

Федеральное государственное бюджетное учреждение науки
Физико-технический институт им. А.Ф. Иоффе
Российской академии наук

На правах рукописи

МАСЛОВ Виктор Николаевич

**МОРФОЛОГИЯ БОКОВОЙ ПОВЕРХНОСТИ ПРОФИЛИРОВАННЫХ
МОНОКРИСТАЛЛОВ ЛЕЙКОСАПФИРА, ВЫРАЩЕННЫХ СПОСОБОМ СТЕПАНОВА**

01.04.07 – физика конденсированного состояния

Диссертация на соискание ученой степени
кандидат физико-математических наук

Научный руководитель
кандидат физико-математических наук
С.И. Бахолдин

Санкт-Петербург 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1 КРИСТАЛЛОГРАФИЯ КОРУНДА (ОБЗОР)	10
2 ВЫРАЩИВАНИЕ КРИСТАЛЛОВ СПОСОБОМ СТЕПАНОВА. ПОЛУЧЕНИЕ ФАКТИЧЕСКОГО МАТЕРИАЛА	16
2.1 Способ Степанова	16
2.2 Выращивание монокристаллических стержней сапфира	19
3 КРАТКИЙ ОБЗОР МЕТОДОВ РАСЧЕТА ОГРАНЕНИЯ КРИСТАЛЛОВ	22
3.1 «Пальцевые» методы	24
3.1.1 Метод Ретикулярных плотностей.....	24
3.1.2 Метод Доннея-Харкера.....	24
3.1.3 Метод периодических цепей связей.....	24
3.2 Молекулярные методы моделирования	27
3.2.1 Метод Ненасыщенности связей.....	27
3.2.2 Результаты расчетов программой METADISE.....	30
3.3 Методы квантового моделирования.....	30
4 ИЗУЧЕНИЕ МОРФОЛОГИИ БОКОВОЙ ПОВЕРХНОСТИ МОНОКРИСТАЛЛИЧЕСКИХ СТЕРЖНЕЙ САПФИРА.....	32
4.1 Метод фотогониографии	32
4.1.1 Исследование ограничения монокристаллических стержней сапфира методом фотогониографии	35
4.1.2 Недостатки метода фотогониографии.....	39
4.2 Конструкция видеогониографа [51]	39
4.3 Видеограммы боковой поверхности монокристаллических стержней сапфира различной ориентации [51].....	45
4.3.1 Ориентация a [1120].....	45
4.3.2 Ориентация c [0001].....	49
4.3.3 Ориентация m [1010]	52

4.4	Модель отражения света от боковой поверхности монокристаллических стержней сапфира	55
4.4.1	Модель отражения от грани с переходными областями	55
4.4.2	Моделирование отражения от грани без переходных областей	58
4.5	Элементы морфологии боковой поверхности кристаллов сапфира	60
4.6	Результаты и выводы	63
5	РАСЧЕТ ПОВЕРХНОСТНЫХ ЭНЕРГИЙ КРИСТАЛЛОВ САПФИРА	65
5.1	Расчет структуры сапфира известными методами [9,10]	65
5.1.1	Расчет ретикулярной плотности	65
5.1.2	Расчет методом Ненасыщенности связей	65
5.2	Результаты расчетов по существующим методам и экспериментальные данные	70
5.3	Метод разрыва связей	72
5.3.1	Принципы метода [53]	72
5.3.2	Модель и анализ решетки сапфира	74
5.3.3	Расчет поверхностных энергий наиболее важных плоскостей кристаллов сапфира	77
5.3.4	Расчет сечений указательной поверхности свободных поверхностных энергий кристаллов сапфира	78
5.4	Результаты и выводы	99
	ЗАКЛЮЧЕНИЕ	101
	СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	104
	СЛОВАРЬ ТЕРМИНОВ	105
	СПИСОК ЛИТЕРАТУРЫ	106
	ПРИЛОЖЕНИЕ 1.1	110
	ПРИЛОЖЕНИЕ 1.2	116
	ПРИЛОЖЕНИЕ 2	121

ВВЕДЕНИЕ

Актуальность и степень разработанности темы.

Сапфир (лейкосапфир) – прозрачная бесцветная разновидность синтетического корунда – вещества обладающего уникальными физическими и химическими свойствами. За десятилетия использования сапфира сформировались две основные области его применения: в качестве конструкционного материала с уникальными механическими характеристиками и в качестве оптического материала с не менее уникальными возможностями. Сапфиры часто используют в ювелирной отрасли, различными методами изменяя его окраску. В технике сапфир занимает отдельную нишу: часовые камни, часовые «стекла», приборные камни (опоры различных механизмов), резцы, подложки для микросхем, химическая посуда, капилляры для исследования микровзрывов, волокна (композиционного материала), термодарные чехлы. В оптике: окна, окна с клином, линзы и призмы, световоды, фокусирующие конусы, мениски, оболочки ламп, рентгеновские интерферометры, сцинтилляторы, люминесцентные экраны и др. В медицине – сапфировые имплантаты.

Одной из проблем, с которыми сталкивается промышленность – это обработка монокристаллов сапфира вследствие их высокой твёрдости. Эту проблему решает метод выращивания кристаллических изделий, предложенный в 1938 г. [1] Александром Васильевичем Степановым. Он позволяет получать монокристаллы заданной формы непосредственно из расплава. Этим способом получены монокристаллы широкого круга полупроводниковых, щелочногидридных, и оксидных материалов.

Управление формой в способе Степанова достигается за счет использования специального элемента – формообразователя, который формирует жидкий столб расплава с сечением, соответствующим сечению выращиваемого кристалла. Однако для монокристаллов характерна анизотропия свободной поверхностной энергии, которая приводит к развитию на их поверхности естественных граней и других морфологических особенностей. Они также влияют на результирующую форму кристаллического изделия. В ряде случаев такие особенности приводят к отклонению формы кристалла от формы, заданной формообразователем. Так, наличие плоских граней на круглом стержне – это отклонение от цилиндричности.

В других случаях наоборот, гранная форма является желательной. Примером может являться выращивание базиснограненных лент сапфира, поверхность, которых не просто гладкая, а образована атомно-гладкой базисной гранью $\{0001\}$ данного кристалла. Такие кристаллические изделия могут быть использованы без какой-либо дополнительной обработки как оптически прозрачные изделия или в качестве подложек в микроэлектронике. Таким

образом для кристаллов, выращенных способом Степанова проявление гранных форм является очень важным свойством.

Хотя в ряде работ [2,3,4,5,6] отмечалось проявление гранных форм на профилированных кристаллах, но систематического изучения количественного влияния этого явления на окончательную форму растущего кристалла не проводилось. В связи с этим, понимание механизма проявления граней на кристаллах, влияющих на форму будущего изделия, в настоящее время является актуальной практической проблемой.

Кроме практических задач – управление формой и качеством поверхности профилированных монокристаллов – изучение морфологии боковой поверхности кристаллов в процессе выращивания имеет большое значение для развития представлений о таких фундаментальных свойствах твердого тела, как свободная поверхностная энергия и определяемая ею равновесная форма кристалла.

Изучение ограничения, как фундаментального свойства твердого тела – анизотропии свободной поверхностной энергии на сегодняшний день является одной из наиболее актуальных тематик в физике твердого тела. Существующие модели позволяют описывать ограниченный набор плоскостей и плохо предсказывают экспериментально получаемую форму кристаллов. Поэтому изучение проявления гранных форм является также и актуальной научной проблемой.

Цели и задачи.

Целью данной работы является выявление закономерностей в проявлении гранных форм и других морфологических особенностей боковой поверхности профилированных кристаллов сапфира, их влияние на окончательную форму кристалла, а также разработка математического аппарата для прогнозирования проявляющихся граней.

Для достижения поставленной цели были сформулированы следующие задачи исследования:

1. Выращивание монокристаллических стержней сапфира способом Степанова трех главных кристаллографических ориентаций направления вытягивания (c , a , m) с максимальным контролем условий роста.
2. Развитие экспериментальной методики описания морфологии боковой поверхности цилиндрических монокристаллов.
3. Разработка прибора для реализации методики описания морфологии боковой поверхности цилиндрических монокристаллов на количественном уровне.
4. Экспериментальное исследование морфологии боковой поверхности цилиндрических монокристаллов сапфира, выращенных способом Степанова.

5. Разработка новой методики расчета поверхностной энергии по сечениям указательной поверхности для прогнозирования граней, проявляющихся на боковой поверхности монокристаллов на примере монокристаллических стержней сапфира, выращенных способом Степанова.

6. Сравнение экспериментальных и расчетных данных для определения достоверности метода расчета и выявления возможного набора гранных форм на боковой поверхности кристаллов на примере монокристаллических стержней сапфира, выращенных способом Степанова.

Научная новизна.

Развита методика исследования морфологии боковой поверхности цилиндрических монокристаллов и создан на ее основе действующий лабораторный макет нового оптического прибора с цифровой регистрацией – видеогониографа, который впервые позволил в полной мере изучить морфологию боковой поверхности цилиндрических монокристаллов сапфира на количественном уровне. В ходе такого изучения были впервые описаны десятки не определявшихся ранее граней на боковой поверхности монокристаллических стержней сапфира. Наряду с гранями метод видеогониографии показал другие неизвестные ранее особенности морфологии поверхности. Установлены также неизвестные до настоящего момента закономерности взаимосвязи между проявляющимися гранями. Метод также позволяет определять разориентацию цилиндрического кристалла, хотя на данном этапе только на качественном уровне. Установлено, что разориентация кристалла сильнее влияет на слабо проявляющиеся простые формы.

Разработан новый метод расчета свободной поверхностной энергии кристаллов по сечениям указательной поверхности для прогнозирования граней, проявляющихся на боковой поверхности цилиндрических монокристаллов. Разработанный метод впервые позволил построить полные сечения указательной поверхности свободной поверхностной энергии для различных ориентаций монокристаллов сапфира. Путём сопоставления рассчитанных значений энергии со средней плотностью энергии тепловых колебаний при температуре плавления выявлены десятки направлений, которые будут проявляться в виде гранных форм и других морфологических особенностей на боковой поверхности. Показано также, что при температуре плавления сечения указательной поверхности не только содержат сингулярные минимумы, которые могут вызывать появления граней, но и в целом отличаются от окружностей, полностью соответствуя симметрии кристаллической решётки сапфира. Это может приводить при выращивании кристаллов к отклонению от цилиндрической формы, задаваемой формообразователем, как на огранённых, так и на неогранённых участках боковой поверхности.

Выполнено сопоставление экспериментальных и расчетных данных, которое показало, что эти данные хорошо коррелируют друг с другом. Показано, что для различных ориентаций направления выращивания кристаллов сапфира, предсказанный набор граней на 75-100% совпадает с наблюдаемым набором морфологических особенностей на боковой поверхности кристаллов. Экспериментальными данными подтверждена достоверность нового метода расчета свободной поверхностной энергии.

Теоретическая и практическая значимость

Полученные экспериментальные и расчётные результаты могут служить основой для дальнейшего изучения влияния ограничения на форму профилированных кристаллов. Выполненные исследования являются необходимым этапом на пути решения задачи расчёта реальной формы кристалла и её отклонения от формы, задаваемой формообразователем вследствие анизотропии свободной поверхностной энергии. Это, в свою очередь, необходимо для развития методов управления формой и оптимизации процесса выращивания.

Разработанный и созданный новый оптический прибор – видеогониограф, являющийся уникальным инструментом изучения морфологии кристаллов, который может также стать устройством для неразрушающего экспрессного анализа разориентации кристаллов, что сейчас не позволяет делать ни один существующий метод.

Данные полученные с помощью видеогониографа и разработанного метода расчета вносят весомый вклад в понимание таких фундаментальных свойств твердых тел, как свободная поверхностная энергия и ее анизотропия, а также влияние этих свойств на реальную форму кристалла.

Методология и методы исследования.

Решение задач базируется на экспериментальных данных и известных теоретических положениях физики твердого тела и математического моделирования. Достоверность полученных результатов подтверждается корректностью разработанных математических моделей, их адекватностью по известным критериям оценки изучаемых процессов, использованием известных положений фундаментальной науки, сходимостью полученных теоретических результатов с данными экспериментов и результатами эксплуатации созданного прибора.

В работе использовалась отработанная технология выращивания профилированных кристаллов сапфира способом Степанова.

На основе известного ранее метода фотогониографии разработан новый метод изучения морфологии боковой поверхности цилиндрических кристаллов – видеогониография и прибор для его реализации.

Полученные данные полностью подтвердили полученные ранее известным методом фотогониографии качественные результаты и позволили дать количественные характеристики морфологических особенностей боковой поверхности цилиндрических монокристаллов сапфира.

Достоверность и надежность результатов

Достоверность и надежность результатов, представленных в диссертации, определяется:

- (1) В экспериментальной части работы: хорошим соответствием результатов, полученных методом видеогониографии ранее известным особенностям морфологии сапфира и фундаментальным законам кристаллографии.
- (2) Использованием известных методов расчётов поверхностной энергии для развития нового подхода.
- (3) В расчетной части: хорошим соответствием результатов расчетов морфологических особенностей боковой поверхности кристаллов сапфира и полученных экспериментальных данных.
- (4) Публикацией результатов исследований в престижных научных журналах и докладах на различных российских и международных конференциях.

Апробация работы и личный вклад автора

Материалы диссертационной работы докладывались автором на следующих российских и международных конференциях: «Конференция стран СНГ по росту кристаллов» (Харьков, Украина, 2012), «Физико-химия и технология неорганических материалов» (Москва, 2013), Международный симпозиум «Физика кристаллов 2013» (Москва, 2013), «Кристаллофизика и деформационное поведение перспективных материалов» (Москва, 2015), а также на научных семинарах в ФТИ им. А.Ф. Иоффе (Санкт-Петербург).

Перечень публикаций, раскрывающих основное содержание диссертации содержит 4 наименования в реферируемых журналах (личный вклад автора указан в скобках):

[1] С.И. Бахолдин, **В.Н. Маслов** «*Применение метода периодических цепей связей для расчета простых форм сапфира*». Кристаллография, том 59, № 4, с. 649-653 (2014) (Участие в расчетах, обсуждение результатов и написание статьи).

[2] С.И. Бахолдин, **В.Н. Маслов** «*Моделирование габитусных форм кристаллов сапфира с использованием принципов подхода периодических цепей связей*». Кристаллография, том 60, № 2, с. 336-341 (2015) (Участие в расчетах, обсуждение результатов и написание статьи).

[3] С.И. Бахолдин, **В.Н. Маслов** «*Моделирование поверхностных энергий кристаллов сапфира*». Физика твердого тела, том 57, вып. 6, с. 1213-1219 (2015) (Участие в разработке метода и расчетах, обсуждение результатов и написание статьи).

[4] С.И. Бахолдин, **В.Н. Маслов** «Метод видеогониографии для изучения ограничения кристаллов сапфира, выращенных способом Степанова». Журнал технической физики, том 85, вып. 9, с. 132-135 (2015) (Разработка метода исследования, разработка и сборка видеогониографа, проведение экспериментов, участие в интерпретации полученных результатов, написание статьи).

Основные положения, выносимые на защиту.

1. Разработан и создан новый оптический прибор с цифровой регистрацией и количественный метод для изучения морфологии боковой поверхности цилиндрических монокристаллов – видеогониограф, который был успешно применен для изучения боковой поверхности профилированных стержней сапфира:

А. Выявлен более полный набор гранных форм на боковой поверхности профилированных кристаллов сапфира, чем это позволял сделать предшествующий метод – метод фотогониографии;

Б. Подтверждена главная последовательность граней на боковой поверхности профилированных цилиндрических монокристаллов сапфира $h_c:h_r:h_a$ как 8:4:1;

В. Выявлены и описаны элементы морфологии боковой поверхности профилированных цилиндрических монокристаллов сапфира, влияющие на форму растущего кристалла – зеркальная грань, ребро, уплощение.

2. Разработан новый экспрессный метод расчета свободной поверхностной энергии кристаллов по сечениям указательной поверхности, которым удалось рассчитать полный набор граней на боковой поверхности кристаллов, хорошо совпадающий с экспериментальными данными:

А. В виде зеркальных проявляются грани, имеющие значение поверхностной энергии менее 4 Дж/м²;

Б. Минимальную поверхностную энергию имеют плоскости, упакованные атомами кислорода.

3. Боковая поверхность профилированных монокристаллических стержней сапфира не является строго цилиндрической. Ее морфология определяется анизотропией свободной поверхностной энергии и согласуется с результатами проведенных расчетов.

1 КРИСТАЛЛОГРАФИЯ КОРУНДА (ОБЗОР)

Монокристаллы корунда относятся к дитригонально-скаленоэдрическому классу тригональной сингонии $D_{3d}^6 - R\bar{3}c$ (L_33L_23PC) [7] с элементами симметрии (рисунок 1):

- зеркально-поворотной осью шестого порядка (осью инверсии третьего порядка);
- перпендикулярными к ней тремя осями второго порядка;
- тремя плоскостями симметрии, перпендикулярными к осям второго порядка и пересекающимися по оси высшего порядка;
- центром симметрии.

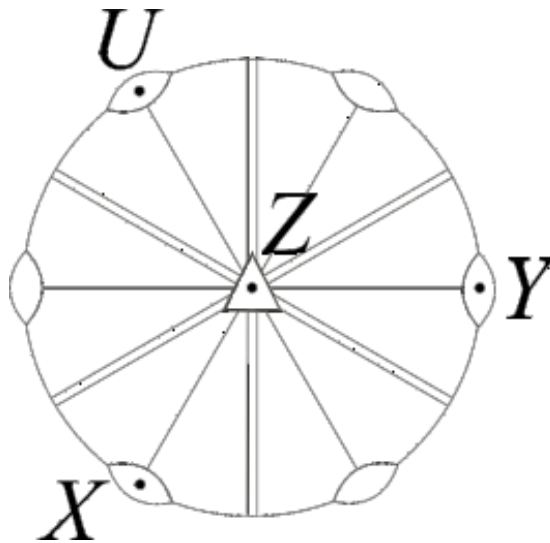
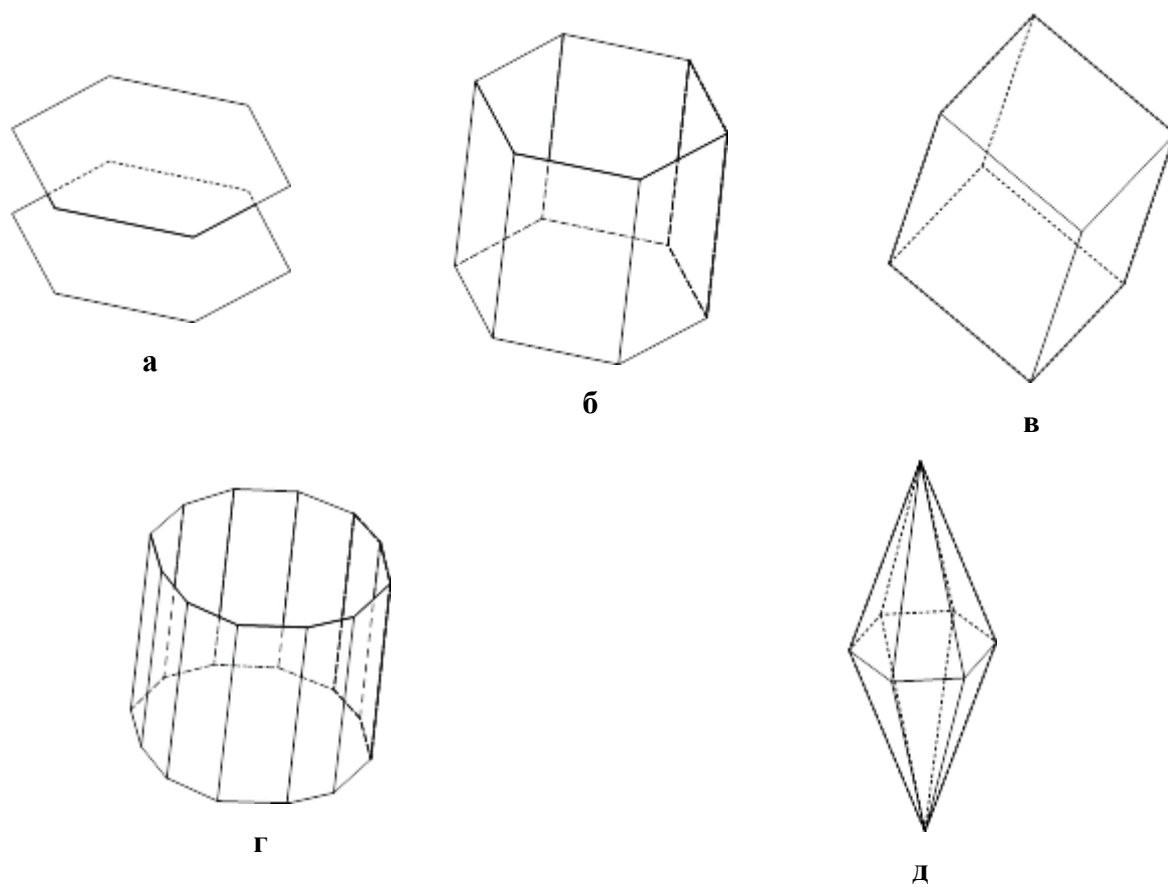


Рис. 1. Пиктограмма тригонального класса симметрии

В этом классе симметрии возможны шесть простых форм:

- пинакоид $\{0001\}$;
- гексагональная призма $\{10\bar{1}0\}$ и $\{11\bar{2}0\}$;
- дигексагональная призма $\{hki0\}$;
- ромбоэдр $\{h0\bar{h}l\}$;
- гексагональная дипирамида $\{hh\bar{2}hl\}$;
- дитригональный скаленоэдр $\{\bar{h}kil\}$.

Скаленоэдр в кристаллах корунда никогда не наблюдался. Изображения простых форм корунда приведены на рисунке 2.



- а – пинакоид $c \{0001\}$;
 б – гексагональная призма $m \{10\bar{1}0\}$ и $a \{11\bar{2}0\}$;
 в – семейство ромбоэдров $r, R, q, S, t \{h0\bar{h}l\}$;
 г – семейство дигексагональных призм $d, f \{hki0\}$;
 д – семейство гексагональных дипирамид $P, n, w, u \{hh2\bar{h}l\}$;

Рис. 2. Простые формы кристаллов корунда

Ромбоэдр по своему положению относительно гексагональных кристаллографических осей может быть «положительным» $\{h0\bar{h}l\}$ или «отрицательным» $\{0h\bar{h}l\}$ [8].

Кристаллическая решетка $\alpha\text{-Al}_2\text{O}_3$ образована ионами алюминия (Al^{3+}) и кислорода (O^{2-}). Если анионы O^{2-} изобразить в виде шаров, то кристаллическую решетку можно в первом приближении представить в виде их гексагональной плотнейшей упаковки (рисунок 3). Катионы Al^{3+} находятся в кристаллическом поле, не имеющего центра симметрии (из-за искажения решетки), и располагаются в октаэдрических пустотах между плотноупакованными ионами O^{2-} , заполняя $2/3$ этих пустот. Октаэдрическая пустота окружена шестью шарами и, если радиус каждого из них принять за единицу, то в пустоте размещается шар с относительным радиусом 0,41. Соотношение ионных радиусов O^{2-} и Al^{3+} (1,40 Å и 0,57 Å) позволяет катионам располагаться в пустотах упаковки анионов, немного исказив решетку, не выходя при этом за пределы устойчивости октаэдрической позиции [7].

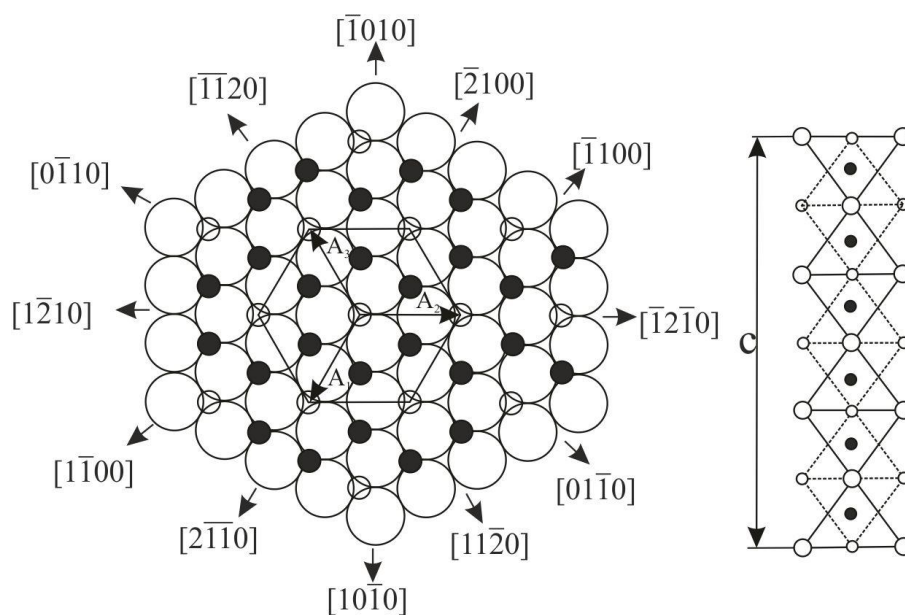


Рис. 3. Идеализированная схема расположения Al^{3+} (черные кружки) и октаэдрических пустот (малые светлые кружки) между двумя слоями O^{2-} (большие светлые кружки) в базисной плоскости (верхний слой O^{2-} не показан). A_1, A_2, A_3 – векторы $\{11\bar{2}0\}$ трансляции гексагональной ячейки сапфира для плоскости базиса. Справа – идеализированная схема упаковки ионов O^{2-} (светлые кружки) и Al^{3+} в направлении оси C_3 [7].

Координационные числа для Al^{3+} – 6, для O^{2-} – 4. Три верхних иона O^{2-} в октаэдре повернуты относительно трех нижних на $64,3^\circ$ и лежат в параллельных плоскостях на расстоянии $2,164 \text{ \AA}$. Искажение плотнейшей упаковки, вызванное некоторым несоответствием размеров Al^{3+} и октаэдрической позиции, приводят к тому, что октаэдры образуют кислородные треугольники разного размера (рисунок 4а), а угол их поворота $64,3^\circ$ превышает значение, характерное для идеальной упаковки (60°) [7].

