'd1', '(h2', 'k2', 'l2)', 'd2', 'СКИ1', 'СКИ2');
281.result = strvcats(s1, s, s2, s);
282.i = 1;
283.while i <= n_out
284.result = strvcats(result, sprintf(...
285.'%8.2f \t %4d %4d %4d %8.3f \t %4d %4d %4d %8.3f \t %4d %4d',...
    data_out(i,:)));
286.q = data_out(i,10);
287.j = i + 1;
288.while j <= n_out && q == data_out(j,10)
289.result = strvcats(result, sprintf(...
    '%8.2f \t %4d %4d %4d %8.3f \t %4d %4d %4d %8.3f \t %4d %4d',...
    data_out(j,:)));
290.j = j + 1;
291.end;
292.result = strvcats(result, s);
293.i = j;
294.end;
295.result
296.else
297.s = '-----
    ';
298.s1 = sprintf('%55s', ['Индексы Миллера, ', sstype, ' система']);
299.s2 = sprintf('%8s \t %4s %4s %4s %8s \t %4s %4s %4s %8s \t %4s
    %4s',...
    'угол', '(h1', 'k1', 'l1)', 'd1', '(h2', 'k2', 'l2)', 'd2', 'СКИ1', 'СКИ2');
300.end;
301.%-----file
302.[fid, message] = fopen('fimiller.ini','w');
303.fprintf(fid,'%1.0f\n', stype);
304.switch stype
305.case 1
306.fprintf(fid,'%8.4f\n', a); fprintf(fid,'%8.4f\n', c);
307.case 2
308.fprintf(fid,'%8.4f\n', a);
309.case 3;
310.fprintf(fid,'%8.4f\n', a); fprintf(fid,'%8.4f\n', b);
311.fprintf(fid,'%8.4f\n', c); fprintf(fid,'%8.4f\n', beta * (180 / pi));
312.case 4

```

```

313.fprintf(fid,'%8.4f\n', a); fprintf(fid,'%8.4f\n', b);
314.fprintf(fid,'%8.4f\n', c);
315.case 5
316.fprintf(fid,'%8.4f\n', a); fprintf(fid,'%8.4f\n', c);
317.end;
318 fclose(fid);
319.if ((n_out > 0) & (fout == 1))
320.[fid, message] = fopen('fimiller_out.txt','w');
321.fprintf(fid,'%8s \t %4s %4s %4s %8s\t %4s %4s %4s %8s\t %4s
    %4s\n',...
    'угол', 'h1', 'k1', 'l1', 'd1', 'h2', 'k2', 'l2', 'd2', 'СКИ1', 'СКИ2');
322.fprintf(fid,'%8.2f \t %4d %4d %4d %8.3f\t %4d %4d %4d %8.3f\t
    %4d %4d\n',...
    data_out');
323 fclose(fid);
324.end;
325.%-----function
326.function [first, second] = extra_line(h1, k1, l1, h2, k2, l2, h3, k3, l3,
    h4, k4, l4)
327.first = false;
328.second = false;
329.if h1 == 0 & k1 == 0 & l1 == 0
330.first = true;
331.elseif h2 == 0 & k2 == 0 & l2 == 0
332.first = true;
333.end;
334.if h3 == 0 & k3 == 0 & l3 == 0
335.second = true;
336.elseif h4 == 0 & k4 == 0 & l4 == 0
337.second = true;
338.end;
339.if first | second
340.return
341.end;
342.mh13 = divide(h1, h3); mk13 = divide(k1, k3);
343.ml13 = divide(l1, l3); mh24 = divide(h2, h4);
344.mk24 = divide(k2, k4); ml24 = divide(l2, l4);
345.if h1 == 0 & h3 == 0 & mk13 == ml13
346.mh13 = mk13;

```

```

347.elseif k1 == 0 & k3 == 0 & mh13 == ml13
348.mk13 = mh13;
349.elseif l1 == 0 & l3 == 0 & mh13 == mk13
350.ml13 = mk13;
351.elseif h1 == 0 & h3 == 0 & k1 == 0 & k3 == 0
352.mh13 = ml13; mk13 = ml13;
353.elseif h1 == 0 & h3 == 0 & l1 == 0 & l3 == 0
354.mh13 = mk13; ml13 = mk13;
355.elseif k1 == 0 & k3 == 0 & l1 == 0 & l3 == 0
356.mk13 = mh13; ml13 = mh13;
357.end;
358.if h2 == 0 & h4 == 0 & mk24 == ml24
359.mh24 = mk24;
360.elseif k2 == 0 & k4 == 0 & mh24 == ml24
361.mk24 = mh24;
362.elseif l2 == 0 & l4 == 0 & mh24 == mk24
363.ml24 = mk24;
364.elseif h2 == 0 & h4 == 0 & k2 == 0 & k4 == 0
365.mh24 = ml24; mk24 = ml24;
366.elseif h2 == 0 & h4 == 0 & l2 == 0 & l4 == 0
367.mh24 = mk24; ml24 = mk24;
368.elseif k2 == 0 & k4 == 0 & l2 == 0 & l4 == 0
369.mk24 = mh24; ml24 = mh24;
370.end;
371.if (mh13 == mk13 & mk13 == ml13) & (mh24 == mk24 & mk24 ==
    ml24) &...
    (sign(mh13) == sign(mh24))
372.if (mh13 * mh24) > 1
373.first = true;
374.else
375.second = true;
376.end;
377.return;
378.end;
379.mh14 = divide(h1, h4); mk14 = divide(k1, k4);
380.ml14 = divide(l1, l4); mh23 = divide(h2, h3);
381.mk23 = divide(k2, k3); ml23 = divide(l2, l3);
382.if h1 == 0 & h4 == 0 & mk14 == ml14
383.mh14 = mk14;

```

```

384.elseif k1 == 0 & k4 == 0 & mh14 == ml14
385.mk14 = mh14;
386.elseif l1 == 0 & l4 == 0 & mh14 == mk14
387.ml14 = mk14;
388.elseif h1 == 0 & h4 == 0 && k1 == 0 & k4 == 0
389.mh14 = ml14; mk14 = ml14;
390.elseif h1 == 0 & h4 == 0 & l1 == 0 & l4 == 0
391.mh14 = mk14; ml14 = mk14;
392.elseif k1 == 0 & k4 == 0 & l1 == 0 & l4 == 0
393.mk14 = mh14; ml14 = mh14;
394.end;
395.if h2 == 0 & h3 == 0 & mk23 == ml23
396.mh23 = mk23;
397.elseif k2 == 0 & k3 == 0 & mh23 == ml23
398.mk23 = mh23;
399.elseif l2 == 0 & l3 == 0 & mh23 == mk23
400.ml23 = mk23;
401.elseif h2 == 0 & h3 == 0 & k2 == 0 & k3 == 0
402.mh23 = ml23; mk23 = ml23;
403.elseif h2 == 0 & h3 == 0 & l2 == 0 & l3 == 0
404.mh23 = mk23; ml23 = mk23;
405.elseif k2 == 0 & k3 == 0 & l2 == 0 & l3 == 0
406.mk23 = mh23; ml23 = mh23;
407.end;
408.if (mh23 == mk23 & mk23 == ml23) & (mh14 == mk14 & mk14 ==
    ml14) &...
    (sign(mh23) == sign(mh14))
409.if (mh23 * mh14) > 1
410.rst = true;
411.se
412.second = true; end; return; end;
413.function z = divide(x, y)
414.if y == 0
415.z = sign(x) * Inf;
416.else
417.z = x / y;
418.end;

```