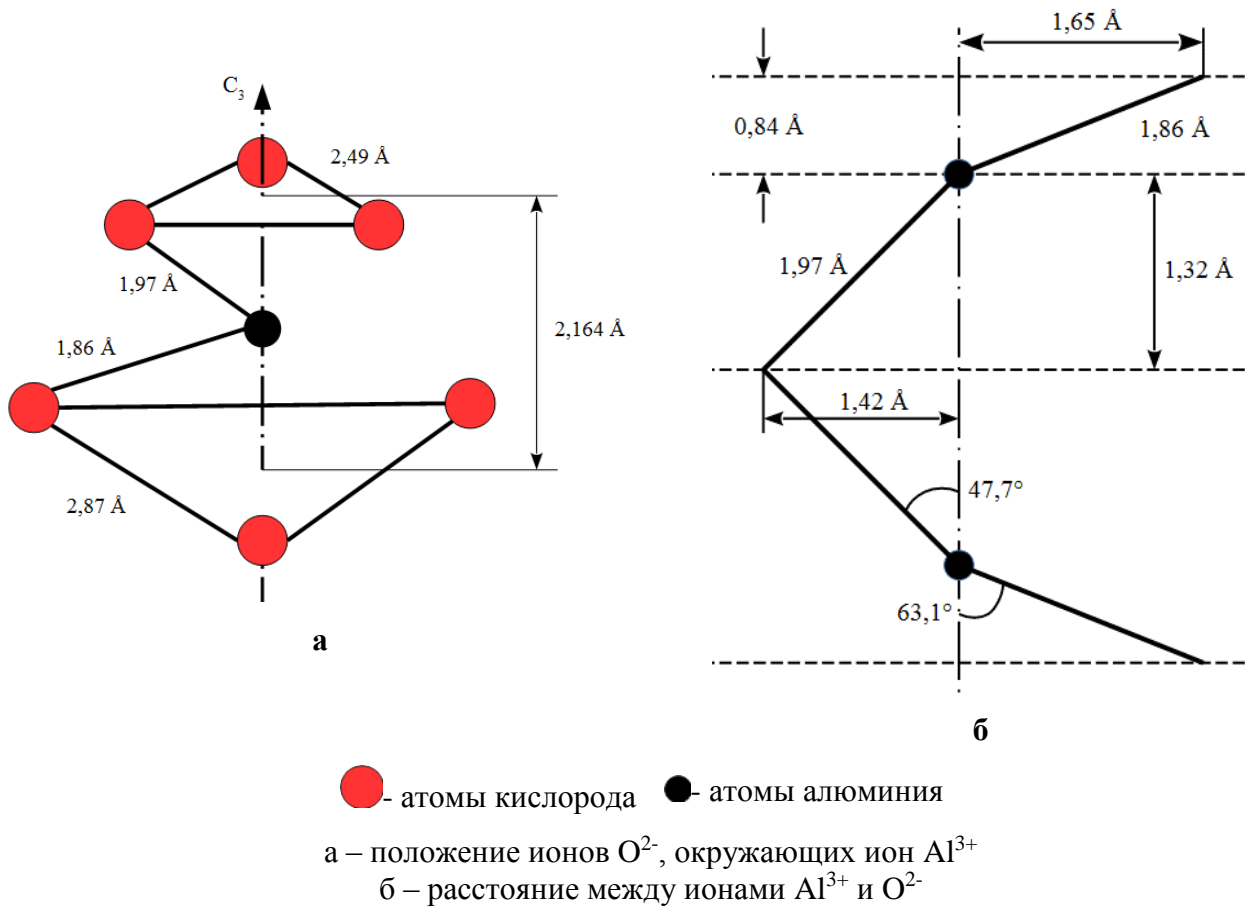


Рис. 4. Схема расположения Al^{3+} и O^{2-} в решетке корунда [7,8]

Пространственное расположение O^{2-} образует так называемый корундовый мотив. Позиции Al^{3+} смещены (рисунок 4б) из-за чередования октаэдрических пустот и повторяются в структуре через каждые три слоя. Расположение структурных единиц вдоль оси третьего порядка повторяется через $12,97 \text{ \AA}$ (период идентичности гексагональной решетки), то есть полностью структура повторяется через шесть слоев O^{2-} и шесть промежуточных слоев Al^{3+} . В направлении оси C три расстояния $Al-O$ равны $1,97 \text{ \AA}$, три другие – $1,86 \text{ \AA}$ (рисунок 4б), расстояние $Al-Al$ – $2,65 \text{ \AA}$, $O-O$ – $2,52-2,87 \text{ \AA}$. Смещение атомов кислорода относительно друг друга и образование неравных кислородных треугольников приводит к образованию в структуре корунда винтовых осей третьего порядка [7].

Для идеальной плотнейшей упаковки из жестких шаров отношение векторов трансляции составляет $1,63$, в то время как у корунда – $1,58$ (искаженная структура имеет более низкую энергию, чем идеальная) [7].

Опираясь на данные рентгеновских исследований нами была построена уточнённая модель структуры корунда [9,10]. Проекция на три основные плоскости структуры корунда

приведены на рисунке 5, где кружками разного цвета показаны усредненные положения атомов Al^{3+} и O^{2-} , а также незаполненные позиции (пустоты).

В дальнейшем будет использоваться структура, приведенная на рисунке 5.

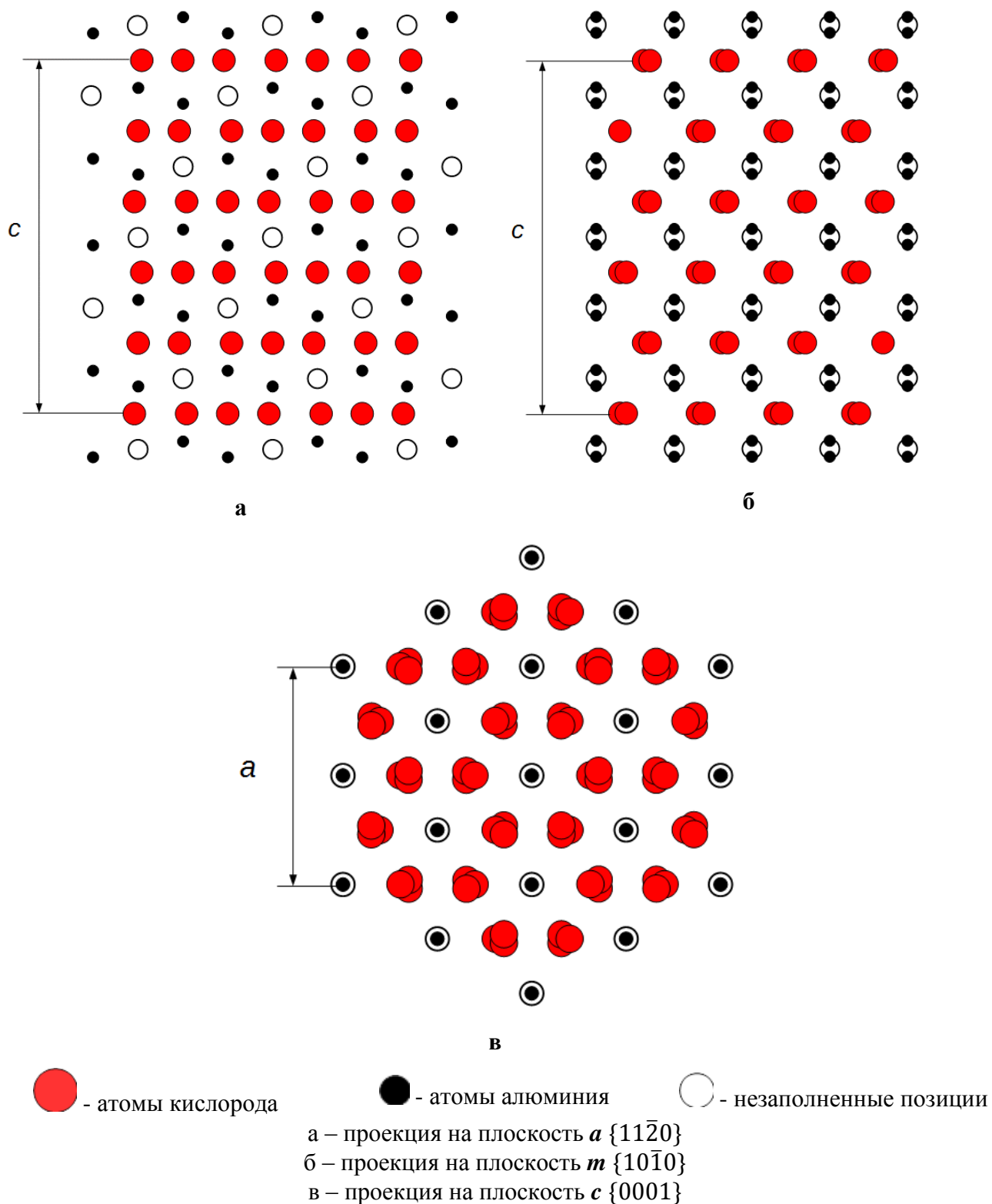


Рис. 5. Проекция на три основные плоскости элементарной ячейки корунда.

В связи с тем, что в корунде исторически выделялись два типа ячеек (структурная и морфологическая), отличающихся по высоте (структурная ячейка выше на 2 слоя кислорода), возникает проблема с указанием символов граней, так как для разных ячеек они будут отличаться (таблица 1). В связи с этим в дальнейшем мы будем пользоваться общепринятыми буквенными обозначениями, а если таких не окажется, будем просто указывать угол с реперной гранью. К реперным граням будем относить:

- пинакоид c $\{0001\}$;
- гексагональную призму a $\{11\bar{2}0\}$;
- гексагональную призму m $\{10\bar{1}0\}$.

Эти простые формы мы будем считать реперными, так как символы их граней одинаковы для морфологической и структурной ячеек.

Таблица 1 – морфологические и структурные символы наиболее распространенных простых форм сапфира

Общепринятые символы граней	Символ грани в морфологической установке	Символ грани в структурной установке
c	$\{0001\}$	$\{0001\}$
a	$\{11\bar{2}0\}$	$\{11\bar{2}0\}$
m	$\{10\bar{1}0\}$	$\{10\bar{1}0\}$
r	$\{10\bar{1}1\}$	$\{10\bar{1}2\}$
R	$\{10\bar{1}2\}$	$\{10\bar{1}4\}$
S	$\{20\bar{2}1\}$	$\{10\bar{1}1\}$
t	$\{30\bar{3}2\}$	$\{20\bar{2}1\}$
u	$\{22\bar{4}1\}$	$\{11\bar{2}1\}$
w	$\{11\bar{2}1\}$	$\{11\bar{2}2\}$
P	$\{11\bar{2}3\}$	$\{11\bar{2}6\}$
n	$\{22\bar{4}3\}$	$\{11\bar{2}3\}$

2 ВЫРАЩИВАНИЕ КРИСТАЛЛОВ СПОСОБОМ СТЕПАНОВА. ПОЛУЧЕНИЕ ФАКТИЧЕСКОГО МАТЕРИАЛА

2.1 Способ Степанова

Различные материалы могут быть использованы для создания конструкций, машин и приборов только в том случае, если им можно придать необходимую форму. Поэтому одна из главных задач технологии заключается в разработке способов придания материалам определенной формы [11].

Формообразование кристаллических материалов осуществляется посредством обработки их в твердом состоянии (прокатка, ковка, прессование, резание, шлифовка и т.п.), разнообразными приемами литья, при которых форма задается стенками сосуда, и соединения отдельных частей различными способами (лепка, склейка, сварка, сборка, клепка, соединение приемами металлокерамики и т.п.) [12].

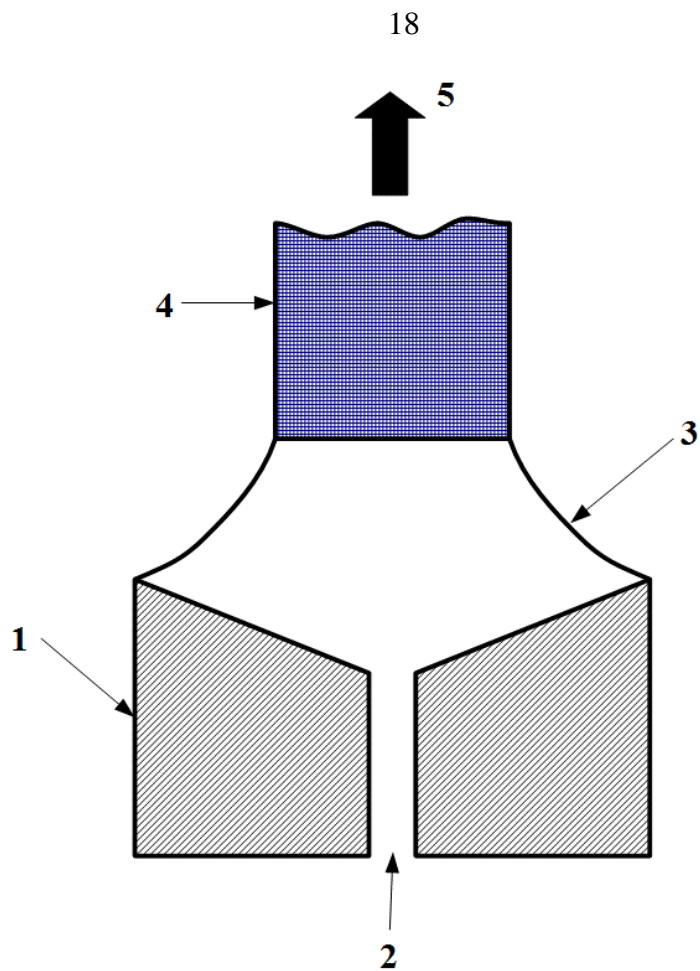
Монокристаллические изделия, как правило, вырезаются из монокристаллических слитков. Однако при резке и механической обработке в монокристалле возникают дефекты структуры, которые могут неконтролируемо изменить его свойства. Кроме того, 70-90% материала переходят в отходы. В ряде случаев эти отходы пропадают. При производстве изделий из остродефицитных, дорогостоящих материалов затрачиваются большие усилия на их регенерацию. Поэтому уже почти полвека ведутся исследования различных методов получения профилированных монокристаллов. Среди них – метод пластической деформации монокристаллических слитков, литьё в формы, эпитаксиальный рост, массовая кристаллизация из растворов в расплавах, кристаллизация из газовой фазы, дендритная кристаллизация. Их преимущества и недостатки рассмотрены в монографии [13].

В 1938 г. член-корреспондент АН СССР А.Н. Степанов предложил новый технологический принцип формообразования твердых тел, при котором создание определенной формы материала осуществляется в жидком состоянии, когда расплав у фронта кристаллизации не контактирует со стенками сосуда.

Форма жидкости, не касающейся стенок сосуда, определяется поверхностным натяжением, удельным весом и характером сцепления жидкости с прилежащими телами. В связи с этим из расплавленного материала нетрудно вытянуть на некоторую высоту столб жидкости желаемой формы через соответствующее отверстие или щель пластины, находящейся на поверхности расплава и называемой формообразователем. Можно создать такие условия, при которых будет наблюдаться непрерывный подъем жидкого материала, вытягиваемого из

расплава, при одновременной кристаллизации его на определенном расстоянии над отверстием формообразователя [11,14]. Дальнейшие исследования этого метода показали, что формообразование столба расплава можно осуществить не только при зацеплении жидкого столба за формообразователь по периметру отверстия или щели. Зацепление можно осуществить при наличии любых кромок на формообразователе. Более того, можно получить столб расплава определенного сечения при сцеплении нижней части его с гладкой поверхностью формообразователя. Это обусловлено постоянством угла смачивания материала формообразователя расплавом кристаллизуемого вещества.

Кроме капиллярных сил для формообразования жидкости могут быть использованы и другие физические силы, например, электромагнитные [15]. В дальнейшем новый принцип формообразования был сформулирован автором способа в обобщенном виде следующим образом: форма или элемент формы, которую желательно получить, создается в жидком виде за счет различных эффектов, позволяющих жидкости сохранить форму, затем сформированный так объем жидкости переводится в твердое состояние в результате подбора соответствующих условий кристаллизации [11]. Схематическое изображение способа приведено на рисунке 6.



- 1 – формообразователь
- 2 – капилляр
- 3 – жидкий мениск
- 4 – растущий кристалл
- 5 – направление вытягивания

Рис. 6. Схематическое изображение принципа выращивания кристалла по способу Степанова

Проблема получения профилированных монокристаллов имеет такое же важное значение, как и проблема изготовления поликристаллических изделий. Это обусловлено тем, что монокристаллы различных материалов в виде пластин, волокон, трубок, стержней разного поперечного сечения широко используются при создании приборов и устройств в таких отраслях науки и техники, как электрика, оптика, акустика, лазерная техника, микроэлектроника и т.п [12].

Форма кристалла при выращивании способом Степанова определяется:

1. Столбом расплава, создаваемым формообразователем и тепловыми условиями;
2. Проявлением гранных форм, как на боковой поверхности кристалла, так и на фронте кристаллизации.

Для способа Степанова проявление гранных форм является очень важным свойством, так как сам метод позволяет получать непосредственно из расплава кристаллы заданной формы, не требующие дополнительной обработки или требующие минимальной обработки.

Для кристаллов, выращенных способом Степанова проявление гранных форм на боковой поверхности может быть, как положительным, так и отрицательным явлением.

Гранные формы играют положительную роль для будущего изделия, когда помогают управлять его формой, как в случае прямоугольного сечения профиля, или для получения атомно-гладкой поверхности на кристалле.

Отрицательную же роль гранная форма играет, когда необходимо получить кристалл круглого или любого другого криволинейного сечения.

2.2 Выращивание монокристаллических стержней сапфира

Для проведения экспериментов по изучению морфологии боковой поверхности монокристаллических стержней сапфира нами были выращены сапфировые стержни трех наиболее важных для сапфира ориентаций – c , a , m . Пиктограммы этих ориентаций приведены на рисунке 7.

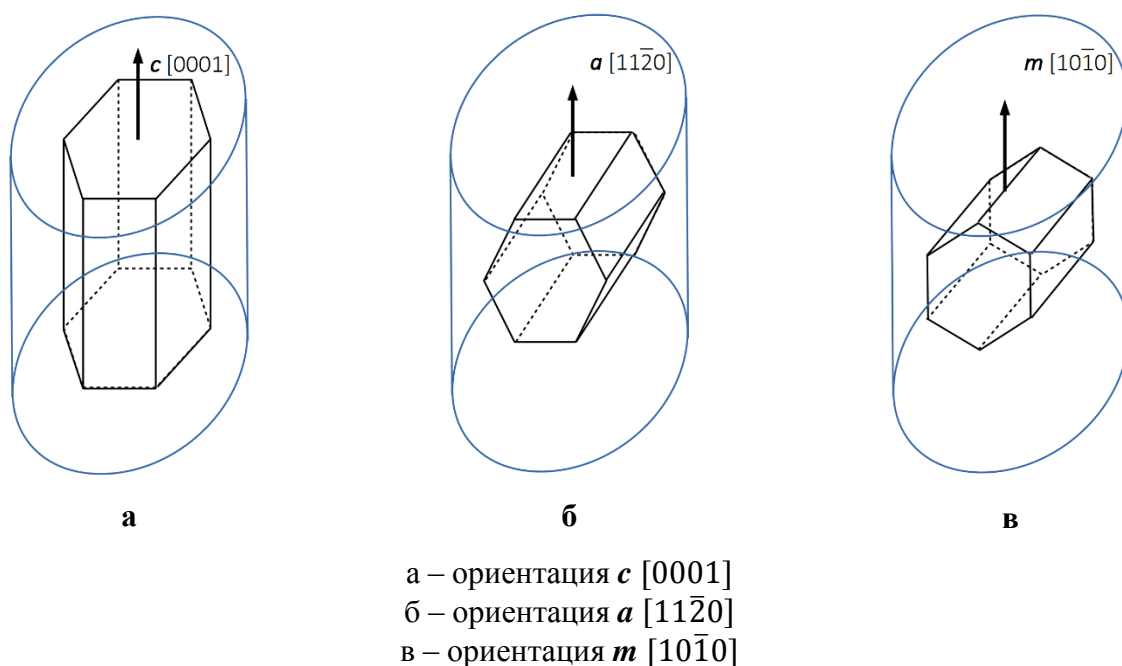


Рис. 7. Пиктограммы ориентаций кристаллов сапфира, выращенных для исследования морфологии боковой поверхности

Такие три направления выращивания были выбраны в связи с тем, что огранение их боковых поверхностей в совокупности наиболее полно отражает набор возможных граней и

позволяет описать кристаллографию монокристаллических стержней сапфира. Помимо этого, направление оптической оси c было выбрано в качестве направления выращивания, поскольку оно является практически важным и используется в большинстве работ по выращиванию кристаллов сапфира из расплава.

Кристаллы всех трех ориентаций выращивались в одной и той же тепловой зоне с максимальным контролем режимов роста. Диаметр кристаллов равняется 8 мм.

Для выставления затравок использовалась точная лазерная юстировка по двум граням (рисунок 8). Процесс лазерной юстировки заключается в следующем. На штатив устанавливается лазер с мишенью (для фиксации отражения) и при помощи пузырькового уровня выставляется в горизонте. Далее лазер юстируется по вытягивающему штоку. На шток устанавливается зеркало, затем лазер выставляется так чтобы отражающийся от зеркала луч приходил обратно в источник. Затем вывешивается затравочный кристалл и ориентируется наиболее широкой гранью к лазеру. Юстировочными винтами на затравкадержателе отраженный луч приводится в источник. После затравка поворачивается другой гранью к лазеру, и процедура повторяется. Это позволяет максимально точно выставить затравку с погрешностью не более 3-5'.

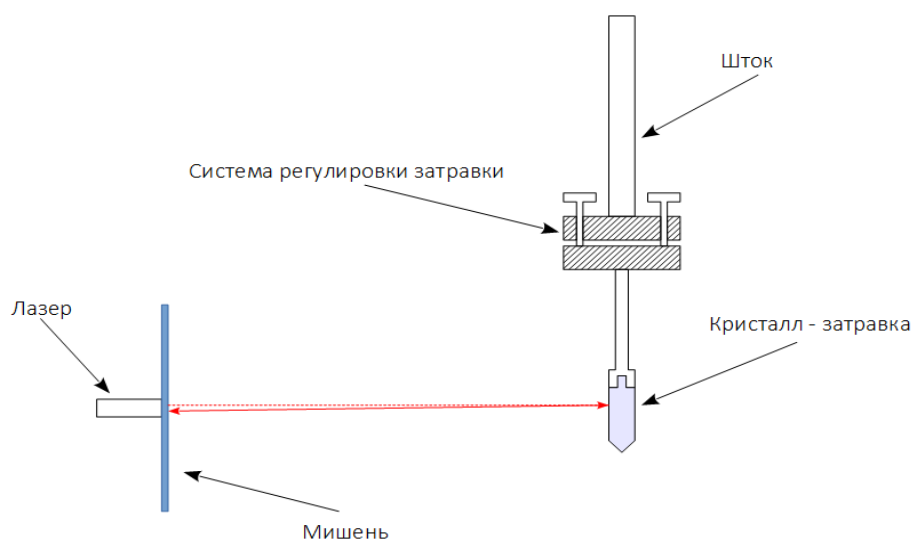


Рис. 8. Схематическое изображение процесса лазерной юстировки затравочного кристалла

Выращенные кристаллы показаны на рисунке 9.



Рис. 9. Кристаллы, выращенные для проведения исследований морфологии боковой поверхности

3 КРАТКИЙ ОБЗОР МЕТОДОВ РАСЧЕТА ОГРАНЕНИЯ КРИСТАЛЛОВ

Данная глава посвящена обзору существующих методов расчета ограничения монокристаллов. Проанализированы работы, в которых даются данные по расчетам для кристаллов корунда.

Известно, что кристалл обладает особенностью самоограняться. Известно также, что гранная форма кристалла – это проявление анизотропии свободной поверхностной энергии, определяемая его структурой и условиями роста. Гранная форма является естественной, как для природных кристаллов, так и для искусственно выращенных.

При решении вопросов морфогенеза кристаллов, при получении кристаллов и пленок (не только из расплавов) анализу тонкого строения граней кристаллов, правильному выбору кристаллографической ориентации поверхностей роста отводится важная роль. Обработка кристаллов также требует учета их анизотропии, знания атомного строения различно ориентированных плоскостей.

За последние десятилетия достигнут существенный прогресс в изучении на атомном уровне реальной поверхности. Обнаружены сверхструктуры на атомно-гладких гранях и явления перестройки ступеней на вицинальных поверхностях, носящие характер фазовых переходов [16,17,18]. Анализ этих явлений, так же, как и решение некоторых актуальных вопросов, относящихся к проблемам наноматериалов [19], невозможен без сведений об идеальном строении граней. В литературе этот вопрос разработан не полно. Например, при рассмотрении кристаллов типа сфалерита обычно используется эмпирическое моделирование с помощью проволочно-шариковых «структур» граней с малыми индексами Миллера [20].

Помимо гранных форм (плоских участков) на поверхности цилиндрических кристаллов сапфира, выращенных методом Степанова (и не только) проявляются округлые (негранные) участки, которые в большинстве случаев занимают большую часть боковой поверхности. [6] Строение и физика образования таких поверхностей до конца не ясна.

Внешняя морфология кристалла зависит от скоростей роста различных граней и изменения их размеров во времени. В свою очередь, скорость роста грани кристалла определяется внешними и внутренними факторами.

Внутренними факторами являются структура кристалла, наличие и распределение дислокаций и присутствие границ двойников. Внешние факторы: температура, давление, степень неравновесности (пересыщение, переохлаждение), приток кристаллизующихся частиц (определяемый их концентрацией и диффузией в некристаллической фазе и степенью

перемешивания в этой фазе) и, наконец, присутствие инородных материалов (включая растворитель) [21].

Перемещение плоскости грани кристалла может происходить двумя путями: в случае идеальных кристаллов – посредством образования двумерных зародышей; в реальных кристаллах – таким же способом или с помощью механизма винтовых дислокаций. Образованию двумерных зародышей способствуют высокие пересыщения, которые уменьшают радиус критического зародыша, и малая плотность дислокаций [21].

Как у идеальных, так и у реальных кристаллов встречаются слои роста, но не все грани кристалла перемещаются таким образом. Для некоторых необходимо только одновременное зародышеобразование, а большинство граней вообще не нуждается в образовании зародышей. Так как последние характеризуются большими скоростями роста, они обычно не встречаются и их роль незначительна. Наиболее выраженные грани кристалла – это грани со слоями роста [22].

Таким образом, ключевыми являются следующие вопросы:

- Каким образом внешняя морфология определяется внутренней структурой кристалла?
- Каков механизм влияния внешних факторов на внутренние структурные характеристики кристалла и морфологию?

Считается, что в случае выращивания кристаллов из собственного расплава методами вытягивания со свободной поверхности расплава, роста на боковой поверхности кристалла нет, т.е. образование боковой поверхности происходит непосредственно на трёхфазной линии, ограничивающей фронт кристаллизации. Таким образом, при рассмотрении морфологии боковой поверхности цилиндрических монокристаллов влияние внешних факторов минимально.

В связи с этим необходимо дать точный ответ на первый вопрос, который можно было бы получить теоретическим путем – расчетом равновесной формы кристалла [22,23,24,25,26,27].

В настоящее время все существующие методы расчета огранения кристаллов через его равновесную форму можно разделить на три больших группы:

1. «Пальцевые» методы – наиболее простые модели, дающие качественную оценку огранению кристаллов. К таким методам относятся: метод Ретикулярной плотности [28], метод Доннея-Харкера [29,30], метод Периодических цепей связей (ПЦС) [23,24,25,26,27].
2. Методы молекулярного моделирования (атомистического подхода) – методы, дающие количественные оценки поверхностных энергий. Наиболее известным в настоящее

время является программный пакет METADISE [31]. Менее распространенным методом является метод Ненасыщенности связей [20,32]

3. Методы квантового моделирования – наиболее ресурсоемкие методы. Они основаны на различных моделях взаимодействия атомов и учитывают распределение электронной плотности в кристалле [33,34,35,36,37,38,39,40,41,42].

3.1 «Пальцевые» методы

3.1.1 Метод Ретикулярных плотностей

Наиболее простой метод оценки огранения кристаллов был предложен ещё в 18 веке Огюстом Браве – оценка ретикулярных плотностей. Он сформулировал свой принцип так: Грани, наиболее часто образующиеся на кристалле в процессе кристаллизации, или, другими словами, плоские сетки наиболее пригодные для огранения кристалла, являются, в общем, теми, ретикулярная плотность которых наиболее значительна [28]. То есть кристалл по Браве будет ограняться теми плоскостями, в которых на единицу поверхности больше атомов, чем у других. К сожалению, в литературе нет данных о расчете ретикулярных плотностей решетки сапфира. Поэтому такой расчёт выполнен нами и будет представлен в главе 5.

3.1.2 Метод Доннея-Харкера

В 1937 г. Доннеем и Харкером был предложен новый принцип качественной оценки огранения кристаллов – закон Доннея-Харкера. Этот закон по сути своей является дополнением к закону Браве и методу Ретикулярных плотностей. Авторами закона было выявлено, что метод Ретикулярных плотностей хорошо работает на кристаллических решетках, не имеющих винтовых осей и не являющихся центрированными. Донней и Харкер предприняли попытку учесть центрированность решетки, наличие в ней винтовых осей и плоскостей скольжения [29,30].

Оценка огранения кристаллов сапфира методом Доннея-Харкера была сделана в работе П. Хартмана [43]. Хартман получил следующую последовательность граней в порядке убывания их размеров *a-R-r-c-n-S-P*.

3.1.3 Метод периодических цепей связей

Основой концепции периодических цепей связей (ПЦС) можно считать работу И.И. Шафрановского и В.И. Михеевой [44], предложивших рентгенометрический способ определения морфологически важных граней. Этот способ учитывает характер реальных связей между атомами в кристалле.

Наиболее полно концепция ПЦС разработана П. Хартманом для классификации кристаллических поверхностей [22]. Она учитывает характер связей между частицами и кристаллохимические особенности структуры.

Суть метода ПЦС заключается в следующем. Очевидно, что энергия связи пары атомов (молекул) убывает с увеличением расстояния между ними. Связь каждой пары атомов (молекул) можно представить в виде отрезка на прямой, соединяющей избранную пару, и при этом длину отрезка принять пропорциональной энергии связи. Тогда все связи в кристалле разобьются на группы, представленные отрезками одинаковой длины, причем отрезки определенной длины будут расположены цепями вдоль прямых с определенными кристаллографическими индексами. Цепи, состоящие из отрезков данной длины и направления, расположены в кристалле периодически и называются, поэтому, периодическими цепями связей [22,45].

Поверхностная энергия складывается из энергий тех связей, которые принадлежат к ПЦС, пересекающим данную поверхность. Таких ПЦС, вообще говоря, бесконечно много, причем вклад от ПЦС все более далеких соседей все меньше и меньше из-за убывания энергии соответствующих связей. Поэтому при грубых оценках поверхностной энергии кристаллов с ионными связями обычно ограничиваются первыми членами этой бесконечной суммы, отвечающими взаимодействию наиболее близких соседей.

Энергию связи можно оценить по теплоте испарения кристалла, считая, что число связей, разрываемых при испарении каждого строительного элемента (атома, молекулы и т.д.), вдвое меньше числа связей этого элемента с соседями в объеме кристалла [21].

Хартман выделяет в структуре периодические цепи сильных связей и предлагает разбивать на основе вышесказанного все возможные грани на F , S и K (рисунок 10) (по количеству параллельных им цепей связи).

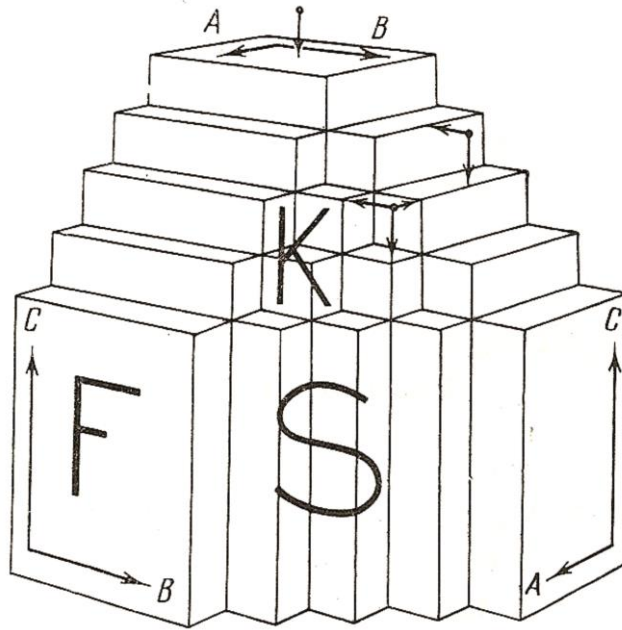


Рис. 10. Кристалл с тремя периодическими цепочками связей, где F-границы – плоские или грани, растущие слой за слоем (по Шубникову «физически возможные грани») и определяют форму кристалла; S-границы – ступенчатые грани (возникают, когда две параллельные ПЦС не соединены связями); K-границы – пирамидальные грани (образуются если в пределах d_{hkl} не существует ПЦС).

В методе ПЦС остаются открытыми следующие вопросы:

- как соотносятся между собой по структурной важности грани одного сорта в кристалле (например, грани одной простой формы, но разных символов). Так, Хартман выделяет в кристаллах типа алмаза в разряд S-граней все $\{hkk\}$.
- какие из них наиболее вероятны как реальные поверхности? В ПЦС-теории совершенно неразличимы геометрически подобные структуры с полярными и неполярными связями. Проблема разрыва полярных связей оказалась непреодолимой и для многих других теорий равновесной формы кристаллов, например, [45].

Хартман в своей работе [43] приводит данные по расчету методом ПЦС ограничения кристаллов сапфира. Для учета ионности связей он приводит результаты расчетов для элементарной ячейки и для некой площади:

- Для элементарной ячейки: $a > c > S > R > r > n > q$
- На единицу поверхности: $a > r > R > n > q > S > c$

Видно, что результаты расчетов отличаются друг от друга. Сам Хартман отмечает также, что его метод расчета ограничения не подходит для таких сложных структур, как у кристаллов сапфира, в связи со сложностью выделения ПЦС.

Существуют также и другие недостатки метода ПЦС. Теория ПЦС позволяет выделить гладкие, ступенчатые и шероховатые грани, но их иерархию не устанавливает; не различимы в

этой теории геометрически подобные структуры, например, алмаза и сфалерита [22]. Из-за отсутствия систематических данных об идеальном строении граней нельзя заранее ответить на вопрос об их стехиометрии, которая вызывает пристальный интерес в связи с проблемой получения совершенных гетероэпитаксиальных структур и может оказаться полезной в решении некоторых вопросов морфогенеза минералов, образованных в различающихся по химизму средах [46].

3.2 Молекулярные методы моделирования

3.2.1 Метод Ненасыщенности связей

В 1988 году М.Д. Любалиным был предложен метод количественной оценки ограничения кристаллов [20,32]. Предложенный метод опирался на метод Периодических цепей связей и был успешно применен автором для ряда кубических и тетрагональных кристаллов простого состава. Ниже дан краткий обзор метода.

В своей работе Любалин вводит понятие элементарного пакета. При росте кристаллов встраивание на их поверхности частиц определяется близкодействующими силами. Поэтому атомы, молекулы и их группировки не обязательно должны присоединяться к кристаллу слоями, содержащими все элементы симметрии структуры, т.е. слоями, имеющими толщину, заданную решеткой Бравэ (элементарной ячейкой структуры). Это можно увидеть уже при рассмотрении однокомпонентного кристалла с гомеоплярными связями, например, любого кристалла со структурой типа алмаза. Здесь в элементарной ячейке атомные плоскости куба отстоят друг от друга на расстояниях $0,25d_{001}$, все они эквивалентны, и на любой такой плоскости встраивание частиц в структуру происходит одинаково, так что каждая из плоскостей может быть граничной при росте или на уже сформированном кристалле. Таким образом, для грани куба толщина элементарного пакета составит $\Delta_{001} = 0,25d_{001}$.

Можно ли считать, что после присоединения только одной атомной плоскости на октаэдрической поверхности возникла бы новая граница при росте или на уже сформированном кристалле? По-видимому, нет, так как при последовательном отложении плоскостей должны были бы образовываться две атомные конфигурации с очень большим различием в свободной энергии. Какие из частиц на поверхности растущего кристалла вообще можно считать вошедшими в его структуру? Должно быть, это атомы, молекулы или их группировки, которые согласованы со структурой, «замыкают» ее своими связями, вливаясь в ансамбль частиц кристалла с присущими ансамблю колебаниями атомов у положений равновесия. Для октаэдрической плоскости в решётке алмаза это произойдет при присоединении двойного слоя атомов и толщина пакета составит $\Delta_{111} = d_{111}$ [20].

Таким образом, элементарный пакет атомных плоскостей – это минимальный по числу плоских сеток пакет атомов, переносом которых в направлении собственной нормали можно воспроизвести всю структуру. Наименьший вектор переноса определяет толщину элементарного слоя на грани Δ_{hkl} .

Далее Любалин вводит понятие ненасыщенности связей как характеристику кристаллизационной силы, действующей на элементарный пакет. Рост тонкими пакетами более вероятен. Анализ ненасыщенности связей показывает, однако, что не всегда их минимальному значению отвечают самые тонкие из возможных пакетов.

В простейших случаях элементарный пакет составлен одной атомной плоскостью. Иногда минимальная трансляция пакета в направлении нормали для воспроизведения структуры должна быть дополнена поворотами вокруг этой нормали и (или) трансляцией в плоскости пакета. При этом операции второго рода (зеркальные отражения) недопустимы. Простейший способ для выделения элементарных пакетов – это построение проекций кристаллической решетки на плоскость.

Для построения проекции сначала «метятся» атомы элементарной ячейки и определяется их положение на двух неколлинеарных направлениях, лежащих в плоскости проекций. Если на проекции одной элементарной ячейки узловых точек для построения плоского атомного узора, отражающего структурные особенности кристалла, недостаточно, исходная ячейка транслируется вдоль векторов решетки. Таким образом, добавляют конечное число атомов и соединяют их кратчайшими связями. Двух-трех плоских проекций достаточно для анализа практически всех важнейших граней в кристаллах, если предварительно построены их вертикальные срезы, позволяющие принять к рассмотрению нужное число атомов на каждой плоскости в тонкой структуре грани в пределах стандартно вычисленного межплоскостного расстояния d_{hkl} . Плоскости проекций выбираются перпендикулярно к главным осям кристалла или ярко выраженным ПЦС-векторам. Если возникают затруднения с выделением ПЦС в структуре или проводится поиск оптимальных геометрических согласований различных структур или их взаимодействия, можно построить проекцию любой произвольной рациональной плоскости.

Следующий этап – количественная оценка ненасыщенности связей (H). В первом приближении считается, что разрыв связи $i-j$ в вакууме приводит к возникновению на поверхностях двух ненасыщенных связей: на атоме i с энергией, пропорциональной $(r_i + r_j)^6 / (2^7 r_i^6)$, и на атоме j с энергией, пропорциональной $(r_i + r_j)^6 / (2^7 r_j^6)$, где r_i , и r_j – ковалентные радиусы атомов i и j . Гипотеза о зависимости H от радиусов атомов базируется на экспериментальных фактах, например на явлении самопроизвольного изгиба тонких пластинок (111) сфалеритоподобных кристаллов $A^{III}B^V$, при котором вогнутыми становятся поверхности A с меньшими значениями r [47].

При таком разделении энергий разорванной связи:

$$H_{hkl} = \sum nm \left[\frac{r_i(j, k, l \dots) + r_p(q, s, t \dots)}{r_{i,p}} \right]^6 \frac{E}{2^7 \Delta_{hkl}}, \quad (1)$$

где суммирование ведется по всем сортам атомов одного пакета и оборванных связей первой координационной сферы; n – число атомов одного сорта $i, j, k, \dots, p, q, s, \dots$; m – число одинаково наклоненных связей $i-k, j-k, \dots, i-p, j-p$; E – энергия единичной связи $i-p, i-j, j-k, \dots$ [20].

В гомодесмических структурах значения E различных связей будем считать одинаковыми и равными, если длины этих связей различаются в пределах первых процентов. В гетеродесмических структурах можно учитывать только самые длинные из связей первой координационной сферы. В случае, если необходимо оценить значение H граней, на которых обрываются связи, сильно различающиеся по длине, значение E следует приближенно считать обратно пропорциональным шестой степени длин этих связей [20]. В данной работе E принимается равным единице (как известно, об энергии этой единичной связи можно судить по теплотам сублимации кристаллов) [20].

Для кристаллов с неполярными связями последовательности H для разных hkl согласуются с последовательностями удельной поверхностной энергии, а для кристаллов с полярными связями возникает возможность оценить вероятность существования, по крайней мере в вакууме или на границе с собственным беспримесным расплавом, разноразных внешних атомных плоскостей. Переход от энергетической к силовой характеристике единичной поверхности позволяет более ощутимо вскрыть анизотропию кристаллизационных свойств вещества. Кроме того, оценка всех нормальных ($H_{hkl\perp}$) и тангенциальных ($H_{hkl\parallel}$) составляющих ненасыщенности дает возможность с помощью параметра $X = H_{hkl\parallel}/H_{hkl\perp}$ численно охарактеризовать атомную шероховатость идеальной поверхности. Вероятность появления, устойчивость и степень развития на кристаллах граней возрастают с уменьшением значений H и X .

Если при подсчете H_{hkl} принимаются во внимание только кратчайшие межатомные связи, то для ряда структур с координационными числами не более шести существуют грани с $H_{hkl}=0$ и $X_{hkl}=0$. Это сингулярные наиболее устойчивые грани. В кристаллах со структурами, характеризующимися большими координационными числами (8, 12), существует по несколько сингулярных неэквивалентных граней с H_{hkl} и X_{hkl} больше нуля. Рост сингулярных граней в отсутствие дислокаций требует преодоления весьма существенного активационного барьера и возможен только после образования на грани определенных размеров двумерного зародыша.

В работах М.Д. Любалина не дается расчета методом Ненасыщенности связей для кристаллов сапфира, поэтому такой расчет сделан нами в главе 5.

3.2.2 Результаты расчетов программой METADISE

В работе [41] приведены данные по расчету поверхностных энергий различных плоскостей сапфира, выполненные в программном пакете METADISE. Данный программный пакет является коммерческим продуктом, поэтому алгоритм расчета не приведен. Известно, что для расчета поверхностных энергий программа использует параметры кристаллической решетки, размеры атомов, расстояние между ними и химические потенциалы.

Последовательность граней в порядке возрастания поверхностной энергии, получившаяся в результате расчёта, следующая: $R - c - a - n - m - r - P$ [41].

3.3 Методы квантового моделирования

Квантовые модели основаны на различных подходах к расчету потенциала межатомных взаимодействий. Они учитывают распределение электронных плотностей в кристаллах.

Существует множество работ, где приводятся данные по расчету поверхностных энергий для кристаллов сапфира.

Результаты расчетов поверхностных энергий кристаллов сапфира квантовыми методами приводятся в работах [33,34,35,36,37,38,39,40,41,42]. В таблице 2 представлен порядок граней по возрастанию поверхностной энергии по результатам работ, в которых рассмотрено наибольшее количество кристаллографических направлений. В таблицу включены только результаты расчётов с учётом релаксации. В большинстве из приведенных работ дается результат расчета при 0 К и только в [39] приведено сравнение энергий релаксированных и нерелаксированных поверхностей при 300 К, 1500 К, 1700 К, но при этом с температурой изменяются только численные значения энергий, а порядок следования граней сохраняется.

Таблица 2 – Результаты расчетов с применением различных квантовых моделей

Ссылка	Порядок граней по возрастанию поверхностной энергии	Примечание
[33]	$r - a - m - c - R$	-
[34]	$c - m - R - a - r$	Расчёт из первых принципов
[37]	$m - c - a - r - R$	Расчёт из первых принципов
[37]	$c - m - r - a - R$	Модель парного взаимодействия
[38]	$c - m - R - a - r - p$	Используется потенциал Макродта
[38]	$R - c - a - m - p - r$	Используется эмпирический потенциал
[38]	$R - a - c - p - r - m$	Используется потенциал «Ab initio»
[40]	$R - c - a - r$	-

Из таблицы видно, что расчеты поверхностных энергий квантовыми методами плохо согласуются между собой, что по-видимому связано как раз с различным подходом к расчету потенциала межатомного взаимодействия.

4 ИЗУЧЕНИЕ МОРФОЛОГИИ БОКОВОЙ ПОВЕРХНОСТИ МОНОКРИСТАЛЛИЧЕСКИХ СТЕРЖНЕЙ САПФИРА

Первые работы по изучению морфологии боковой поверхности цилиндрических монокристаллов проводились на германии, кремнии и полупроводниках A^3B^5 [3,4,5].

Морфология кристаллов корунда сначала природных, а потом и искусственно выращенных, изучается уже давно. Существует целый ряд работ по экспериментальному изучению монокристаллов корунда, выращенных в различных условиях различными методами [6,41,42,48,49], но лишь только одна из них описывает ограничение боковой поверхности монокристаллических стержней сапфира, выращенных способом Степанова [6]. В [6] для изучения ограничения кристаллов сапфира, используется оптический прибор – фотогониограф.

4.1 Метод фотогониографии

Концепция фотогониографа была предложена Антоновым П.И. в 70-х годах 20-ого века [3]. Прибор был изготовлен на базе ФТИ им. А.Ф. Иоффе. Ранее он успешно использовался для получения разверток боковой поверхности оптически не прозрачных кристаллов германия и антимонида индия. Фотограмма стержня антимонида индия диаметром 10 мм приведена на рисунке 11 [5].

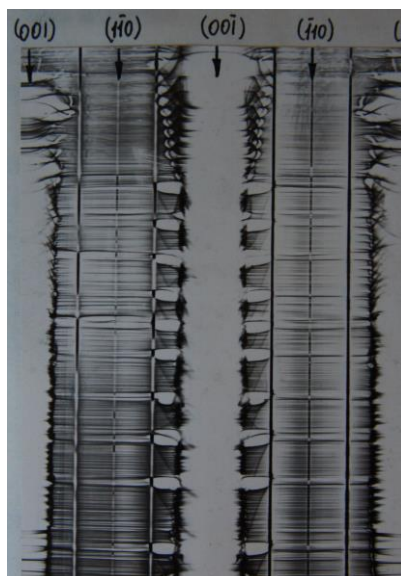


Рис. 11. Фотограмма кристалла антимонида индия [5].

Принцип действия фотогониографа состоит в следующем (рисунок 12).

Исследуемый кристалл 1 освещается параллельным пучком света 2. Луч, падающий на поверхность кристалла под углом 45° и отражающийся от нее, попадает через щель 3 на барабан с фотобумагой 4. При синхронном вращении кристалла и барабана на фотобумаге регистрируется фотограмма боковой поверхности кристалла.

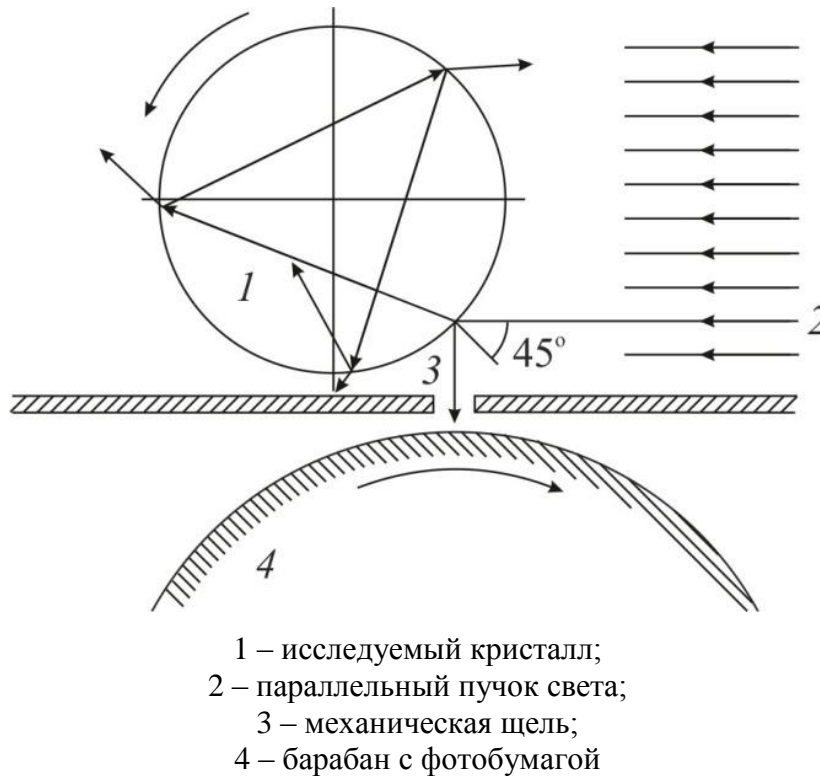


Рис. 12. Схема устройства фотогониографа [6].

Зеркальные грани параллельные поверхности вытягивания, проявляются на фотограммах в виде узких черных полос. На практике из-за неизбежных ошибок ориентации затравки такие грани всегда наклонены к направлению вытягивания на небольшой угол. Если грань наклонена в сторону расплава, то она по-прежнему проявляется в виде зеркальной плоскости, содержащей отдельные ступени, и отображается на фотограмме черной полосой. Если грань оказывается отклоненной в противоположную сторону, то она может не образовать зеркальной плоскости. В соответствующей угловой позиции обычно наблюдается небольшой выступ или излом поверхности, проявляющийся на фотограмме в виде белой полосы. Грани кристалла наклоненные к направлению вытягивания на большие углы и обращенные в процессе выращивания к расплаву, проявляются на кристалле в виде достаточно широкой, обычно шероховатой, полосы и также хорошо регистрируются на фотограммах.

Применимость этого метода к прозрачным монокристаллам сапфира вызывала сомнения по двум причинам. Во-первых, интенсивность отражения света от поверхности прозрачного

кристалла значительно ниже, чем от практически зеркальной поверхности таких монокристаллов, как германий. Однако расчет по формулам зависимости коэффициента отражения от угла падения луча на границу раздела двух сред [50] показал, что при рабочем угле падения 45° интегральный коэффициент отражения для оптически гладкой поверхности сапфира равен $\sim 10\%$, что вполне достаточно для регистрации изображения светочувствительным материалом.

Во-вторых, лучи освещающего потока, претерпевающие после преломления многократные отражения в кристалле. Если бы такие лучи могли попасть в регистрационную щель прибора, то на фотограмме имело бы место наложение изображений различных участков поверхности кристалла. Детальное рассмотрение хода преломляющихся и отражающихся лучей в прозрачном кристалле показало [6], что возникновение ложных изображений практически невозможно. Для примера на рисунке 12 показан ход лучей для первых трех отражений внутри исследуемого кристалла. Луч, вошедший в кристалл в точке рабочего отражения, показан сплошной линией. Пунктирной линией показан луч, который мог бы дать ложный рефлекс. Из рисунка видно, что для того, чтобы попасть в регистрирующую щель этот луч должен входить в кристалл с направлений, не параллельных падающему пучку. Таких лучей в приборе нет. Следует также отметить, что при каждом отражении внутри кристалла 90% процентов света будет выходить из кристалла и после нескольких отражений интенсивность этого луча станет пренебрежимо малой. Можно предположить, что многократное отражение большого числа лучей и их рассеяние на неровностях боковой поверхности кристалла и дефектах в его объеме даст некоторую паразитную засветку фотоматериала, но не может привести к появлению ложных изображений.

Практика съемки боковой поверхности монокристаллов сапфира [6] показала, что на фотограммах регистрировались лишь значимые рефлексы – паразитные отражения не попадают на фотобумагу. В то же время фотограмма дает наглядное изображение всей боковой поверхности и позволяет определять угловые расстояния между гранями. Для приведенной на рисунке 12 схемы, угловая разрешающая способность прибора при определении положения рефлекса от зеркальной грани составляет (град)

$$\delta \approx 45 \frac{b}{R} \quad (2)$$

где $2b$ - ширины щели, а R радиус кристалла. Для использованных в цитируемой работе значений $b=0,1$ мм и $R=5$ мм величина δ составляет $0,9^\circ$ [6].

4.1.1 Исследование огранения монокристаллических стержней сапфира методом фотогониографии

В работе [6] приведены результаты изучения методом фотогониографии огранения боковой поверхности монокристаллических стержней сапфира различной ориентации (направления вытягивания).

4.1.1.1 Кристаллы ориентации c [0001]

В начале работы [6] приводятся фотограммы монокристаллических стержней сапфира, выращенных способом Степанова ориентации c , как наиболее практически важной.

Фотограмма кристалла сапфира ориентации c , выращенного с использованием формообразователя диаметром 9,2 мм, приведена на рисунке 13. Буквенные обозначения граней даны по [8].

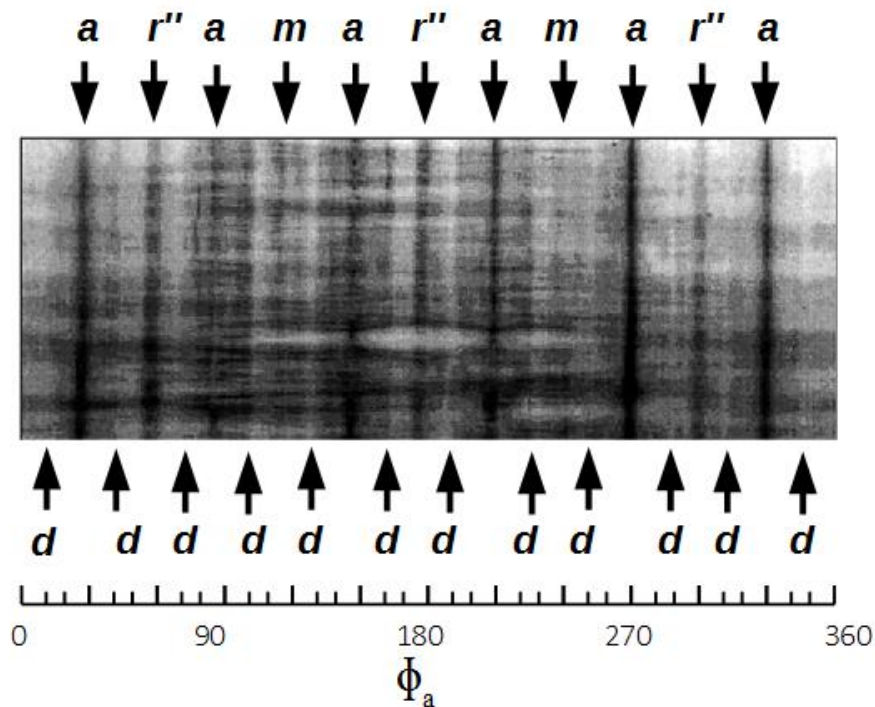


Рис. 13. Фотограмма монокристаллического стержня сапфира ориентации c [0001]. Φ_a – угол на фотограмме, заданный в градусах [6].

Было выявлено, что в огранении боковой поверхности принимают участие только грани трех типов. Грани ромбоэдра r , проявляющиеся в этой ориентации в виде наклонных граней на фронте кристаллизации, на боковой поверхности дают три дорожки из наклонных ступеней, обозначенные на рисунке 13, как r'' .

В виде граней, параллельных направлению вытягивания, для этой ориентации могут проявляться грани трех простых форм: гексагональных призм m $\{10\bar{1}0\}$ и a $\{11\bar{2}0\}$ и семейства дигексагональных призм $\{hki0\}$. На всех выращенных кристаллах обнаружено присутствие ясно выраженных граней призмы a . На всех фотограммах также наблюдаются 12 относительно слабых рефлексов дигексагональной призмы d . Ширина этих граней мала. При сильном развитии дорожек r'' они частично маскируют рефлексы d . В работе [6] отмечается также присутствие граней призмы m , но она проявляется очень слабо и зачастую полностью размывается наклонными гранями r'' .

Список граней выявленных в результате фотогониографического исследования боковой поверхности на кристаллах ориентации c $\{0001\}$ приведен в таблице 3.

Таблица 3 – Установленные грани на монокристаллических стержнях сапфира ориентации c методом фотогониографии

Общепринятое обозначение	a	d	m
Угол с плоскостью a	0	19,2	30

4.1.1.2 Кристаллы ориентации a $\{11\bar{2}0\}$

Ориентация a дает уникальную возможность сравнить проявление на боковой поверхности монокристаллических стержней сапфира граней c и r , поскольку в этом случае обе грани ориентированы вдоль цилиндрической поверхности кристалла и в процессе роста находятся в одинаковых капиллярных и тепловых условиях. В работе [6] изучалось ограничение специально выращенного для этой цели монокристалла диаметром 10 мм с предельно малой ошибкой ориентации (не более 3-5' по обеим граням).

На фотограммах кристаллов ориентации a (рисунок 14) проявляются четко выраженные рефлексы от двух граней пинакоида c и двух граней ромбоэдра r . Было установлено, что ширина этих граней различна на различных участках кристалла, но грань c всегда шире. Максимальные значения ширины составили ≈ 3 мм для грани c и $\approx 1,5$ мм для грани r . Таким образом, в первом приближении можно считать, что на этом кристалле грань базиса в два раза шире грани ромбоэдра.

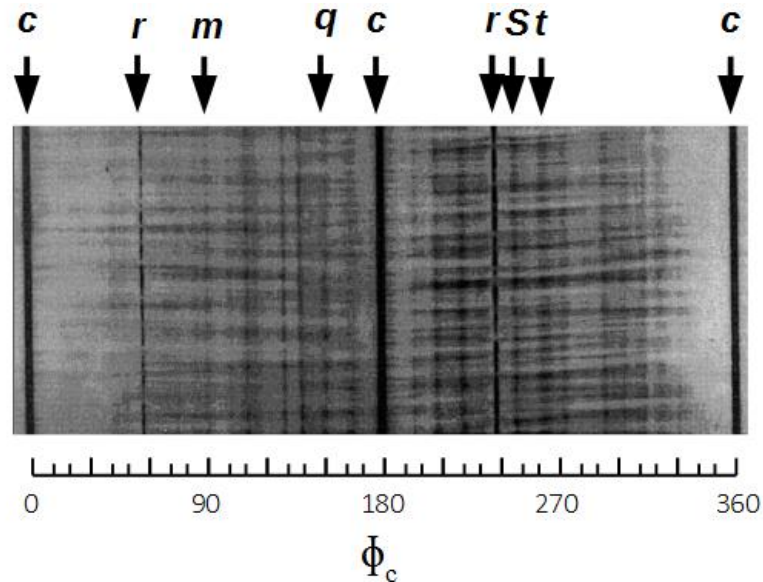


Рис. 14. Фотограмма монокристаллического стержня сапфира ориентации $a \{11\bar{2}0\}$. ϕ_c – угол на фотограмме, заданный в градусах [6].

На фотограммах кристаллов ориентации a также наблюдается значительное количество слабо выраженных рефлексов. Из кристаллографии корунда ясно, что в рассматриваемой ориентации в качестве зеркальных граней кроме пинакоида c и призмы m могут проявляться только грани семейства ромбоэдров $\{h0\bar{h}l\}$. Наблюдаемые рефлексы идентифицированы как принадлежащие формам S , t , q и ряду ромбоэдров, не имеющих общепринятых обозначений. Установленные грани приведены в таблице 4.

Таблица 4 – Установленные грани на монокристаллических стержнях сапфира ориентации a методом фотогониографии

Общепринятое обозначение	c	-	-	-	q	-	-	r	-	S	t	m
Угол с плоскостью c	0	16	17,6	24,2	32,2	42	51,6	57,6	68,4	72,4	81	90

4.1.1.3 Кристаллы ориентации $m \{10\bar{1}0\}$

Аналогично кристаллам ориентации a , кристаллы ориентации m дают возможность сравнить грани c и a . Результаты исследования методом фотогониографии приведены в работе [6], где они выполнялись на кристаллах диаметром 6,4 мм. Фотограмма одного из таких кристаллов приведена на рисунке 15.

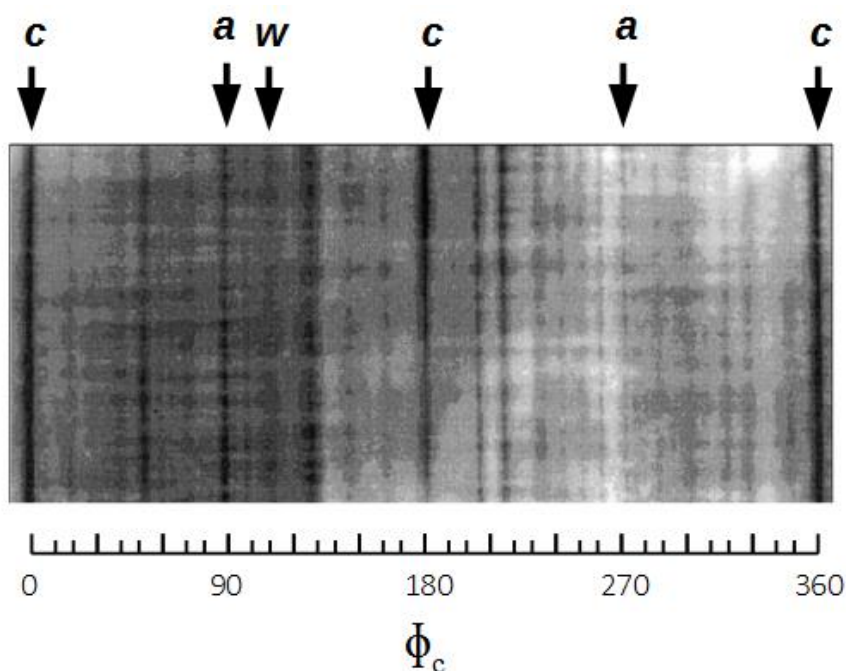


Рис. 15. Фотограмма монокристаллического стержня сапфира ориентации $m \{10\bar{1}0\}$. ϕ_c – угол на фотограмме, заданный в градусах [6].

Фотограмма хорошо иллюстрирует, что присутствуют развитые грани пинакоида шириной до 0,7-0,8 мм и слабые грани призмы a , ширина которых не превышает 0,1 мм. Таким образом, по огранке кристаллов, выращенных с такой ориентацией авторы [6] делают вывод, что грань c в 7-8 раз шире грани a .

Как и в случае с кристаллами ориентации a , на фотограмме наблюдается ряд слабо выраженных рефлексов. В соответствии с кристаллографией корунда они могут принадлежать только семейству гексагональных дипирамид $\{hh\bar{2}hl\}$, причем каждый рефлекс должен повторяться на фотограмме четыре раза в симметричных положениях относительно отражений от граней c и a .

Помимо рефлексов от граней c и a на рисунке 15 отмечены рефлексы от гексагональной дипирамиды w . Также обнаружены рефлексы от граней, не имеющих общепринятых обозначений. Все выявленные грани приведены в таблице 5.

Таблица 5 – Установленные грани на монокристаллических стержнях сапфира ориентации m методом фотогониографии

Общепринятое обозначение	c	-	-	w	-	a
Угол с плоскостью c	0	42,4	53,8	69,8	79,6	90

Авторами в [6] отмечается, что полученные данные по кристаллам всех трех ориентаций нужно рассматривать как предварительные, поскольку многие из этих рефлексов недостаточно четкие, что приводит к большим погрешностям в определении угла.

4.1.2 Недостатки метода фотогониографии

Принцип фотогониографии и сам фотогониограф не только являются незаменимыми для изучения морфологии боковой поверхности цилиндрических монокристаллов, но и не имеют аналогов в мире. Несмотря на это, фотогониограф имеет ряд недостатков, не позволяющих полноценно использовать его для детального описания морфологии боковой поверхности цилиндрических монокристаллов:

1. Метод фотогониографии является методом качественного анализа морфологии боковой поверхности цилиндрических монокристаллов;
2. Метод фотогониографии имеет очень низкое разрешение, связанное с размером щели (≈ 1 мм) и размером зерна фотобумаги. В связи с этим разрешающая способность фотогониографа равняется $1-2^\circ$, что не позволяет фиксировать небольшие особенности морфологии боковой поверхности цилиндрических монокристаллов;
3. Метод фотогониографии исключает цифровую обработку результатов анализа из-за использования фотографического метода регистрации.
4. Метод фотогониографии даёт ошибку при определении положения широких зеркальных граней в силу использования пучка света, падающего не нормально к поверхности кристалла.

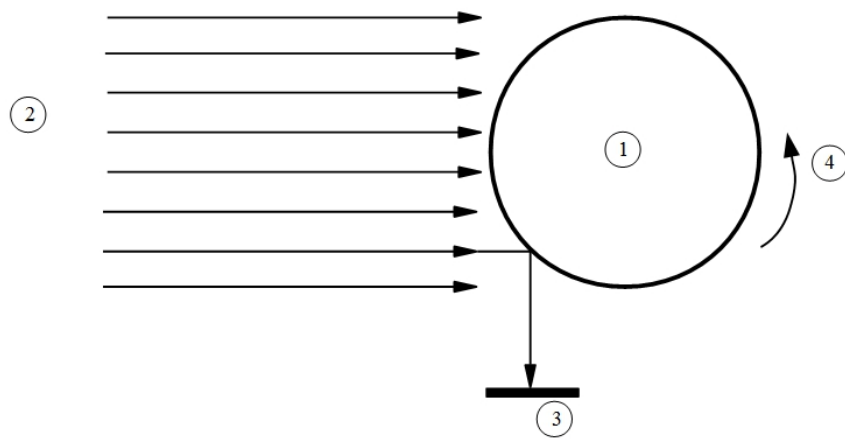
В связи с указанными недостатками метода фотогониографии, нами было принято решение о создании на его базе нового оптического прибора с цифровой регистрацией – видеогониографа.

4.2 Конструкция видеогониографа [51]

На базе ФТИ им. А.Ф. Иоффе был разработан и создан действующий лабораторный макет видеогониографа (в дальнейшем – видеогониограф). Конструкция и принцип действия видеогониографа основаны на принципах работы фотогониографа и заключаются в следующем (рисунок 16).

Исследуемый кристалл 1 освещается параллельным пучком света 2. Луч, падающий на поверхность кристалла под углом 45° и отражающийся от нее, попадает на матрицу цифровой камеры высокого разрешения 3, сигнал с которой передаётся в компьютер по каналу связи

USB 2.0, где обрабатывается специализированной программой по захвату, обработке и сохранению полученных данных (рисунок 17).



- 1 – исследуемый кристалл;
- 2 – параллельный пучок света;
- 3 – матрица цифровой видеокамеры;
- 4 – направление вращения кристалла.

Рис. 16. Схема устройства видеогониографа

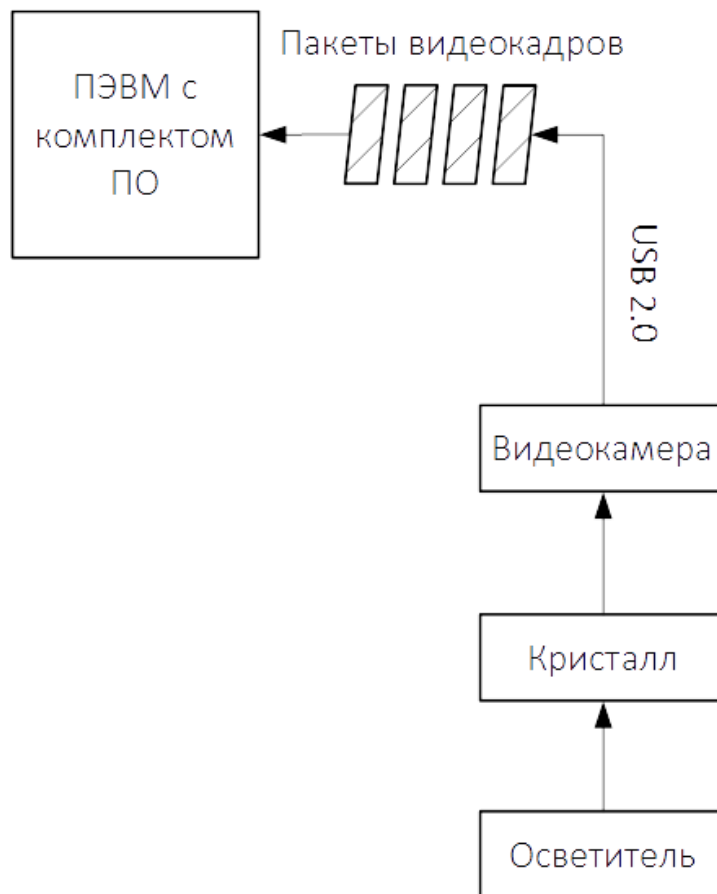


Рис. 17. Функциональная схема видеогониографа

На рисунке 18 приведено схематическое изображение конструкции видеогониографа. Прибор состоит из следующих частей:

1. Модуль анализатора:
 - 1.1. Жесткий стальной корпус на регулировочных ножках, выставляемый по круглому пузырьковому уровню;
 - 1.2. Шток камеры;
 - 1.3. Фиксатор штока камеры, жестко закрепляющий шток камеры на жестком стальном корпусе;
 - 1.4. Столик видеокамеры, перемещающийся по штоку камеры и фиксирующийся винтом после выставления на нужный уровень;
 - 1.5. Изучаемый кристалл;
 - 1.6. Цифровая видеокамера без объектива, служащая приемником отражений от кристалла;
 - 1.7. Верхний держатель, закрепленный на штоке верхнего держателя, закрепленный на жестком стальном корпусе с возможностью регулировки кристалла по отвесу;
 - 1.8. Мотор с редуктором, вращающий кристалл с необходимой скоростью;
2. Осветитель, создающий параллельный пучок света.

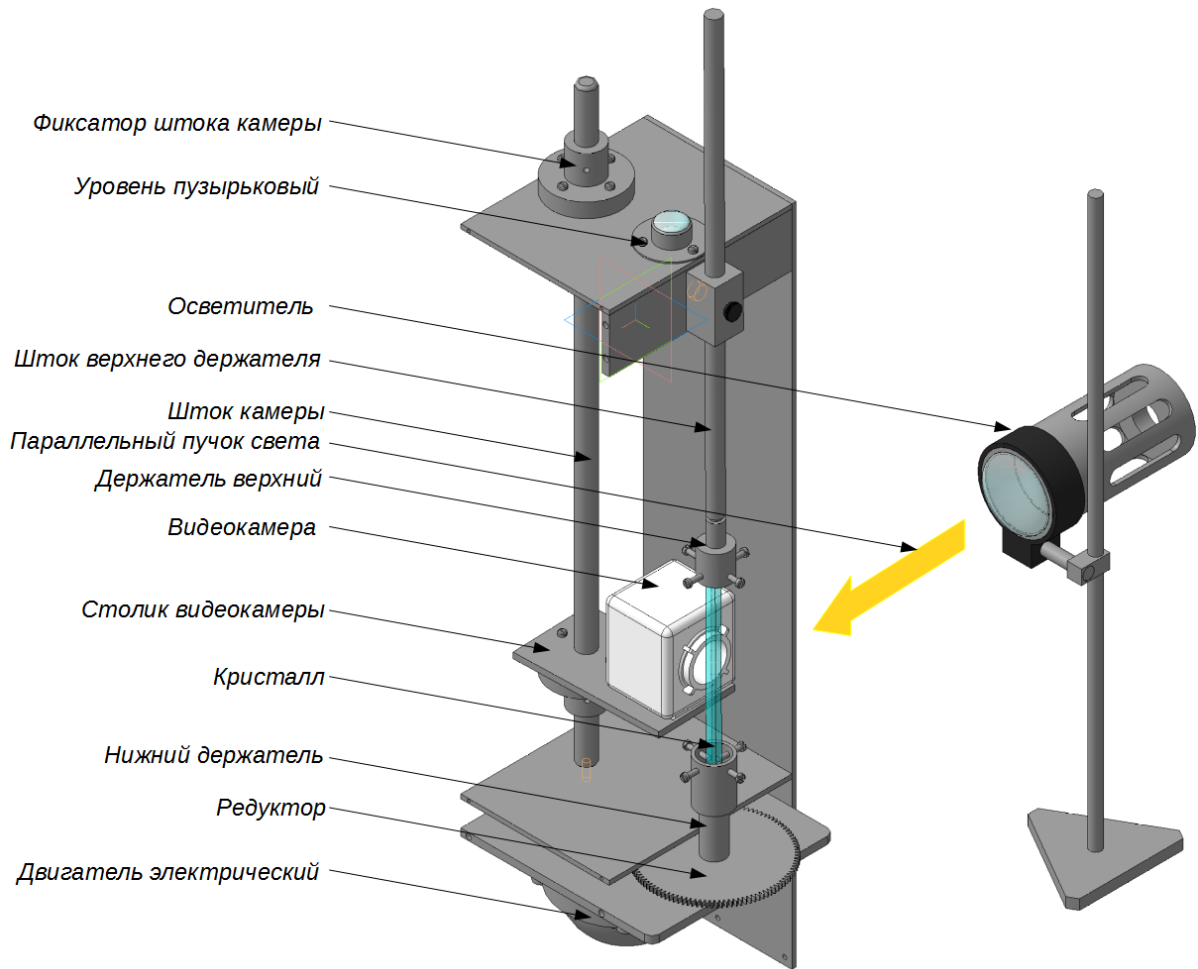
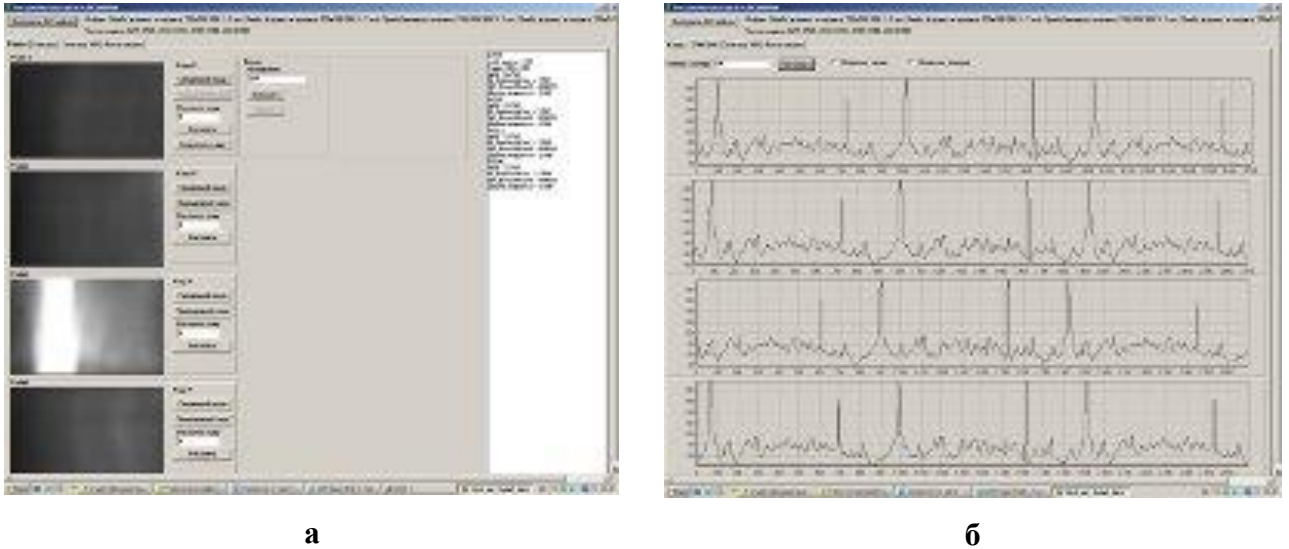


Рис. 18. Схематическое изображение конструкции видеогониографа

Обработка потока видеoinформации, получаемой телекамерой, происходит следующим образом. Как следует из рисунка 17, пучок света, после попадания на кристалл, отражается от его поверхности и попадает на матрицу цифровой видеокамеры. Дальнейший алгоритм обработки данных заключается в следующем.

Изображения захватывается в формате *.avi и сохраняется на жестком диске компьютера в виде отдельного видео файла длительностью не менее времени полного оборота кристалла. Данная процедура производится путем применения стандартного программного обеспечения и драйверов, идущих в комплекте с цифровой видеокамерой.

Для анализа видеофайлов нами было написано специализированное программное обеспечение по захвату, обработке и сохранению полученных данных – ЗОС (рисунок 19).



а – главное окно ЗОС
б – окно анализа раскадровки

Рис. 19. Основные окна программы по захвату, обработке и сохранению данных

Алгоритм работы ЗОС состоит в следующем.

С помощью диалогового окна файл или группа файлов (ЗОС позволяет анализировать одновременно до 4 видеофайлов) загружается в программу, которая производит его анализ и раскадровку (рисунок 19а). В специальном окне выводится один кадр, причём предусмотрена функция выбора кадра, путем введения номера кадра в диалоговое окно.

Далее производится выбор электронного аналога механической щели фотогониографа – вертикальной полоски на кадре, по которой в дальнейшем будет строиться видеограмма (рисунок 20).

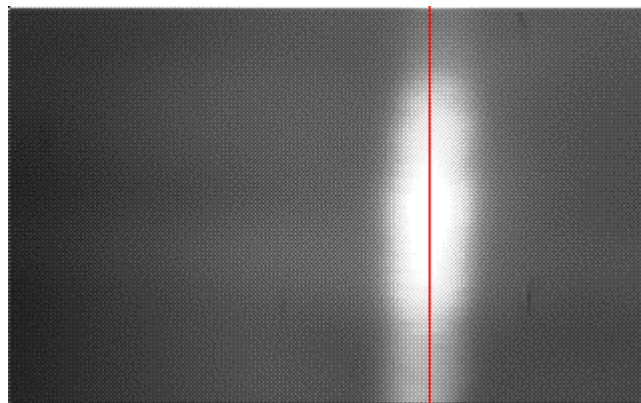


Рис. 20. Один кадр видеофильма, на котором хорошо виден блик от грани с отмеченной цифровой щелью – аналогом механической щели фотогониографа.

Такая щель выбирается из каждого кадра, а её ширина в видеогониографе равняется ширине одного столбца пикселей – в нашем случае эта величина равняется 0,02 мм (против ~1 мм у фотогониографа). Выбранная цифровая щель соответствует тому лучу, который попадает на поверхность кристалла под углом 45° . Необходимо отметить, что такую щель не только крайне сложно получить механическим путем, но также при использовании столь малой механической щели могут возникать нежелательные оптические эффекты. Благодаря такой узкой щели разрешающая способность видеогониографа равняется $0,2^\circ$.

После выбора щели строится график относительных интенсивностей отражений по всему отснятому видеофайлу (рисунок 19б). Такое построение производится путем усреднения значений относительных интенсивностей по высоте кадра.

Во время съемки из-за медленной скорости работы канала связи, персонального компьютера и операционной системы неизбежно возникают пропущенные кадры в видеофайле. Обычно таких пропусков не более 0,3-0,5 %. ЗОС предлагает исправить пропуски. Алгоритм исправления пропусков заключается в замене пропущенных кадров усредненным по значениям относительной интенсивности отражения между предыдущим и следующим кадрами.

После построения графиков относительной интенсивности по всему видеофайлу производится построение такого графика в пределах $0 - 360^\circ$ путем выбора реперных точек. Как правило, такими реперными точками являются отражения от хорошо выраженных граней. В силу кристаллографии корунда все простые формы дают попарно параллельные грани, что сильно облегчает выбор таких реперных точек.

Далее по выбранному интервалу строится видеограмма путем последовательного составления столбиков пикселей (щелей). Высота видеограммы равняется высоте матрицы камеры и в нашем случае равна 8 мм.

После построения видеограммы она может быть сохранена в виде отдельного графического файла. Данные по каждому сечению кристалла выгружаются в отдельный текстовый файл в цифровом виде, который содержит угол и значение относительной интенсивности отражения для этого угла.

Определение уровня шума видеогониографа проводилось посредством съемки полированного стержня. После съемки данные по относительной интенсивности отражения усреднялись и рассчитывалось стандартное отклонение. Уровень шума видеогониографа составляет 5-7 единиц относительной интенсивности отражения.

В параграфе 4.3 приведены результаты исследования морфологии боковой поверхности монокристаллических стержней сапфира различной ориентации.

4.3 Видеogramмы боковой поверхности монокристаллических стержней сапфира различной ориентации [51]

В данном разделе приводятся результаты изучения морфологии боковой поверхности монокристаллических стержней сапфира различной ориентации, выращенных способом Степанова (см. параграф 2.2).

4.3.1 Ориентация a $[11\bar{2}0]$

На рисунке 21 приведена видеogramма и график относительных интенсивностей отражения от боковой поверхности цилиндрического монокристалла сапфира ориентации a диаметром 8 мм, выращенного способом Степанова.

На графике относительных интенсивностей по оси ординат отложено значение интенсивности отражения от боковой поверхности в относительных единицах, по оси абсцисс дан угол φ_c – угол, откладываемый в данном случае от грани пинакоида c (как уже говорилось ранее – одной из граней, используемой нами в роли реперной). На графике также отмечен уровень шума (5-7 относительных единиц относительной интенсивности отражения). Хорошо видно, что уровень шума в сравнении с интенсивностью пиков на графике незначителен, поэтому фактически все пики являются значимыми и отражают особенности морфологии боковой поверхности цилиндрических монокристаллов сапфира.

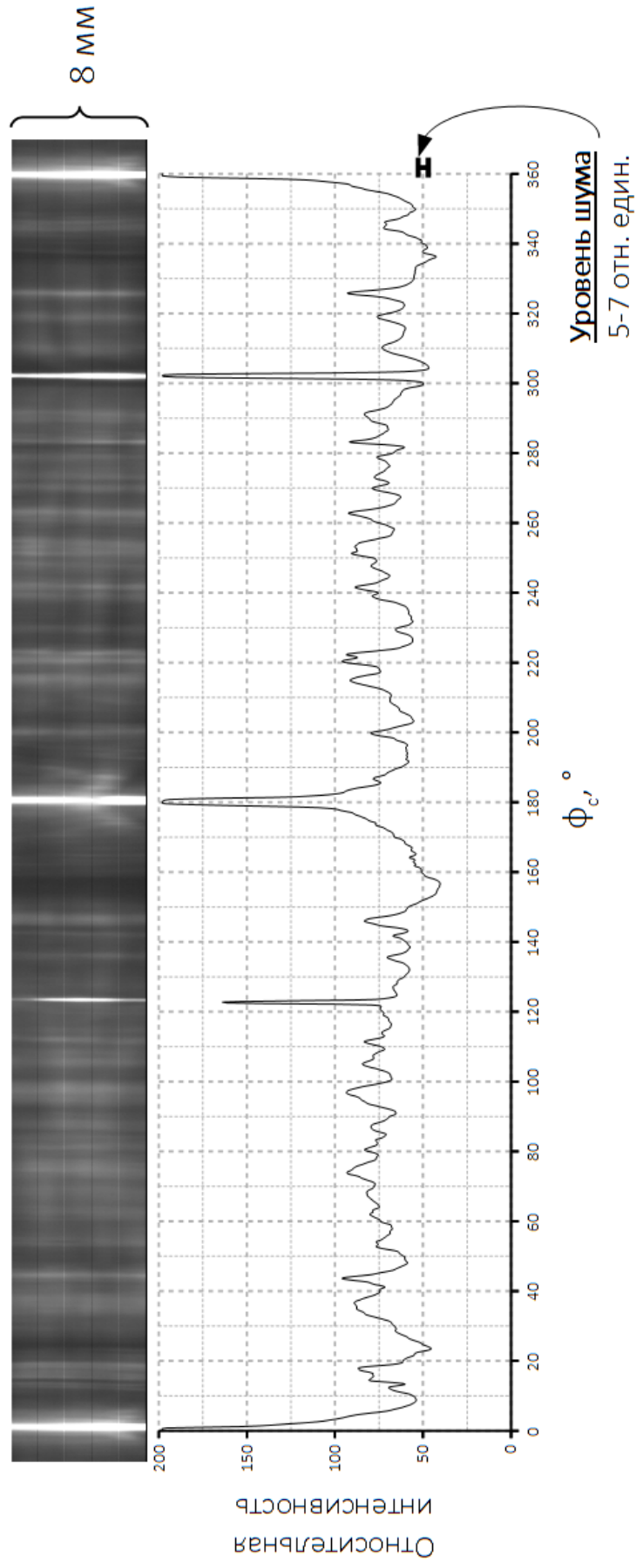


Рис. 21. Видеограмма и ее график относительных интенсивностей для одного из сечений боковой поверхности кристаллов сапфира ориентации *a*.

Так как метод видеогониографии является количественным методом изучения морфологии боковой поверхности монокристаллических стержней сапфира и дает на выходе график относительных интенсивностей отражения от боковой поверхности появляется возможность сравнивать не только морфологию нескольких кристаллов, но и сравнивать морфологические особенности нескольких уровней на одном кристалле. Это предоставляет возможность проследить изменение морфологии по длине кристалла, говорить о стабильности процесса и о том насколько стабильно сам кристалл держит сечение. Так как все кристаллы выращивались в максимально схожих условиях с максимально стабильными параметрами в процессе роста, можно оценить насколько стабильно кристалл держит сечение. На рисунке 22 приведены графики относительных интенсивностей для нескольких уровней одного кристалла сапфира ориентации *a*. На графиках также отмечены грани, имеющие общепринятые буквенные обозначения.

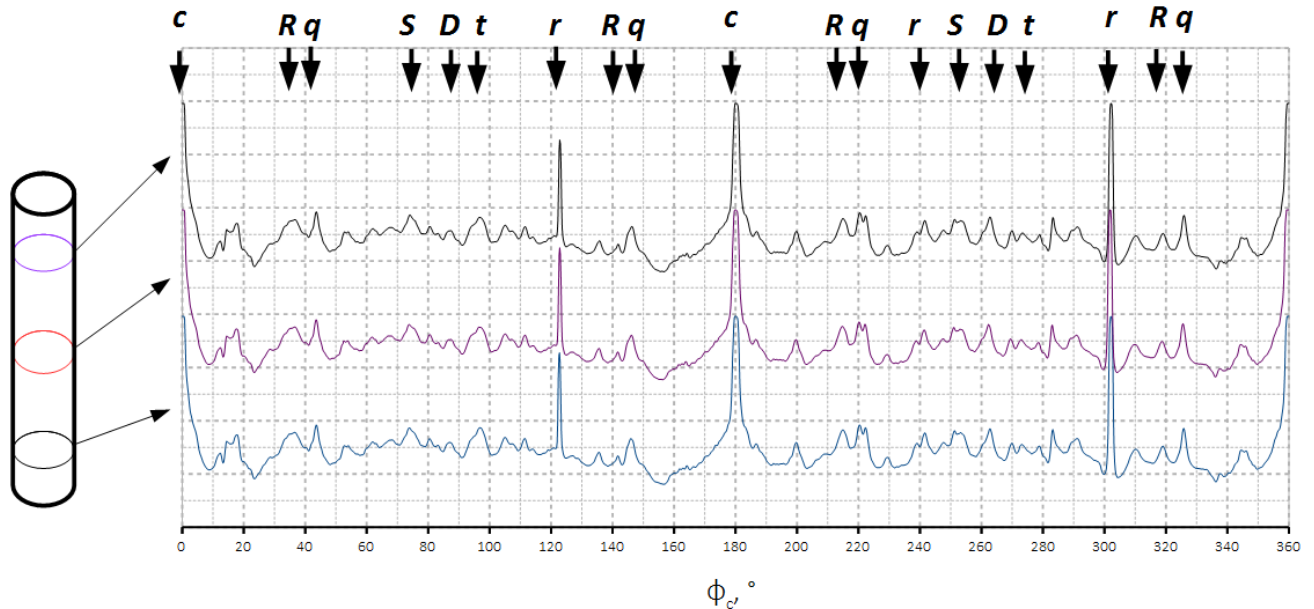


Рис. 22. Сравнение особенностей морфологии боковой поверхности цилиндрического монокристалла сапфира ориентации *a* на нескольких уровнях.

Графики, приведенные на рисунке 22, хорошо иллюстрируют, что морфология боковой поверхности цилиндрических монокристаллов ориентации *a* меняется незначительно (в пределах значений уровня шума).

Рисунки 21 и 22 хорошо иллюстрируют, что наиболее ярко выраженными являются грани пинакоида *c* и ромбоэдра *r*, что согласуется с визуальными наблюдениями. Форма пиков *c* и *r* различна, что связано с морфологией этих граней.

Микроскопическое наблюдение показывает, что в случае грани c переход к округлой поверхности происходит плавно путем сгущения ступеней, что проявляется на графике интенсивности в виде пьедестала у основания пика. У грани r такого пьедестала не наблюдается, но и на кристалле переход от плоской грани к округлой части происходит резко с четким ребром между гранной и округлой поверхностью. Такие ребра на графике проявляется в виде провалов с двух сторон от пика r .

На боковой поверхности монокристаллов сапфира данной ориентации методом видеогониографии выявлено большое количество второстепенных граней ромбоэдров, намного превышающее количество граней, выявляемых методом фотогониографии. Полный перечень граней, выявленных нами в ходе видеогониографического изучения морфологии боковой поверхности монокристаллических стержней сапфира ориентации a приведен в таблице 6.

Таблица 6 – Простые формы, выявленные на боковой поверхности цилиндрических монокристаллов сапфира ориентации a методом видеогониографии

Обозначение	c	-	-	-	-	q	R	-	-	-
ϕ_c	0	12,6	16	21,6	29,4	32,2	38,2	42	44,6	49,8
Обозначение	-	-	r	-	S	a	t	-	m	D
ϕ_c	51,6	57	57,6	68,4	72,4	75,8	81	87,2	90	95,6
Обозначение	p	-	S	r	-	-	-	R	-	-
ϕ_c	97,2	103	107,6	122,4	128,4	130,2	131,6	141,8	155,8	162,4
Обозначение	-									
ϕ_c	168,8									

На кристаллах ориентации a выявлена 31 грань, из них 20 ранее не описывалось в ограничении боковой поверхности цилиндрических монокристаллов сапфира.

Из графика (рисунки 21 и 22) видно, что интенсивность пиков r различна, что свидетельствует о наличии небольшой остаточной разориентации кристалла. Об этом свидетельствует также проявление некоторых параллельных плоскостей в виде граней или ребер.

На кристаллах данной ориентации в виде зеркальных граней проявляются только грани пинакоида c и положительного ромбоэдра r . Необходимо отметить, что соотношение размеров граней c и r на кристалле $c:r$ как 2:1-2,5:1 – ширина пиков этих граней на середине высоты имеет такое же соотношение. Данное наблюдение совпадает с установленным ранее соотношением в работе [6].

4.3.2 Ориентация c [0001]

На рисунке 23 приведена видеограмма и график относительных интенсивностей отражения от боковой поверхности цилиндрического монокристалла сапфира ориентации c , выращенного способом Степанова.

На графике относительных интенсивностей для ориентации c аналогично такому же графику ориентации a фактически все пики являются значимыми.

Аналогично кристаллам ориентации a для кристаллов ориентации c был проведен анализ особенностей морфологии боковой поверхности на нескольких уровнях цилиндрического монокристалла (рисунок 24), где также отмечены грани имеющие общеупотребительные символы.

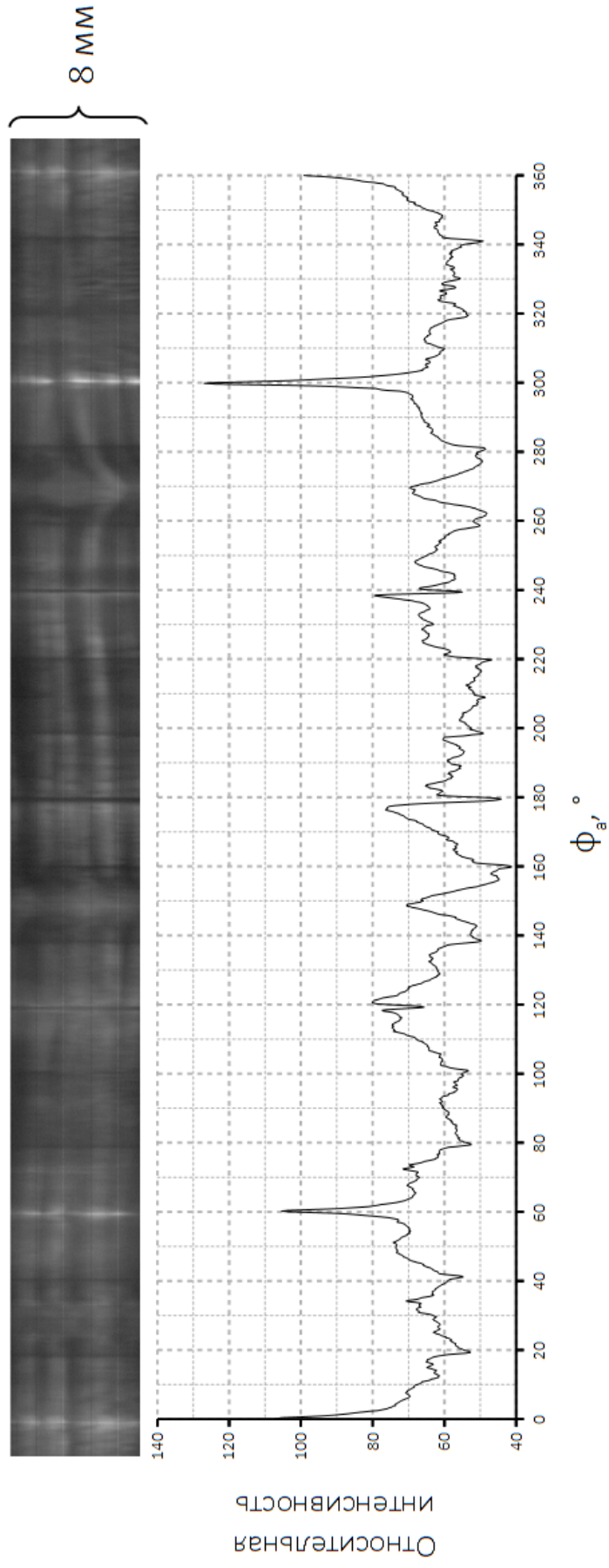


Рис. 23. Видеограмма и ее график относительных интенсивностей для одного из сечений боковой поверхности кристаллов сапфира ориентации c .

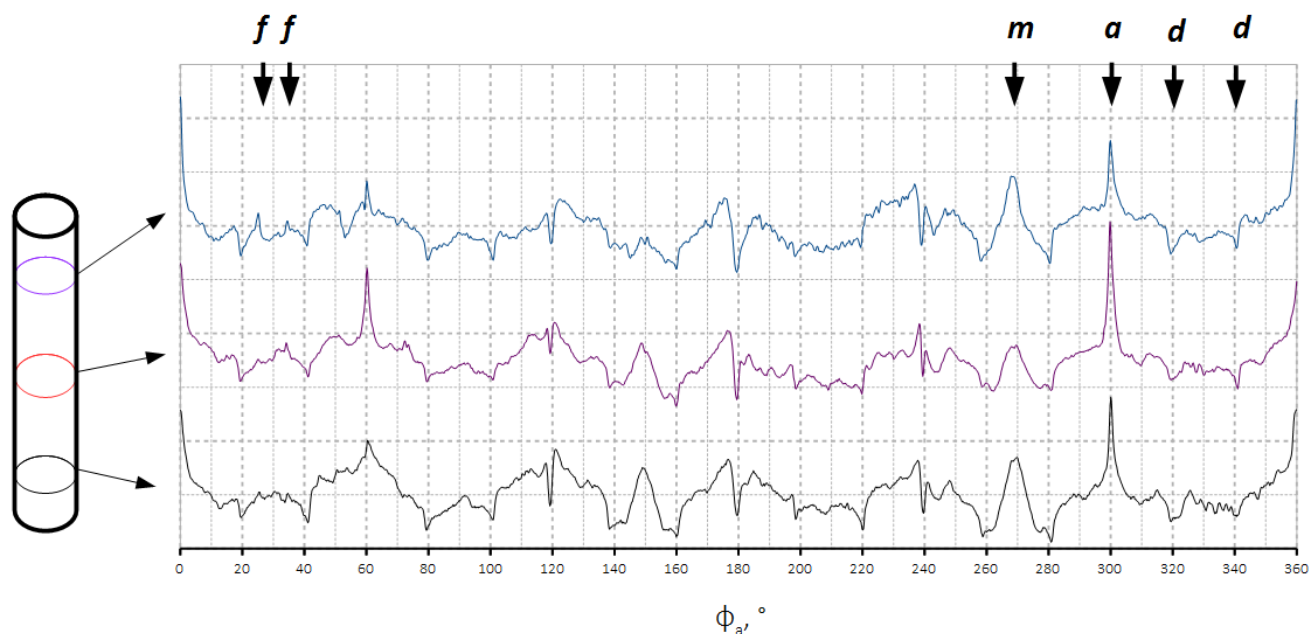


Рис. 24. Сравнение особенностей морфологии боковой поверхности цилиндрического монокристалла сапфира ориентации c на нескольких уровнях.

Графики, приведенные на рисунке 24, хорошо иллюстрируют, что морфология боковой поверхности цилиндрических монокристаллов ориентации c меняется гораздо значительнее, чем морфология боковой поверхности ориентации a , что, скорее всего, свидетельствует о том, что морфология боковой поверхности кристаллов данной ориентации более склонна реагировать на незначительные изменения в параметрах роста. Таким образом можно говорить о том, что кристаллы ориентации c хуже держат сечение, задаваемое формообразователем, чем кристаллы ориентации a .

Рисунки 23 и 24 хорошо иллюстрируют, что наиболее ярко выраженной на кристаллах данной ориентации является грань гексагональной призмы a . У этих граней, также, как и у грани c (рисунок 21), наблюдается плавный переход от грани к окружности, видимо также путем сгущения ступеней, что проявляется на графике интенсивности в виде пьедестала у основания пика.

На месте где должны находиться отражения от граней гексагональной призмы m (через одну позицию положения m , значения ϕ_a на графике (рисунок 23) 30° , 150° и 270°) наблюдаются высокие подъемы с достаточно пологими крыльями и провалом на вершине. Это говорит о том, что грани гексагональной призмы m осложнены наклонными гранями ромбоэдра, выходявшими во время роста на фронт кристаллизации и проявившимися на боковой поверхности. Сама же грань m вырождается в ребро, т.е. становится очень узкой (участком с бесконечно малым радиусом кривизны).

На боковой поверхности монокристаллов сапфира данной ориентации методом видеогониографии выявлено большое количество второстепенных граней дигексагональных призм, превышающее количество граней, выявляемых методом фотогониографии. Как правило, второстепенные грани дигексагональных призм проявляются на графике относительной интенсивности в виде небольших пиков или провалов, слабо отличающихся от уровня шума, что затрудняет их диагностику. Полный перечень граней, выявленных нами в ходе видеогониографического изучения морфологии боковой поверхности монокристаллических стержней сапфира ориентации c приведен в таблице 7.

Таблица 7 – Простые формы, выявленные на боковой поверхности цилиндрических монокристаллов сапфира ориентации c методом видеогониографии

Обозначение	a	-	d	-	f	m
ϕ_c	0	6,6	19,2	21,8	23,4	30

На кристаллах ориентации c выявлено 6 граней из них 3 ранее не описывались на боковой поверхности монокристаллических стержней сапфира. На кристаллах данной ориентации в виде зеркальной грани проявляется только грань гексагональной призмы a .

4.3.3 Ориентация m $[10\bar{1}0]$

На рисунке 25 приведена видеограмма и график относительных интенсивностей отражения от боковой поверхности цилиндрического монокристалла сапфира ориентации m , выраженного способом Степанова.

На графике относительных интенсивностей для ориентации m , аналогично таким же графикам ориентаций a и c , фактически все пики являются значимыми. Но в отличие от кристаллов ориентации a и c морфология боковой поверхности кристаллов данной ориентации очень сильно реагирует даже на незначительные изменения в условиях выращивания, что хорошо иллюстрируют графики, приведенные на рисунке 26.

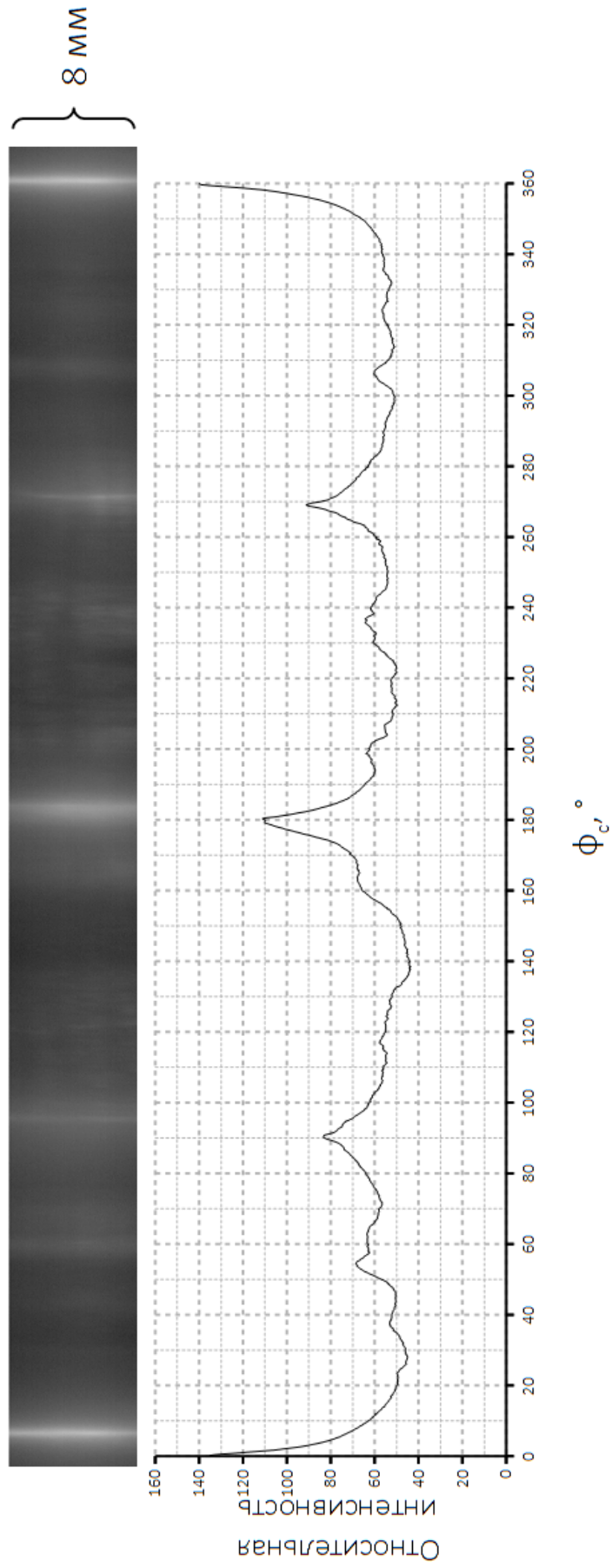


Рис. 25. Видеограмма и ее график относительных интенсивностей для одного из сечений боковой поверхности кристаллов сапфира ориентации *m*.

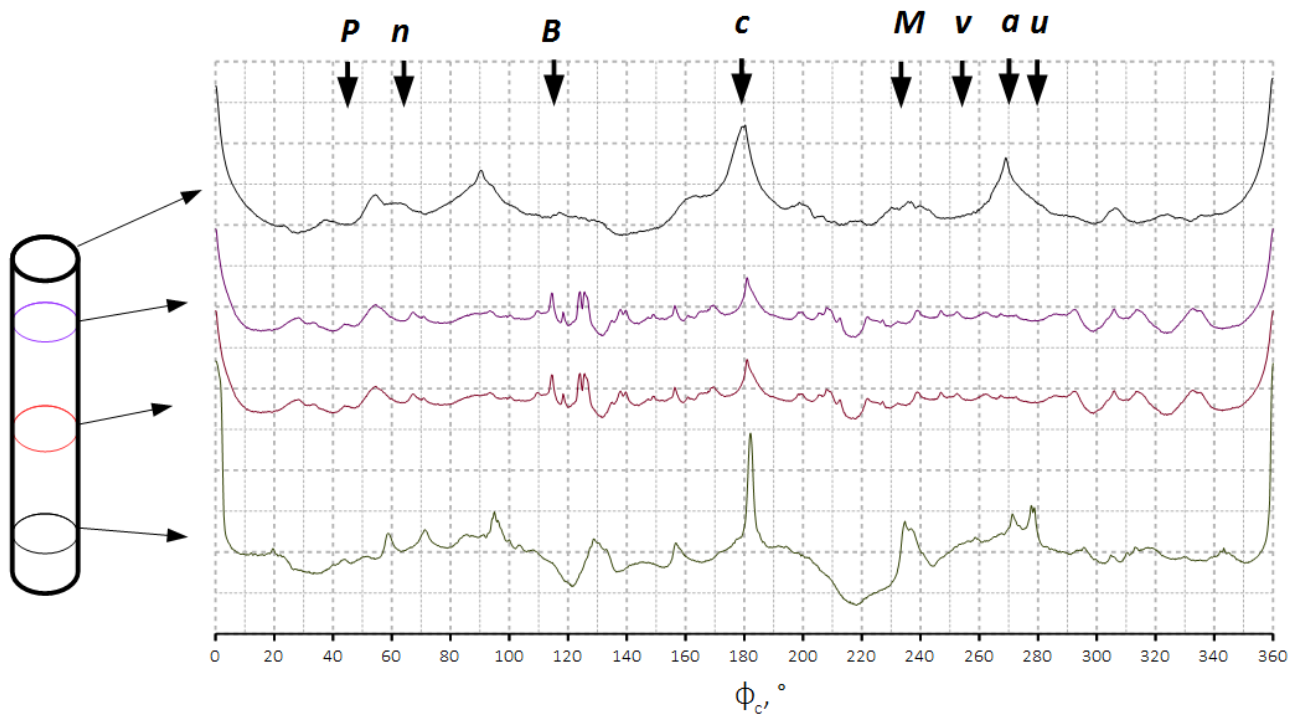


Рис. 26. Сравнение особенностей морфологии боковой поверхности цилиндрического монокристалла сапфира ориентации *m* на нескольких уровнях.

Рисунки 25 и 26 хорошо иллюстрируют, что наиболее ярко выраженной на кристаллах данной ориентации является грань гексагональной призмы *a* и пинакоида *c*, но при этом только грань пинакоида *c* наблюдается на боковой поверхности всех кристаллов и на всех уровнях кристалла. Грань гексагональной призмы *a* постоянно выклинивается. У этих граней, также, как и у грани *c*, на кристалле ориентации *a* $[11\bar{2}0]$ (рисунок 25) наблюдается плавный переход от грани к окружности, видимо также путем сгущения ступеней, что проявляется на графике интенсивности в виде пьедестала у основания пика.

На боковой поверхности монокристаллов сапфира данной ориентации методом видеогониографии выявлено большое количество гексагональных дипирамид, но в отличие от ориентаций *a* и *c*, где все выявленные грани в той или иной степени наблюдаются в одном сечении, на боковой поверхности кристаллов ориентации *m* для составления полного списка проявляющихся граней необходимо было снять большое количество разных по высоте уровней. Например, для приведенного на рисунках 25 и 26 кристалла, необходимо было снять 12 уровней для того что бы выявить все возможные грани для кристаллов этой ориентации.

Как правило, грани гексагональных дипирамид проявляются на графике относительной интенсивности в виде небольших пиков или провалов, слабо отличающихся от уровня шума, что затрудняет их диагностику. Полный перечень граней, выявленных нами в ходе

видеогиографического изучения морфологии боковой поверхности монокристаллических стержней сапфира ориентации m приведен в таблице 8.

Таблица 8 – Простые формы, выявленные на боковой поверхности цилиндрических монокристаллов сапфира ориентации m методом видеогиографии

Обозначение	<i>c</i>	-	-	-	-	-	-	<i>P</i>	<i>M</i>
ϕ_c	0	12,8	14,6	20	24,4	31,2	34,4	42,4	53,8
Обозначение	-	<i>n</i>	<i>B</i>	-	<i>w</i>	<i>v</i>	<i>u</i>	<i>a</i>	
ϕ_c	58,6	61,2	65,4	67,6	69,8	74,6	79,6	90	

На кристаллах ориентации m выявлено 17 граней из них 11 ранее не описывались на боковой поверхности монокристаллических стержней сапфира. На кристаллах данной ориентации в виде зеркальных граней проявляются грани пинакоида c и гексагональной призмы a . Соотношение размеров $c:a$ определить достаточно сложно из-за нестабильности морфологии боковой поверхности монокристаллических стержней, но примерно оно равно 8:1.

4.4 Модель отражения света от боковой поверхности монокристаллических стержней сапфира

4.4.1 Модель отражения от грани с переходными областями

Для анализа возможных форм рефлексов от граней кристалла был выполнен расчёт кривых интенсивностей светового сигнала, отраженного от поверхности цилиндрического стержня с двумя параллельными зеркальными плоскостями при наличии переходных участков и с резким переходом от грани к цилиндру. При этом предполагается, что во всех точках боковой поверхности, как на цилиндрической части, так и на плоских гранях и переходных областях, имеет место зеркальное отражение света. Рассеянное отражение, которое может возникать из-за наличия ступеней на гранях и переходных областях и горизонтальной штриховки на кристалле не учитывается.

За основу приняты размеры грани c на кристалле ориентации a из работы [6]. Радиус кристалла $R = 5$, ширина плоского участка грани $2a = 2,6$ мм, каждой из переходных областей $b = 1,3$ мм. Смещение на длине переходной кривой $v_b = 0,25$ мм. Сечение кристалла в области грани, построенное по экспериментальным данным показано на рисунке 27.

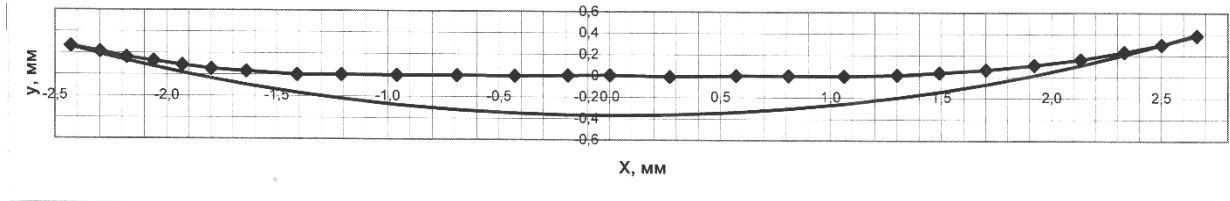


Рис.27. Профиль грани с переходными областями, используемый для модельного расчета [6]

Переходная кривая аппроксимируется кубической параболой, что обеспечивает плавное нарастание кривизны от 0 до $\frac{1}{R}$. Схема съёмки показана на рисунке 28. Датчик находится в точке $\left(x_0 = \frac{R\sqrt{2}}{2}, y_0 = 12\right)$. Свет падает справа, параллельно оси x . За исходное принимается такое положение, когда на датчик попадает луч, отраженный от цилиндрической поверхности, а грань перпендикулярна оси x . Теоретически грань войдёт в отражающее положение, когда кристалл повернётся на угол $\varphi_0=45^\circ$.

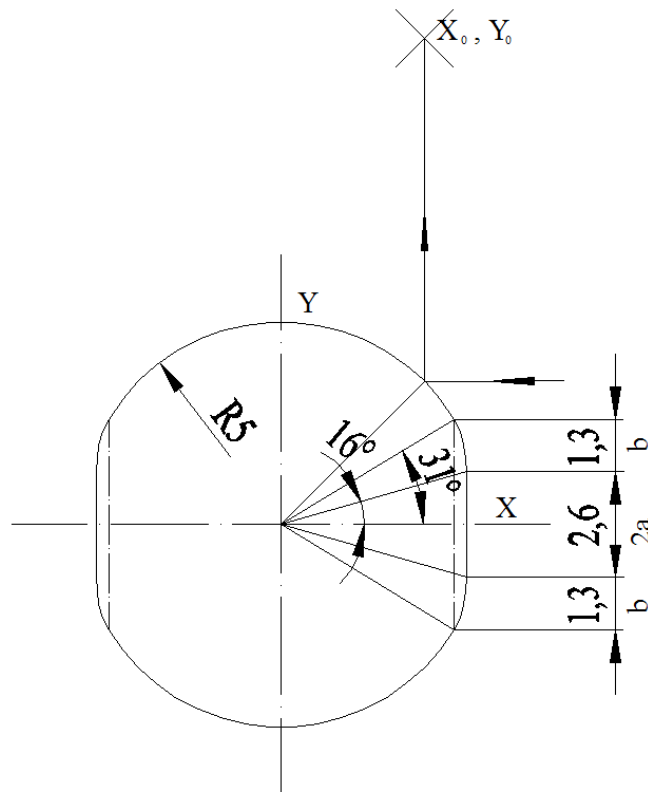


Рис. 28. Схема съёмки кристалла с переходной областью

Расчёт видеограммы кристалла с широкой гранью и переходными областями, построенной по этим данным, приведен в Приложении 1.1. Результат расчета показан на рисунке 29. За единицу принят уровень сигнала, возникающего при отражении от плоского

зеркального участка (границы) в случае нормального падения луча на плоскость датчика. Интенсивность отражения от криволинейной части определяется расхождением светового потока, отраженного от выпуклого цилиндрического зеркала. Она убывает обратно пропорционально расстоянию точки регистрации от мнимого фокуса. Этим и определяется убывание интенсивности на переходных участках и её постоянный уровень при отражении от кругового цилиндра вне областей границы.

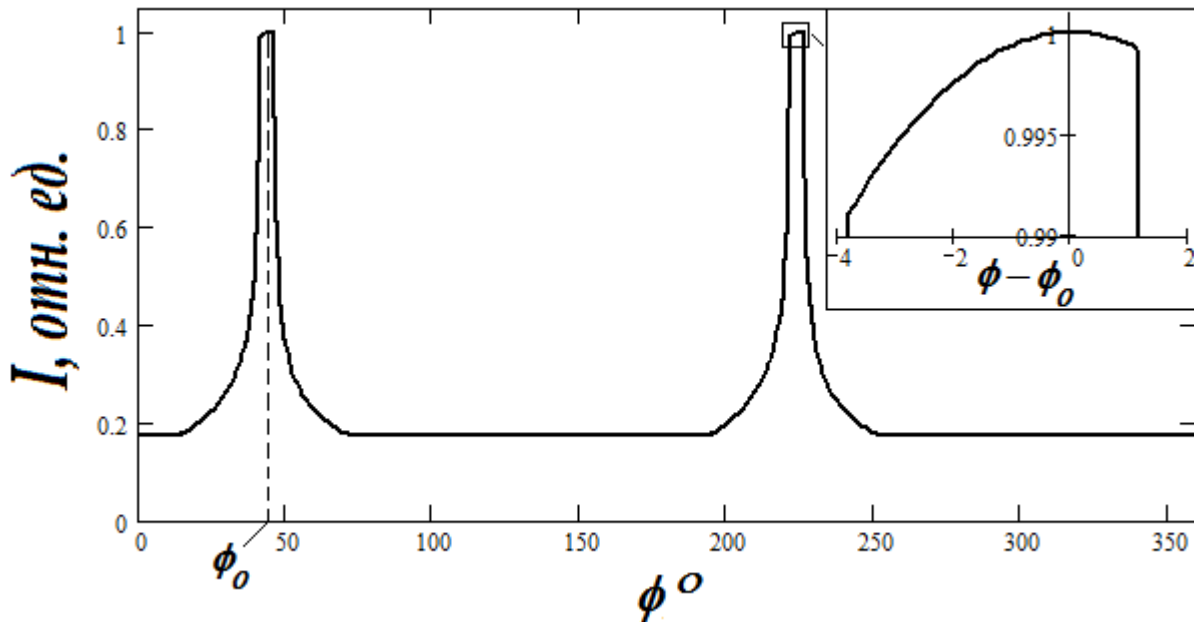


Рис. 29. Модельный график относительных интенсивностей, построенный для стержня с двумя параллельными гранями, имеющими переходные области

В результате расчета получилось, что:

- В данной геометрии грань шириной 2,6 мм даёт пик шириной 5° .
- Переходные области с плавным нарастанием кривизны дают расширение пика в начале с резким спадом, а в нижней части плавно до 62° .
- Пик несимметричен; левое и правое крылья смещены относительно теоретического положения грани $\varphi_0=45^\circ$ на $\sim 1,5^\circ$.
- Вершина пика уплощённая, но не плоская, что связано с небольшим изменением угла падения лучей, отражённых от плоской грани на датчик. При этом максимум интенсивности в пределах вершины точно соответствует его теоретическому положению.

Рассчитанная форма пика хорошо согласуется с формой рефлексов от грани *c* на кристаллах ориентаций *a* и *m*.

4.4.2 Моделирование отражения от грани без переходных областей

При той же ширине грани 2,6 мм исключены переходные области (рисунок 30а). На границе грани и цилиндра возникает ребро. В этой точке две касательных (как и две нормали), а кривизна изменяется скачком.

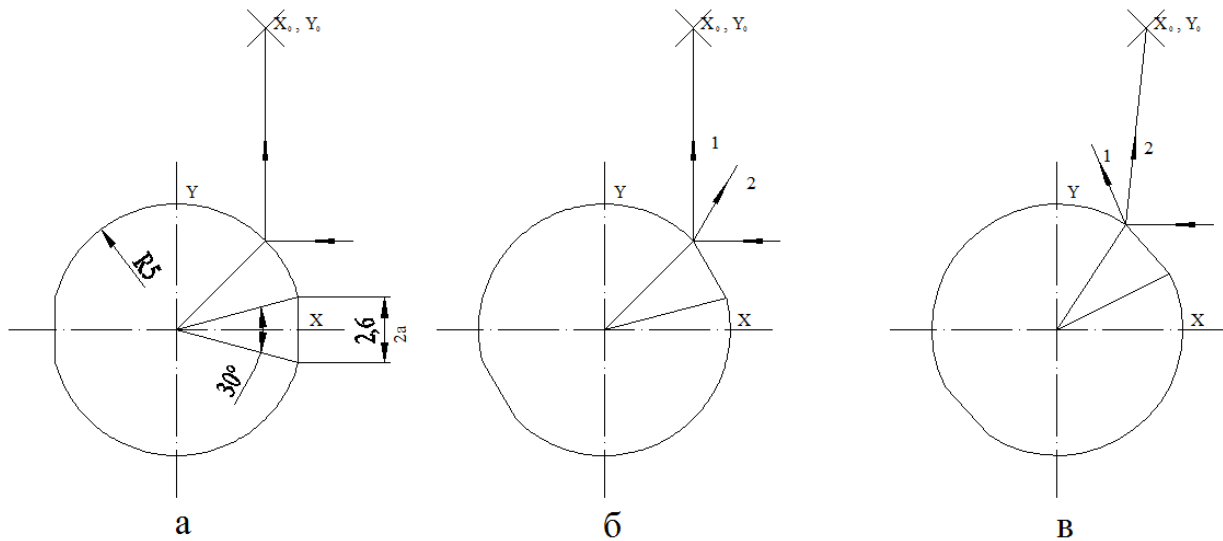


Рис. 30. Схема съемки кристалла с широкой гранью без переходной области

При повороте кристалла отражение от цилиндрической части будет иметь место до того момента, когда в отражающее положение не войдёт точка на ребре (рисунок 30б, луч 1). Затем для некоторого диапазона углов поворота сигнала на датчике не будет, пока на него не попадёт луч 2, отражённый в той же точке, но уже от плоской грани (рисунок 30в). Далее датчик будет фиксировать отражение от точек грани, пока в отражающее положение не войдёт второе ребро. Здесь вновь будет интервал углов без сигнала датчика и далее – отражение от цилиндрической поверхности.

Расчёт видеограммы кристалла с широкими гранями без переходных областей приведен в приложении 1.2. Результат расчета показан на рисунке 31.

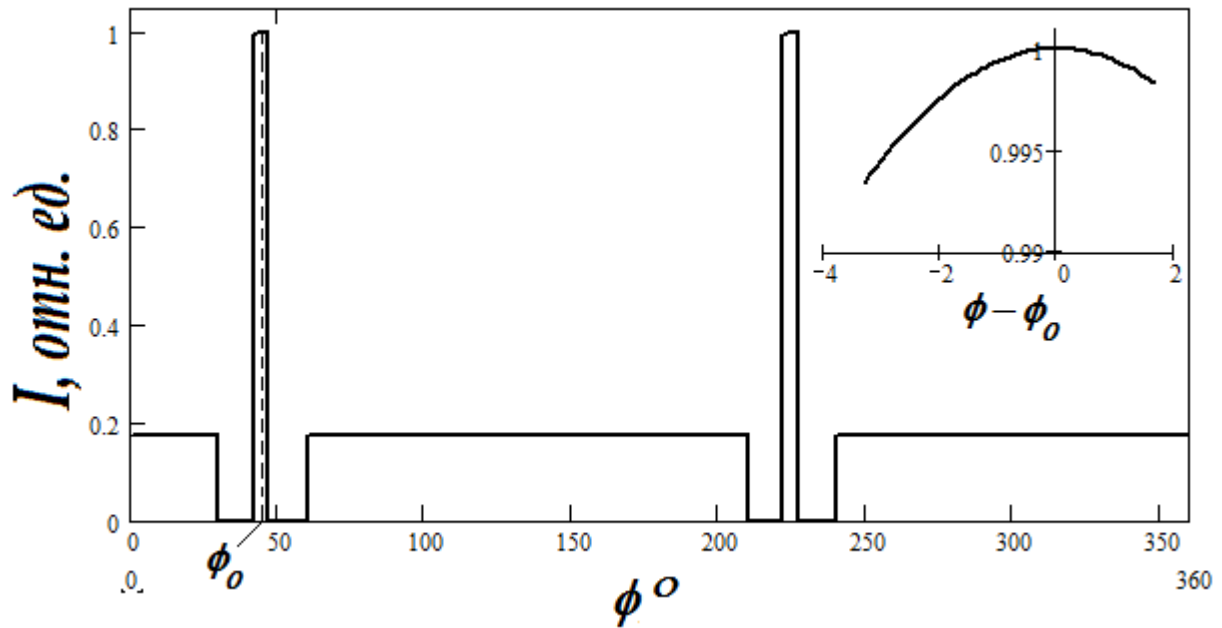


Рис. 31. Модельный график относительных интенсивностей, построенный для стержня с двумя параллельными гранями, без переходных областей

В результате расчета получилось, что:

- В данной геометрии грань шириной 2,6 мм даёт пик немного меньшей ширины с крутыми спадами.
- Из-за наличия рёбер рядом с пиком возникают нулевые области шириной 12° каждая.
- Как и в варианте грани с переходными областями, крылья пика смещены относительно теоретического положения грани $\varphi_0=45^\circ$ на $\sim 1,5^\circ$.
- Форма вершины пика аналогична варианту грани с переходными областями. Максимум интенсивности в пределах вершины, как и ранее точно соответствует его теоретическому положению.

Такая форма пика хорошо коррелирует с рефlekсами от грани r на кристаллах ориентации a .

4.5 Элементы морфологии боковой поверхности кристаллов сапфира

На боковой поверхности монокристаллических стержней сапфира, выращенных способом Степанова, нами выделено 3 вида отклонений от заданного формообразователем диаметра:

1. Зеркальная грань – будет проявляться на графике интенсивности в виде высокого узкого пика. При большой ширине этот пик имеет закругленную вершину (рисунок 32).

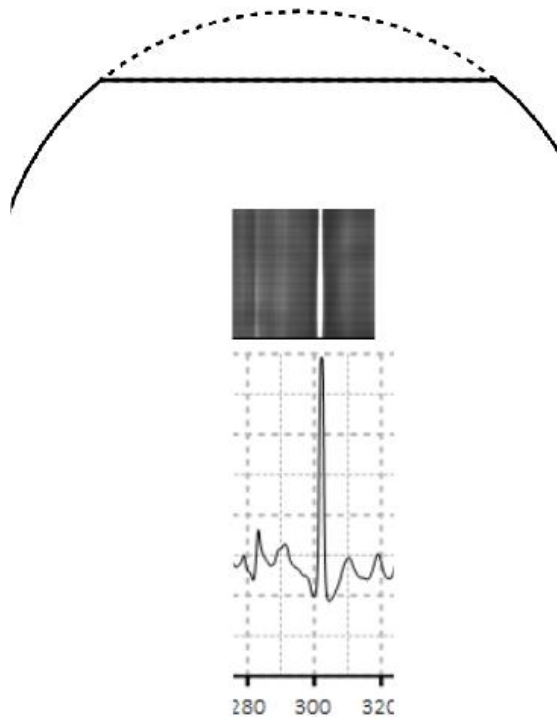


Рис. 32. Плоская грань на боковой поверхности цилиндрического кристалла и ее проявление на видеogramме и графике относительной интенсивности.

В связи с тем, что в конструкции видеогониографа используется не нормальное отражение, вершина пика получается ассиметричной, что подтверждается результатами моделирования отражения. Это явление наблюдалось и ранее на фотограммах и, как указано в [3], приводило к ошибке в определении точного углового положения грани. В видеогониографе этот недостаток устранён. Пологая вершина высокого пика отражения от широкой грани ассиметрична, но её максимум соответствует положению грани с точностью до разрешения видеogramмы. В случае кристаллов сапфира в силу их кристаллографии все плоские грани попарно параллельны. Если кристалл рос без нарушения ориентации, то интенсивность пиков от параллельных плоских граней должна быть одинаковой, в ином случае интенсивность может отличаться, причём большую интенсивность будет давать та грань, которая была наклонена к

расплаву. Это хорошо видно на кристаллах ориентации *a* и *c* по разным значениям интенсивностей граней (рисунки 21 и 23).

2. Ребро – проявляется на видеодиаграмме в виде тёмной полосы, а на графике интенсивности - в виде острых узких провалов, так как ребро – это участок с бесконечно малым радиусом кривизны (см. главу 4.4.2). Ребра на наших кристаллах могут возникать в случае отклонение ориентации направления выращивания кристалла, когда грань рационального направления несколько наклонена в сторону от расплава. В этом случае она выклинивается (рисунок 33).

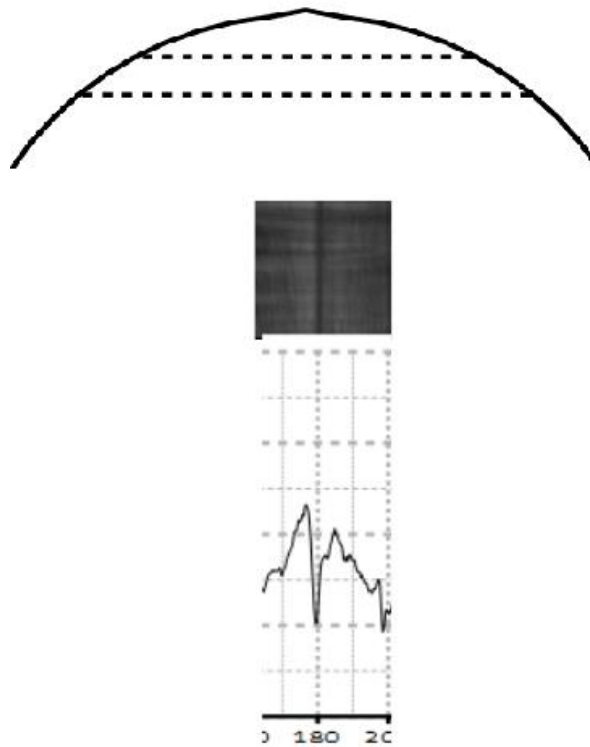


Рис. 33. Ребро на боковой поверхности цилиндрического кристалла и его проявление на видеодиаграмме и графике относительной интенсивности.

Наиболее явно ребра видны на рисунке 23, где им соответствуют рефлекссы, находящиеся на отметках 60° , 120° , 180° , и 240° – призма *a*; 150° , 270° – призма *m*.

3. Уплотнения (участки с радиусом кривизны большим радиуса кристалла, задаваемого формообразователем) – такие участки на графике интенсивности будут проявляться в виде невысоких подъемов, не имеющих четкого максимума, с пологими крыльями (иногда осложнёнными другими формами – гранями или ребрами). Уплотнение проиллюстрировано на рисунке 34.

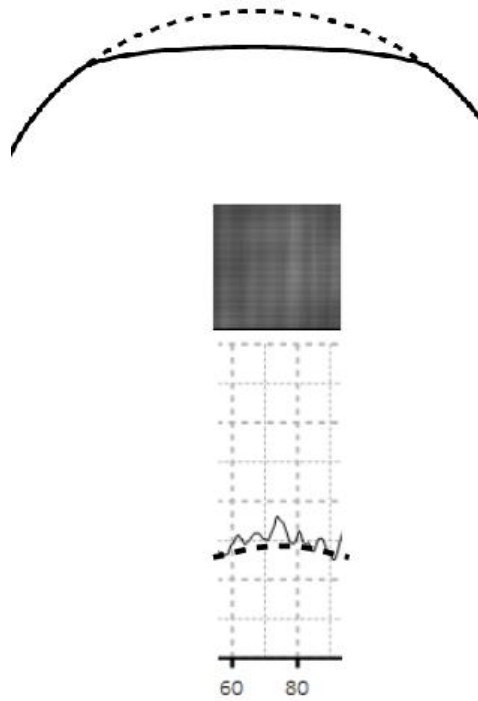


Рис. 34. Уплотнение на боковой поверхности цилиндрического кристалла и его проявление на видеограмме и графике относительной интенсивности.

Характерным примером является интервал $144^\circ - 156^\circ$ на рисунке 23. В этом месте на фронте кристаллизации существовала наклонная зеркальная грань r , проявившаяся на боковой поверхности в виде видимой глазом ступенчатой полосы, совпадающей по положению со слабо проявляющимися гранями призмы m . Ширина этих подъёмов $\sim 10^\circ$ соответствует ширине полос, наблюдаемых на кристалле.

4.6 Результаты и выводы

Создан новый оптический прибор с цифровой регистрацией, который позволил наиболее полно описать морфологию боковой поверхности цилиндрических монокристаллов сапфира на количественном уровне.

По результатам видеогониографического исследования морфологии боковой поверхности цилиндрических монокристаллов сапфира различных ориентаций, выращенных способом Степанова можно сделать следующие заключения:

1. На боковой поверхности цилиндрических кристаллов сапфира было выявлено следующее количество граней:
 - На кристаллах ориентации a $[11\bar{2}0]$ – 31 грань;
 - На кристаллах ориентации c $[0001]$ – 6 граней;
 - На кристаллах ориентации m $[10\bar{1}0]$ – 17 граней.
2. На боковой поверхности цилиндрических кристаллов сапфира выявлены три морфологические единицы, влияющие на конечную форму растущего кристалла: плоская грань, ребро, уплощение.
3. Установлено, что наиболее стабильны во время роста кристаллы ориентации a и c , кристаллы ориентации m сильно чувствительны даже к незначительным изменениям условий роста, что необходимо учитывать во время планирования технологического процесса;
4. В виде зеркальных граней на боковой поверхности цилиндрических монокристаллов проявляются грани пинакоида c , положительного ромбоэдра r и гексагональной призмы s . Эти грани проявляются на фотограммах в виде светлой полосы, а на графиках интенсивности в виде высокого максимума.
5. Установлена следующая последовательность граней на боковой поверхности цилиндрических монокристаллов по их ширине – $c > r > a >$ (все остальные). Грани, не проявляющиеся в виде зеркальных участков боковой поверхности, представляют собой уплощения, ширину которых невозможно измерить, но очевидно, что она меньше, чем у трёх зеркальных граней. На графиках интенсивности им соответствуют невысокие острые максимумы без уплощённой вершины, а на видеограмме – слабые светлые полосы.
6. При точной ориентации затравки все грани должны были бы отображаться светлыми полосами, но из-за наличия остаточных разориентаций условия развития граней на противоположных сторонах кристалла оказываются неэквивалентными. Грани, наклонённые к расплаву, проявляются в виде более широких полос, а наклонённые от

расплава оказываются уже, а высота их пиков - меньше. При этом слабые грани часто превращаются в рёбра, давая тёмные полосы на видеограммах и провалы на графиках интенсивности.

7. Видеогониограф полностью подтвердил полученные ранее на фотогониографе качественные результаты, позволил дать им относительную количественную оценку, а также выявил большое количество неустановленных ранее элементов морфологии боковой поверхности кристаллов.

5 РАСЧЕТ ПОВЕРХНОСТНЫХ ЭНЕРГИЙ КРИСТАЛЛОВ САПФИРА

5.1 Расчет структуры сапфира известными методами [9,10]

5.1.1 Расчет ретикулярной плотности

Как уже было сказано в главе 3 – расчета ретикулярной плотности для решетки сапфира в литературе не приводится. Мы выполнили такой расчет. Его результаты приведены на диаграмме (рисунок 35).

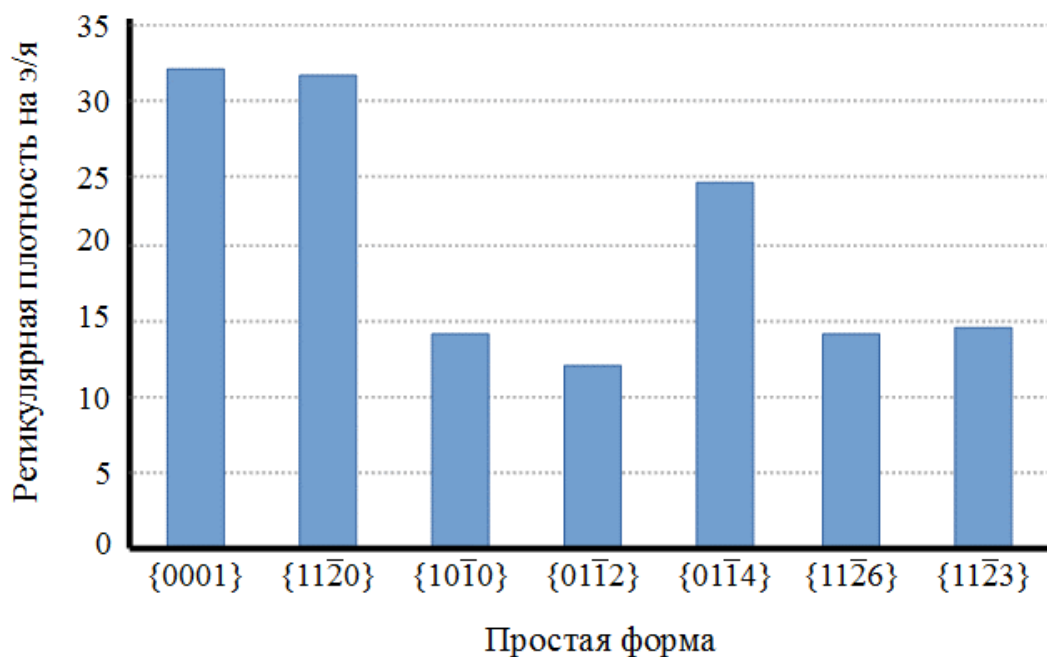


Рис. 35 Ретикулярные плотности для главнейших плоскостей корунда

Напомним, что кристалл по закону Браве ограничивается гранями с наибольшей ретикулярной плотностью. Диаграмма (рисунок 35) хорошо иллюстрирует, что последовательность граней по убыванию значения ретикулярной плотности следующая: $c - a - R - P - n - m - r$. Отсюда следует, что и последовательность граней по размерам должна быть такой же.

5.1.2 Расчет методом Ненасыщенности связей

Как уже упоминалось в главе 3, метод Ненасыщенности связей был предложен М.Д. Любалиным [20], как продолжение метода Периодических цепей связей (ПЦС) – его количественная характеристика. Данный метод был успешно применен М.Д. Любалиным к

кубическим и тетрагональным кристаллам относительно простой структуры [20]. К сожалению, решетка сапфира в [20] не рассматривалась.

Нами был произведен расчет решетки сапфира, построенной по рентгеновским данным (Глава 1), методом Ненасыщенности связей. Были построены проекции решетки сапфира на три основные плоскости, выделены слои d_{hkl} элементарные пакеты (Δ_{hkl}) и рассчитан параметр ненасыщенности связей для наиболее часто встречающихся плоскостей сапфира: пинакоид $c\{0001\}$, гексагональная призма $a\{11\bar{2}0\}$, гексагональная призма $m\{10\bar{1}0\}$ (рисунок 36), ромбоэдры $+q\{10\bar{1}5\}$ и $-q\{\bar{1}015\}$, ромбоэдры $+R\{10\bar{1}4\}$ и $-R\{\bar{1}014\}$, ромбоэдры $+r\{10\bar{1}2\}$ и $-r\{\bar{1}012\}$, ромбоэдры $+S\{10\bar{1}1\}$ и $-S\{\bar{1}011\}$, ромбоэдры $+t\{20\bar{2}1\}$ и $-t\{\bar{2}021\}$ (рисунок 37), гексагональные дипирамиды $P\{11\bar{2}6\}$, $n\{11\bar{2}3\}$, $w\{11\bar{2}2\}$ и $u\{11\bar{2}1\}$ (рисунок 38)

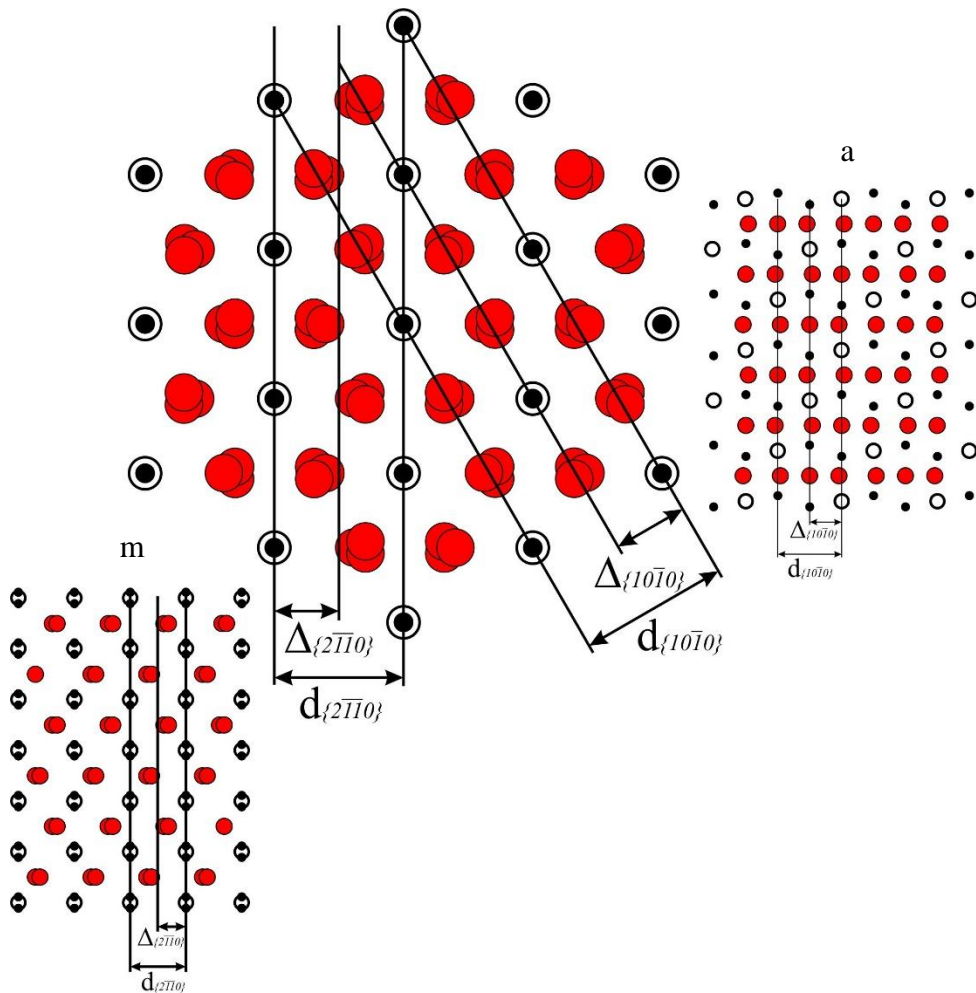
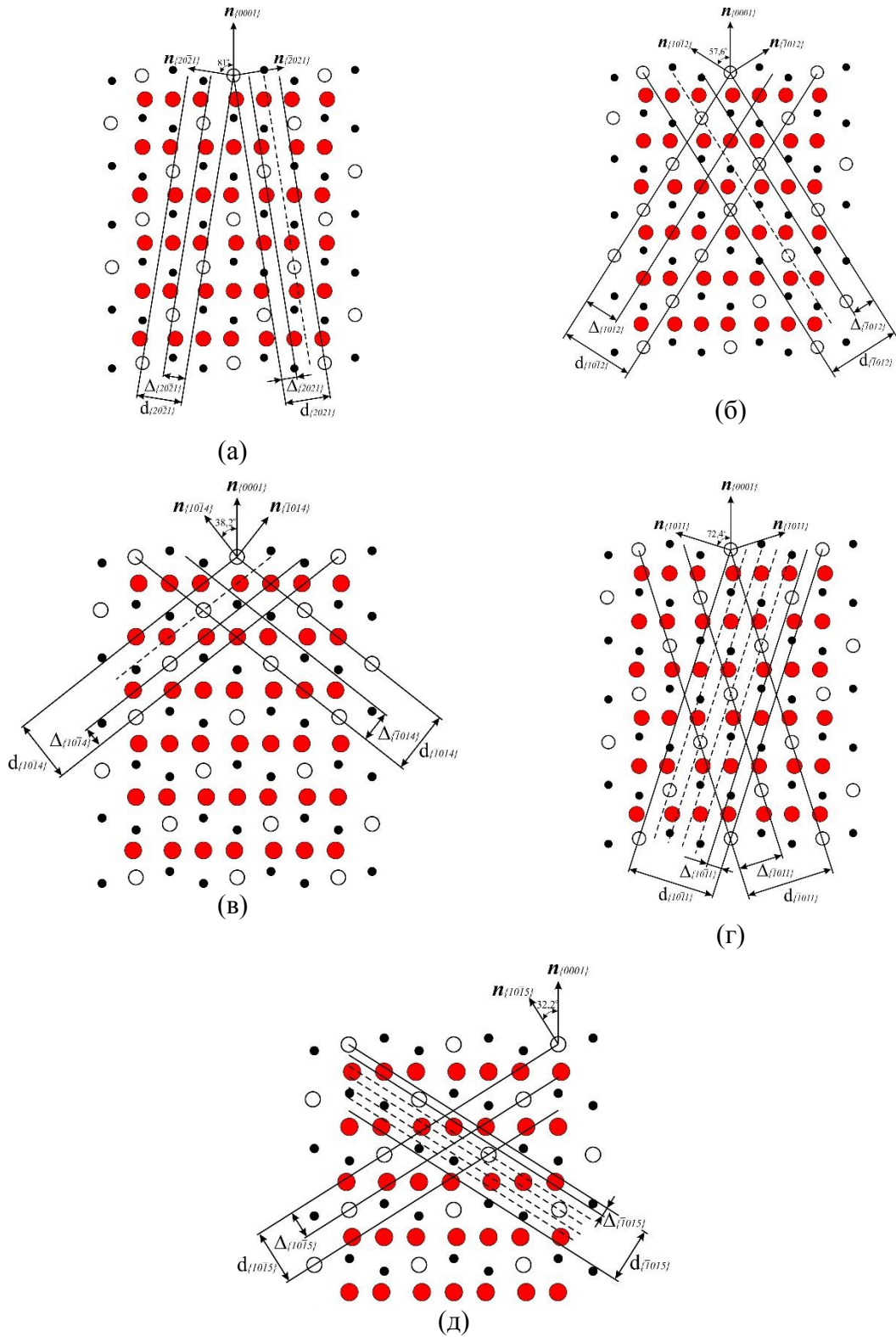


Рис. 36. Проекция на плоскость $\{0001\}$



- (a) $+t \{20\bar{2}1\}, -t \{\bar{2}021\}$
 (б) $+r \{10\bar{1}2\}, -r \{\bar{1}012\}$
 (в) $+R \{10\bar{1}4\}, -R \{\bar{1}014\}$
 (г) $+S \{10\bar{1}1\}, -S \{\bar{1}011\}$
 (д) $+q \{10\bar{1}5\}, -q \{\bar{1}015\}$

Рис. 37. Проекция на плоскость $\{11\bar{2}0\}$

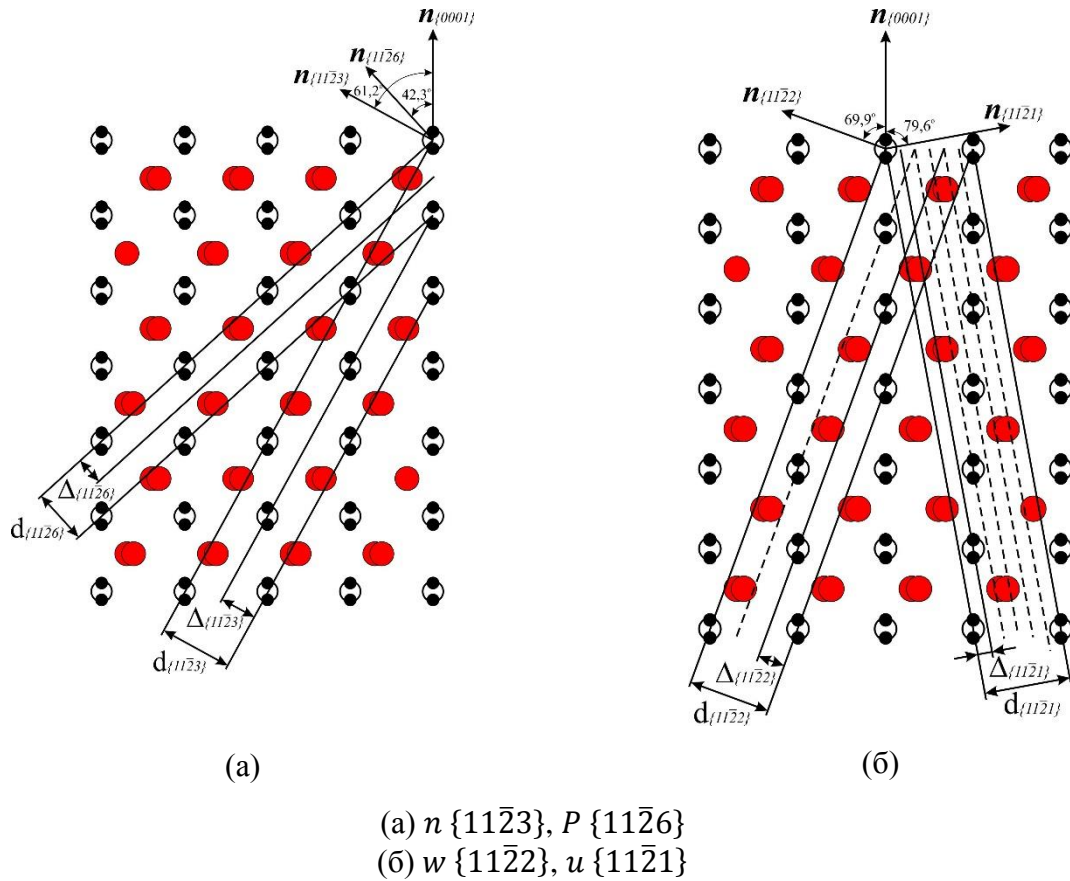


Рис. 38. Проекция на плоскость $\{10\bar{1}0\}$.

Для определения ненасыщенности связей нами были рассчитаны её нормальная и тангенциальная составляющие в соответствии с методикой, предложенной М.Д. Любалиным, для каждой плоскости по формулам:

$$H_{hkl\perp} = \frac{n_i * m_i * \cos\alpha * E}{2^7 * \Delta_{hkl}} * \left(\frac{r_i + r_j}{r_{i,j}} \right)^6, \quad (4)$$

$$H_{hkl\parallel} = \frac{n_i * m_i * \sin\alpha * E}{2^7 * \Delta_{hkl}} * \left(\frac{r_i + r_j}{r_{i,j}} \right)^6, \quad (5)$$

где n_i – количество атомов; m_i – количество связей; Δ_{hkl} – толщина элементарного пакета; α – угол наклона связи; E – энергия единичной связи; r_i – радиус атома i ; r_j – радиус атома j .

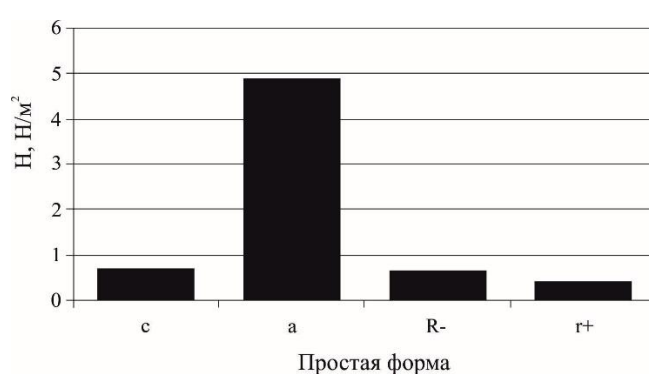
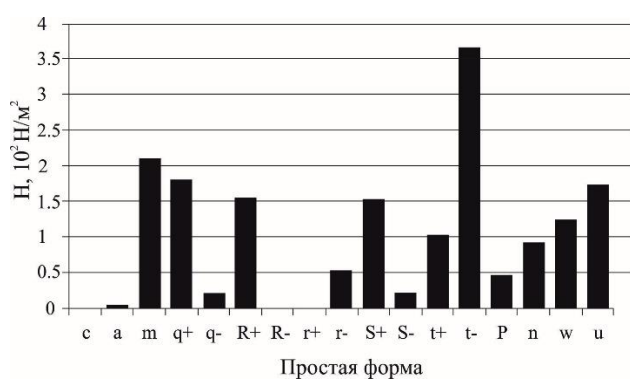
Далее, в соответствии с этой методикой, рассчитывался параметр ненасыщенности связей следующим образом:

$$H_{hkl} = \sqrt{H_{hkl\perp}^2 + H_{hkl\parallel}^2} \quad (6)$$

Результаты расчетов методом Ненасыщенности связей для решетки сапфира приведены в таблице 9, где помимо значений ненасыщенности связей, приведены параметры d_{hkl} и Δ_{hkl} для каждого направления. Результаты расчетов проиллюстрированы на диаграммах (рисунок 39).

Таблица 9 – Результаты расчетов ненасыщенности связей для идеальной решетки сапфира

Плоскость	$d_{hkl}, \text{Å}$	$\Delta_{hkl}, \text{Å}$	$H_{hkl}, \text{Н/М}^2$
$c \{0001\}$	12,97	2,161667	0,71
$a \{11\bar{2}0\}$	2,375	2,375	4,88
$m \{10\bar{1}0\}$	4,1136	1,3712	212,75
$+q \{10\bar{1}5\}$	2,1942	1,0971	179,98
$-q \{\bar{1}015\}$	2,1942	2,1942	222,78
$+R \{10\bar{1}4\}$	2,5465	0,848833	154,74
$-R \{\bar{1}014\}$	2,5465	2,5465	0,66
$+r \{10\bar{1}2\}$	3,4737	3,4737	0,42
$-r \{\bar{1}012\}$	3,4737	1,1579	53,30
$+S \{10\bar{1}1\}$	3,9211	0,653517	153,52
$-S \{\bar{1}011\}$	3,9211	1,96055	22,21
$+t \{20\bar{2}1\}$	2,0314	1,0157	104,37
$-t \{\bar{2}021\}$	2,0314	0,338567	366,46
$P \{11\bar{2}6\}$	1,5986	1,5986	47,90
$n \{11\bar{2}3\}$	2,0816	2,0816	92,30
$w \{11\bar{2}2\}$	2,2301	0,743367	124,64
$u \{11\bar{2}1\}$	2,3362	0,778733	174,58



(а) – для всех рассчитанных граней,
 (б) – для граней с минимальным значением ненасыщенности связей

Рис. 39. Диаграммы, иллюстрирующие результаты расчета ненасыщенности связей для идеальной структуры сапфира

Построенные по результатам расчетов графики (рисунок 39) хорошо иллюстрируют, что наименьшим параметром ненасыщенности связей обладают плоскости c , a , $-R$, $+r$ и имеют следующую последовательность по возрастанию указанного параметра: $+r - R - c - a$. Все остальные плоскости имеют значение ненасыщенности связей существенно большее.

5.2 Результаты расчетов по существующим методам и экспериментальные данные

Как было экспериментально установлено в главе 4, главная последовательность граней по ширине $c > r > a$. В соответствии с принципом минимума поверхностной энергии, который был сформулирован Вульфом, размер грани на кристалле обратно пропорционален значению поверхностной энергии этой грани, т.е. чем меньше энергия, тем больше размер грани. [21]. Отсюда следует, что поверхностные энергии, установленной нами последовательности граней, должны выстраиваться в обратном порядке, что будет выглядеть как $c < r < a$. Для того чтобы избежать путаницы, в дальнейшем мы будем пользоваться следующей записью: $c - r - a$. Данная последовательность главных граней принимается нами за экспериментальный результат, который должен подтвердить достоверность метода расчета.

В таблице 10 приведены последовательности граней по результатам расчетов различными известными методами.

Таблица 10 – Сравнение экспериментально полученных данных по ограничению боковой поверхности кристаллов сапфира с результатами расчетов существующими методами

Ссылка	Наименование метода расчета	Последовательность граней	Примечание
Наш расчёт	Ретикулярная плотность	$c - a - R - P - n - m - r$	-
[43]	По Доннею-Харкеру	$a - R - r - c - n - S - P$	-
[43]	ПЦС	$a - c - S - R - r - n - q$	На ячейку
[43]		$a - r - R - n - q - S - c$	На единицу поверхности
Наш расчёт	Ненасыщенности связей	$+r - -R - c - a - \dots$	Приведены только плоскости с наименьшими значениями ненасыщенности
[41]	METADISE	$R - c - a - n - m - r - P$	-
[33]	Квантовые методы расчетов	$r - a - m - c - R$	-
[34]		$c - m - R - a - r$	Расчёт из первых принципов
[37]		$m - c - a - r - R$	Расчёт из первых принципов
[37]		$c - m - r - a - R$	Модель парного взаимодействия
[38]		$c - m - R - a - r - p$	Используется потенциал Макродта
[38]		$R - c - a - m - p - r$	Используется эмпирический потенциал
[38]		$R - a - c - p - r - m$	Используется потенциал «Ab initio»
[40]		$R - c - a - r$	-
Результаты нашего эксперимента	Результаты эксперимента по изучению ограничения боковой поверхности профилированных кристаллов сапфира (главная последовательность)	$c - r - a$	-

Из таблицы видно, что ни один из существующих методов расчётов ограничения кристаллов сапфира не дает последовательности, удовлетворяющей результатам экспериментов. В связи с этим нами предлагается альтернативный подход к расчету поверхностных энергий, речь о котором пойдет в следующем разделе.

5.3 Метод разрыва связей

К настоящему времени достаточно полно разработан аппарат описания кристаллических структур, в которых для упрощения структурные частицы считают точками, совпадающими со средним во времени положением атомных ядер. Дефекты реального кристалла во внимание не принимаются, и кристалл мыслится занимающим все пространство. Такой подход позволяет ограничить анализ структуры в целом рассмотрением элементарной ячейки или блока элементарных ячеек. Элементарная ячейка представляет собой минимальную часть структуры, полностью отвечающую ее симметрии. Вся структура получается из элементарной ячейки действием групп чистых трансляций, относящихся к 14 известным типам (решеткам Бравэ).

Кристаллические грани легко представить в виде плоских сечений структур. Геометрически возможных поверхностей в кристалле бесконечное множество. Однако устойчивы только некоторые из них, что обусловлено требованием минимума свободной поверхностной энергии. На границе кристалл-вакуум поверхностную энергию при абсолютном нуле температуры принято считать приближенно равной полусумме энергий всех межатомных связей, которые надо разорвать для образования рассматриваемой поверхности.

Для отбора наиболее устойчивых поверхностей в структуре разрывают самые слабые (обычно самые длинные) связи. Это современное правило отбора, в котором геометрический и физический аспекты вопроса смыкаются на уровне «длина связи - энергия связи», близко к старинной трактовке О. Бравэ: «Возможность появления и развития каждой рациональной грани кристалла должна быть, по крайней мере, отчасти пропорциональна плотности ее сетки» [52]. Некоторые винтовые оси и плоскости скользящего отражения могут резко изменить ретикулярную плотность граней [29]. Для количественной характеристики некубических кристаллов эти элементы бесконечной симметрии учесть трудно [20] поэтому некубические структуры рассмотрены в литературе мало.

5.3.1 Принципы метода [53]

В первом приближении будем считать, что разрыв связи алюминий-кислород (Al-O) в вакууме приводит к возникновению на поверхностях двух ненасыщенных связей: на атоме

алюминия с энергией, пропорциональной квадрату радиуса атома алюминия, и на атоме кислорода с энергией, пропорциональной квадрату радиуса атома кислорода.

$$\frac{E_{Al}}{E_O} = \left(\frac{r_O}{r_{Al}}\right)^2 \quad (7)$$

При этом выполняется закон сохранения энергии и энергия единичной связи (E) равняется сумме энергий ненасыщенных связей на атомах алюминия и кислорода:

$$E = E_{Al} + E_O \quad (8)$$

Для монокристаллов сапфира в качестве средней энергии связи нами берется табличное значение энергии атомизации в вакууме, которое равняется 731 ккал/моль [7].

При проведении аналогичных расчётов для других кристаллов, при отсутствии значения энергии атомизации, для оценки относительных значений поверхностной энергии возможно принять среднюю энергию связи равной 1.

Решая систему уравнений из (7) и (8) получим энергию разорванной связи на атомах алюминия и кислорода

$$E_O = \frac{E \cdot r_{Al}^2}{r_O^2 + r_{Al}^2} \quad (9)$$

$$E_{Al} = \frac{E \cdot r_O^2}{r_O^2 + r_{Al}^2} \quad (10)$$

Для проверки подставив (9) и (10) в (8) получим:

$$E = \frac{E \cdot r_{Al}^n}{r_O^n + r_{Al}^n} + \frac{E \cdot r_O^n}{r_O^n + r_{Al}^n} = \frac{E \cdot (r_{Al}^n + r_O^n)}{r_O^n + r_{Al}^n} \Rightarrow \\ \Rightarrow E = E \text{ ч. т. д.} \quad (11)$$

В случае если необходимо оценить значение поверхностной энергии граней, на которых обрываются связи, сильно различающиеся по длине, значение энергии связи следует считать обратно пропорциональными квадратам радиусов атомов. Мы будем использовать такой подход для сапфира, у которого разница в длине связей составляет более 5 %.

Для вычисления поверхностной энергии необходимо построить модель участка кристаллической решетки не менее 1 элементарной ячейки, провести плоскость сечения и определить количество связей, идущих от атома в сторону отрыва.

Таким образом, формула для расчета поверхностной энергии кристаллов сапфира будет выглядеть:

$$E_{\text{пов.}} = \frac{2E}{S \cdot \left(1 + \left(\frac{L_l}{L_s}\right)^2\right) \cdot (r_O^2 + r_{Al}^2)} \left[r_{Al}^2 \left(n_{O_l} + \left(\frac{L_l}{L_s}\right)^2 n_{O_s} \right) + r_O^2 \left(n_{Al_l} + \left(\frac{L_l}{L_s}\right)^2 n_{Al_s} \right) \right] \quad (12)$$

где E – энергия единичной связи, S – площадь сечения, L_l – длина длинной связи, L_s – длина короткой связи, r_O – радиус атома кислорода, r_{Al} – радиус атома алюминия, n_{O_l} – количество атомов кислорода с длинной связью, n_{Al_l} – количество атомов алюминия с длинной связью, n_{O_s} – количество атомов кислорода с короткой связью, n_{Al_s} – количество атомов алюминия с короткой связью.

5.3.2 Модель и анализ решетки сапфира

Компьютерная модель идеальной структуры сапфира строилась по опубликованным рентгеновским данным с учетом среднего во времени положением атомов, длины и направленности связей, а также постоянных решетки (Глава 1). Проекция на главные плоскости решетки были приведены на рисунке 5.

Для начального расчета нами были выбраны те же плоскости, что и для расчета методом ненасыщенности связей – наиболее часто встречающиеся в литературе плоскости сапфира.

В связи с тем, что среди выбранных плоскостей есть плоскости с очень крутыми углами, которые целиком не помещаются в одну ячейку, была построена модель идеальной структуры сапфира размером 30x30x30 элементарных ячеек в направлениях a , m и c (в дальнейшем – блок). Из блока вырезался параллелепипед со стороной, соответствующей 30 постоянным ячейки по трём осям декартовой системы координат. Схематический пример такого вырезания для меньшего блока (высотой в одну элементарную ячейку) приведен на рисунке 40.

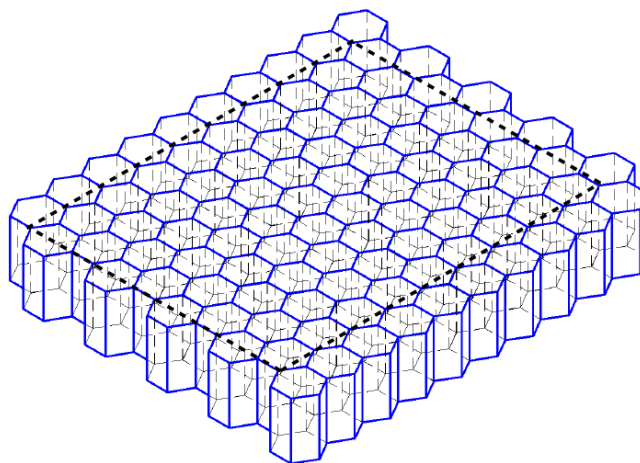
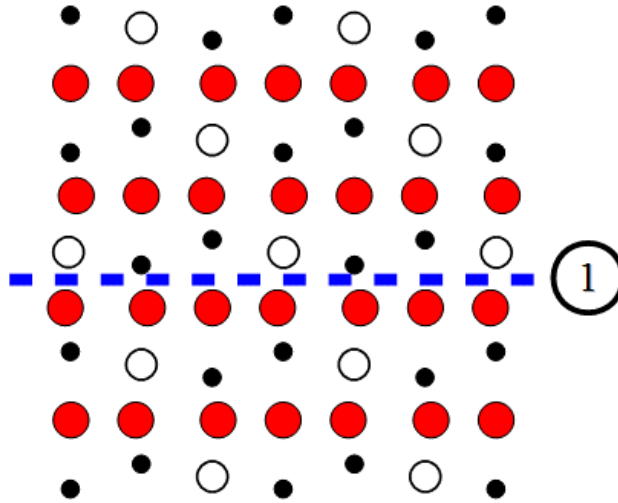


Рис. 40. Схематическое изображение обрезаемого блока ячеек сапфира

Далее вводится секущая плоскость – воображаемая плоскость в пространстве, по которой производится разрыв связей (рисунок 41).



(1) – секущая плоскость

Рис. 41. Пример секущей плоскости, параллельной плоскости c в структуре сапфира, на проекции $\{11\bar{2}0\}$

На рисунке 41 секущая плоскость параллельна плоскости c , но в модели предусмотрена возможность задания плоскости c произвольной нормалью. Первоначально эта плоскость проводилась через начало координат, а затем прогонялась в направлении нормали c интервалом 1000 точек на ячейку. Во время прогонки для каждого положения плоскости считалось количество разрываемых связей (длинных и коротких) и атомов на поверхности отрыва. По этим данным в каждой точке по формуле (12) рассчитывалась свободная поверхностная энергия.

По результатам прогонки строились профили поверхностной энергии кристаллографического направления (рисунок 42).

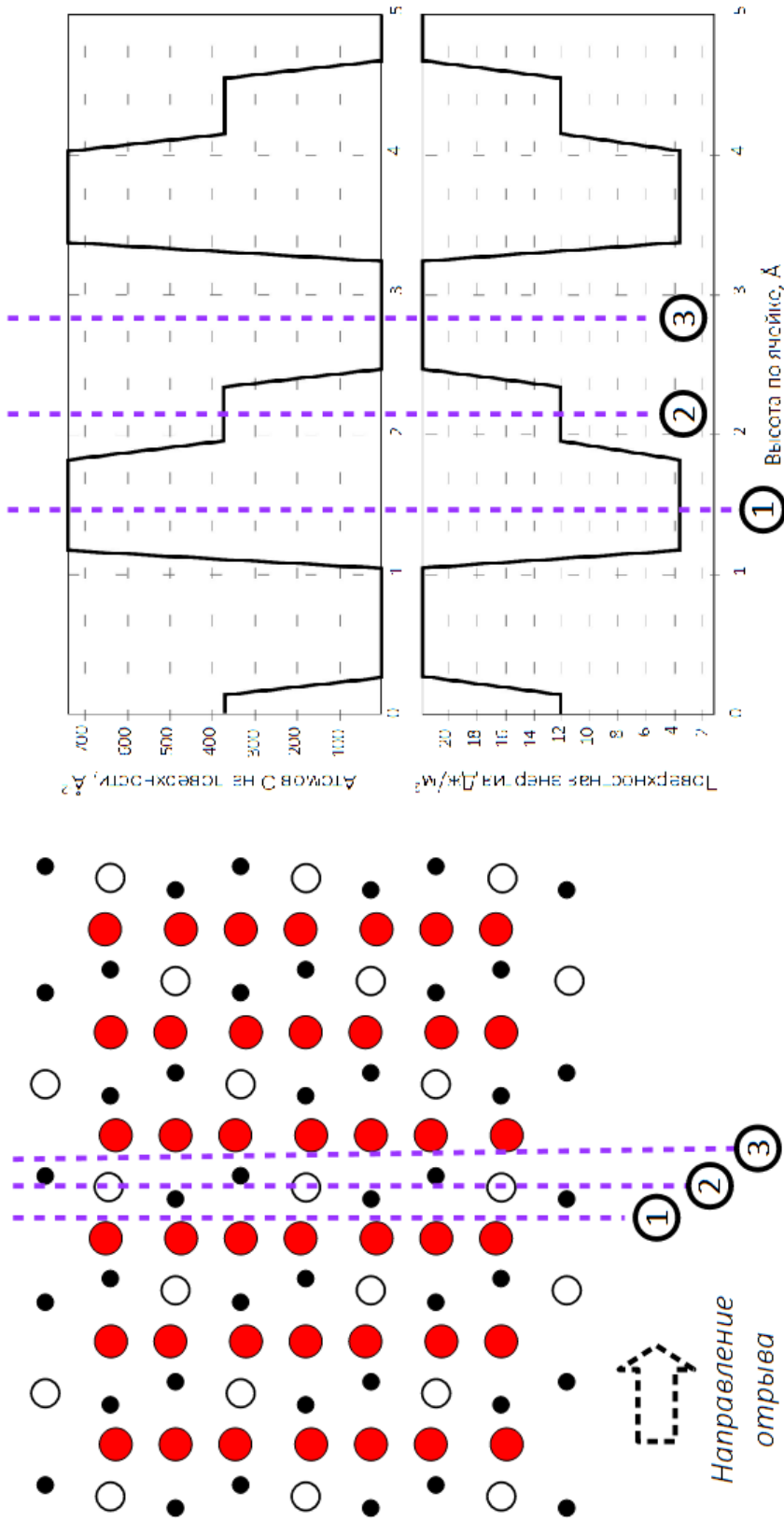


Рис. 42. Три положения секущей плоскости и профили энергии для плоскости пинакоида

Очевидно, что при перемещении секущей плоскости будет изменяться количество разрываемых связей, а также количество и сорт атомов на поверхности отрыва. В связи с этим было вполне логично ожидать, что одна и та же простая форма будет иметь разные значения поверхностной энергии в зависимости от положения секущей плоскости, что хорошо иллюстрируется рисунком 42.

Наблюдается хорошая корреляция между количеством атомов кислорода на поверхности отрыва и значение свободной поверхностной энергии.

В соответствии с законом минимума поверхностной энергии за значение поверхностной энергии каждой плоскости нами было принято минимальное значение энергии в пределах интервала прогонки.

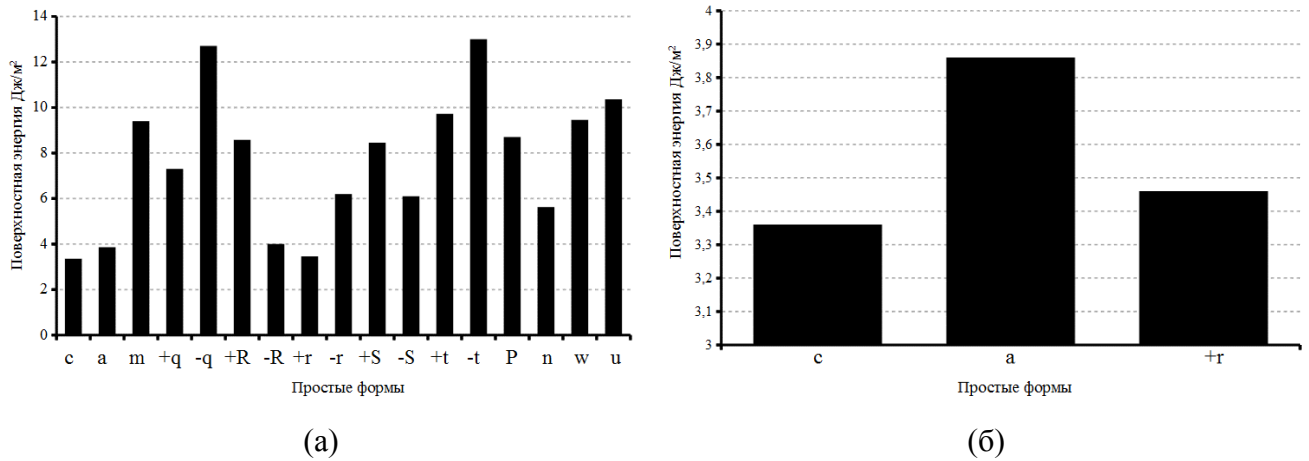
Тексты программ, при помощи которых производились расчеты, приведены в приложении 2.

5.3.3 Расчет поверхностных энергий наиболее важных плоскостей кристаллов сапфира

В соответствии с алгоритмом, изложенным выше, был выполнен расчет для наиболее важных плоскостей сапфира. Результаты расчетов приведены в таблице 11 и проиллюстрированы графиками (рисунок 43 а, б).

Таблица 11 – Результаты расчетов свободной поверхностной энергии методом разрыва связей для наиболее важных плоскостей сапфира

Плоскость	$E_{\text{пов}}, \text{Дж/м}^2$	Плоскость	$E_{\text{пов}}, \text{Дж/м}^2$
$c \{0001\}$	3,36	$+t \{20\bar{2}1\}$	9,72
$a \{11\bar{2}0\}$	3,86	$-t \{\bar{2}021\}$	13
$m \{10\bar{1}0\}$	9,4	$P \{11\bar{2}6\}$	8,7
$+q \{10\bar{1}5\}$	7,3	$n \{11\bar{2}3\}$	5,62
$-q \{\bar{1}015\}$	12,7	$w \{11\bar{2}2\}$	9,45
$+R \{10\bar{1}4\}$	8,58	$u \{11\bar{2}1\}$	10,36
$-R \{\bar{1}014\}$	4		
$+r \{10\bar{1}2\}$	3,46		
$-r \{\bar{1}012\}$	6,2		
$+S \{10\bar{1}1\}$	8,45		
$-S \{\bar{1}011\}$	6,1		



(a) – для всех рассчитанных граней
 (б) – для граней главной экспериментальной последовательности

Рис. 43. Гистограммы иллюстрирующие результаты расчета свободной поверхностной энергии для идеальной структуры сапфира

В результате расчетов свободной поверхностной энергии методом разрыва связей была установлена следующая последовательность граней по возрастанию значения свободной поверхностной энергии: $c \{0001\}$ - $+r \{10\bar{1}2\}$ - $a \{11\bar{2}0\}$ - $-R \{\bar{1}014\}$ - $-S \{\bar{1}011\}$ - $-r \{\bar{1}012\}$ - $m \{10\bar{1}0\}$ - $+q \{10\bar{1}5\}$ и т.д.

Таким образом, расчет методом разрыва связей описывает экспериментально установленную последовательность граней.

5.3.4 Расчет сечений указательной поверхности свободных поверхностных энергий кристаллов сапфира

5.3.4.1 Ориентация $c \{0001\}$

В отличие от других существующих методов, возможности предлагаемого нами метода гораздо шире и позволяют не только рассчитывать значения поверхностных энергий для выбранных направлений, но и прогнозировать полный набор возможных граней путем расчета сечений указательной поверхности свободной поверхностной энергии (кристаллографического пояса).

В связи с тем, что в отличие от метода Ненасыщенности связей, мы не берем в расчет плохо обоснованные с физической точки зрения параметры такие как толщина элементарного пакета, а опираемся только на расположение атомов в структуре, длины и направленность связей (делаем расчет из первых принципов), возникает возможность поворачивать секущую плоскость на любой заданный угол (углы) в пространстве и рассчитывать поверхностную энергию любого направления в кристаллической решетке.

Как было сказано ранее, для моделирования мы берем блок 30х30х30 элементарных ячеек. При расчете в таком блоке минимальный угол поворота должен быть не менее $0,2^\circ$, так как при меньшем угле поворота секущая плоскость не будет разрывать новых связей (будет оставаться внутри одной плоскости), т.е. для большего разрешения необходимо брать больший блок. Но в связи с тем, что такое разрешение, совпадает с разрешением нашего экспериментального метода (глава 4), остановимся на нем.

Для получения сечения указательной поверхности, перпендикулярного выбранному направлению, секущая плоскость проводилась через это направление и поворачивалась на 360° с шагом $0,2^\circ$. При этом на каждом шаге проводился расчёт, аналогичный описанному в разделе 5.3.2.

Результаты расчета для ориентации c приведен на графике (рисунок 44). Угол поворота отсчитывается от направления a .

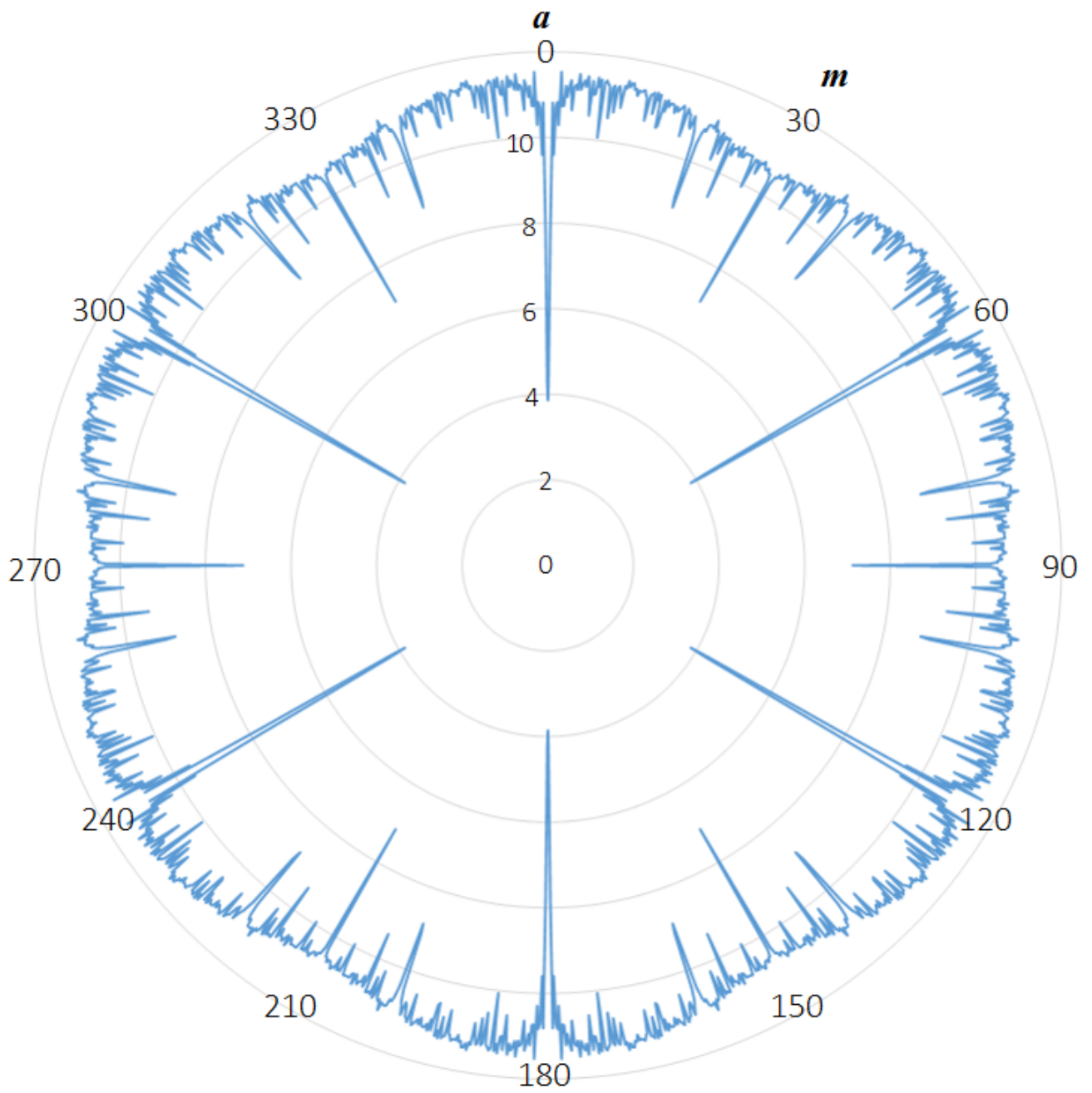


Рис. 44. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c

Рисунок 44 хорошо иллюстрирует, что на сечении указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c выделяются все элементы симметрии, соответствующие данному сечению кристалла, т.е. расчетное сечение соответствует симметрии кристаллов сапфира.

Метод Разрыва связей не учитывает температуру, т.е. расчет ведется для абсолютного нуля. Как известно, при абсолютном нуле каждому рациональному направлению в кристалле должен соответствовать сингулярный минимум, что также хорошо иллюстрируется графиком (рисунок 44).

Для того чтобы оценить ограничение кристалла при температуре $2000\text{ }^{\circ}\text{C}$ нами было сделано следующее.

Допустим, что внешняя огибающая, представленного на рисунке 44, графика является энергией неограниченной поверхности кристалла. Построив ее по максимальным точкам сечения, указательной поверхности свободной поверхностной энергии кристалла получим график с гладкой кривой, отвечающей симметрии кристалла (рисунок 45).

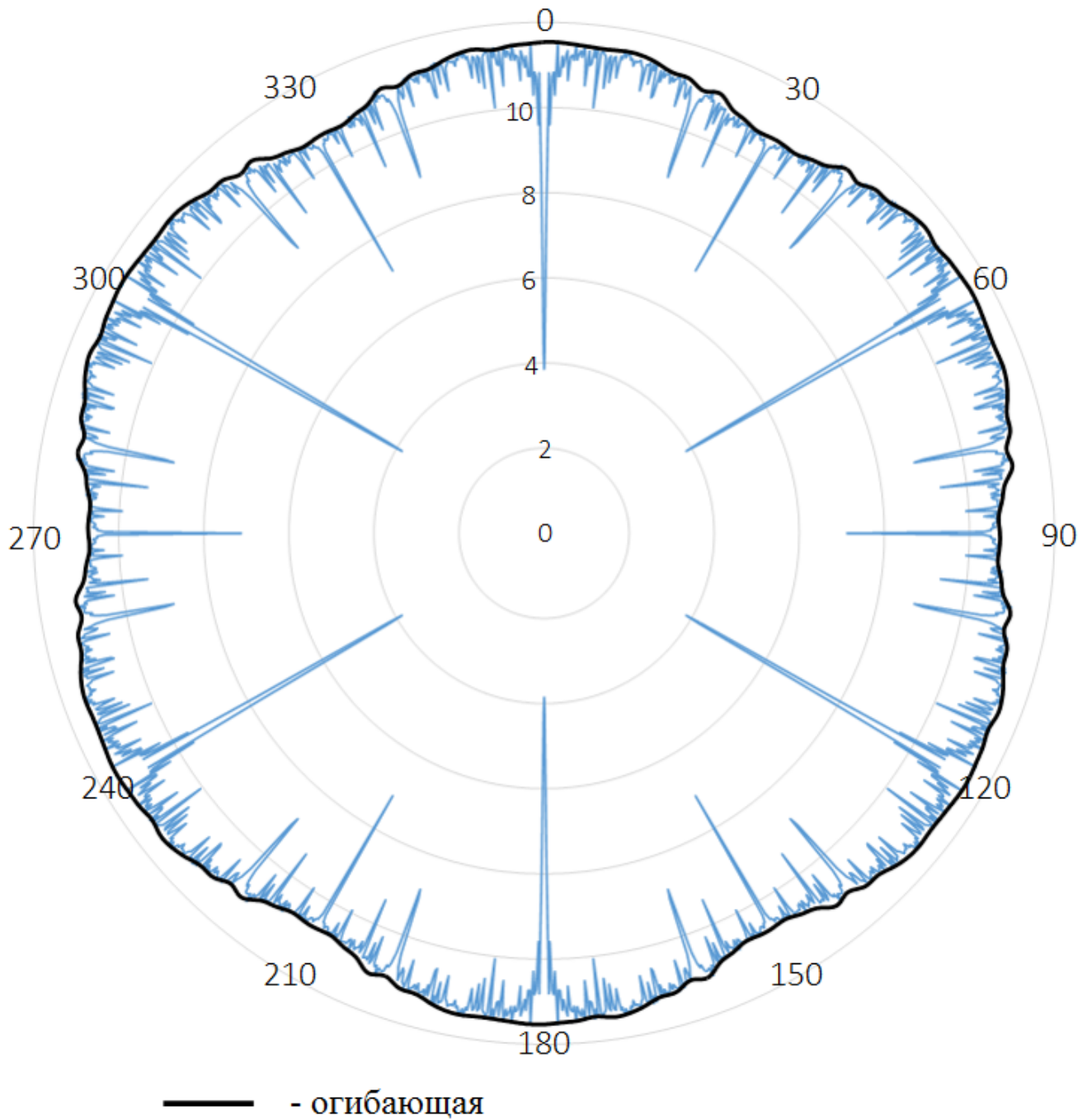


Рис. 45. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c с нанесенной оггибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации

Построенная огибающая (рисунок 45), как уже было сказано, отвечает свободной поверхностной энергии неограниченной поверхности кристалла. Для того чтобы учесть температуру введем в расчет параметр, который отражал бы энергию тепловых колебаний атомов. Известно, что таким параметром является kT – произведение постоянной Больцмана на заданную температуру.

Теперь умножим kT на количество разрываемых связей для каждой точки на огибающей и отложим получившееся значение внутрь графика (рисунок 46). Назовём полученную кривую линией пороговых значений (рисунок 46).

Для расчета мы используем значение температуры равное температуре плавления $2050\text{ }^{\circ}\text{C}$.

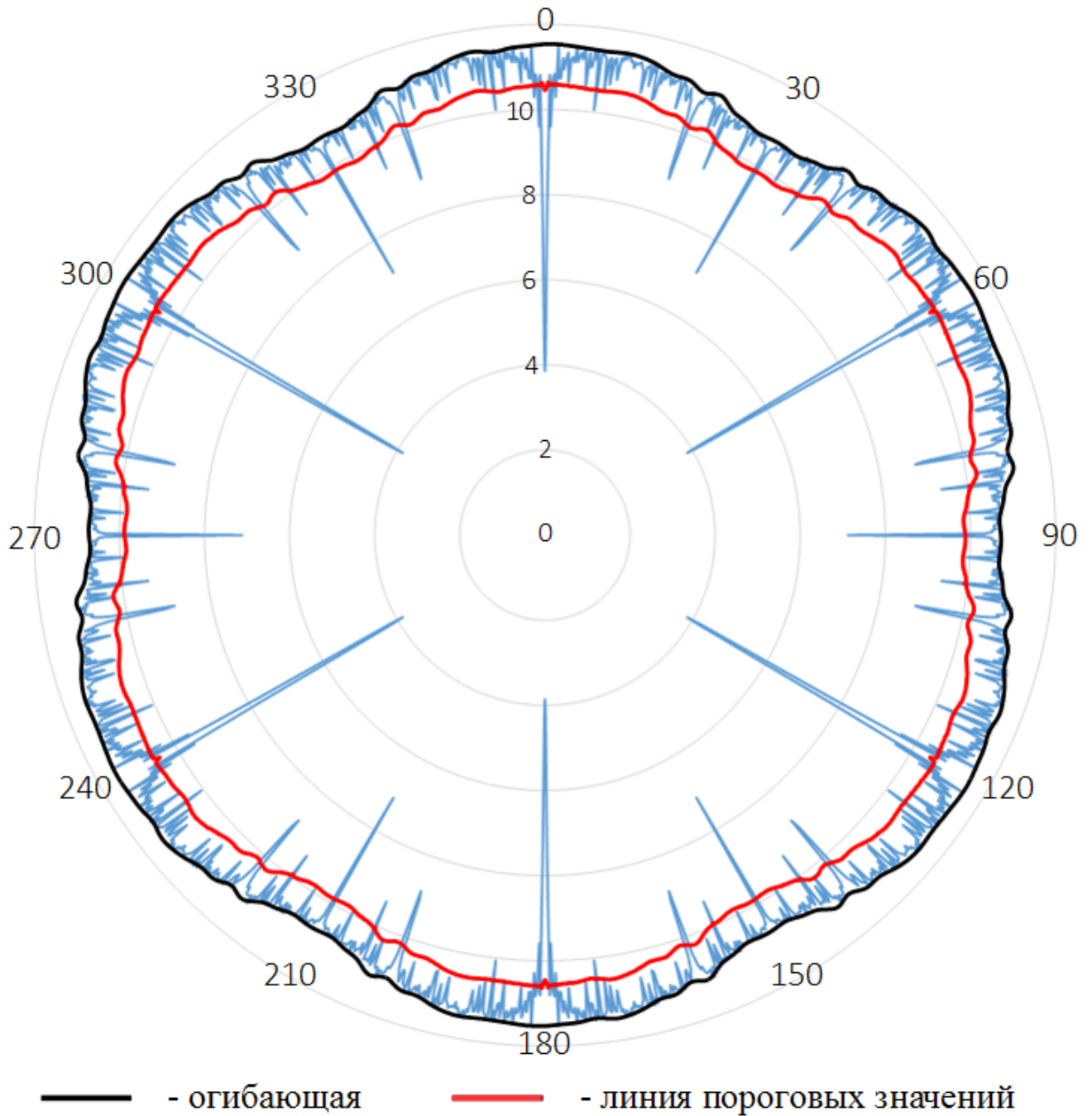


Рис. 46. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c с нанесенной оггибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации и линией пороговых значений.

График, приведенный на рисунке 46, хорошо иллюстрирует, что в результате действий, описанных выше, получился «коридор» между кривой, отвечающей свободной поверхностной энергии неограниченной поверхности и пороговой кривой. Допустим, что сингулярные минимумы, попадающие внутрь такого «коридора», будут размываться тепловыми колебаниями. Напротив, те сингулярные минимумы, которые пробивают пороговую кривую, будут проявляться на кристалле. Сохраняя на кривой поверхностной энергии только эти минимумы, получаем сечение указательной поверхности для температуры 2050 °С (рисунок 47).

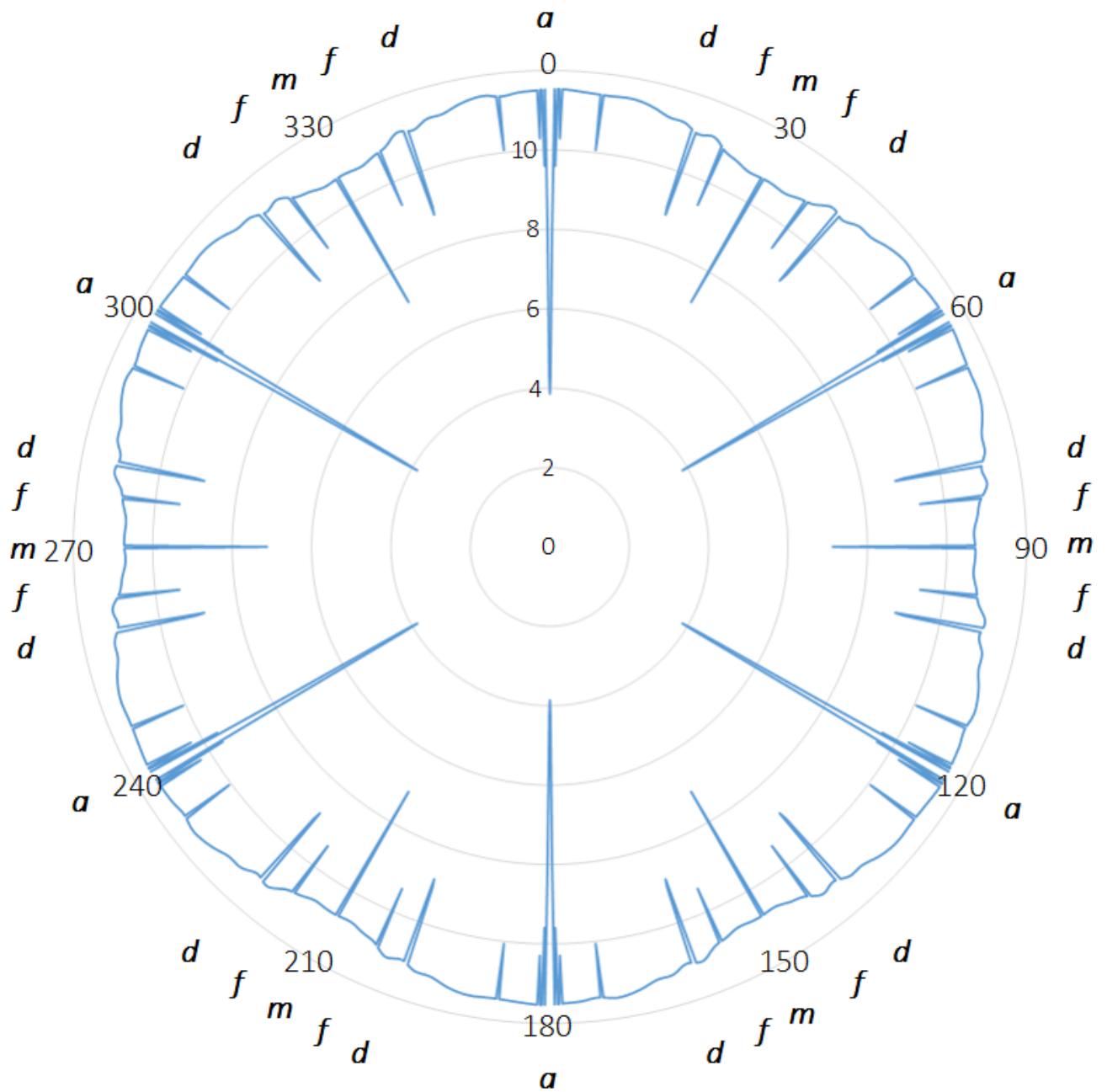


Рис. 47. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c при $2050\text{ }^{\circ}\text{C}$

График, приведенный на рисунке 47, хорошо иллюстрирует, что после учета энергии тепловых колебаний для 2050 °С остались считанные единицы значимых сингулярных минимумов, которые могут проявиться на кристалле в виде граней. Список таких граней для ориентации *c* приведен в таблице 12.

Таблица 12 – Возможные сингулярные грани на боковой поверхности кристаллов сапфира ориентации *c*.

Обозначение	<i>a</i>	-	<i>d</i>	-	<i>f</i>	<i>m</i>
ϕ_a	0	6,6	19,2	21,8	23,4	30
Поверхностная энергия, Дж/м ²	3,4	10,6	10,25	10,3	9,5	9,4

В связи с тем, что удалось рассчитать полное сечение указательной поверхности свободной поверхностной энергии и спрогнозировать ряд граней, интересно сравнить полученные результаты расчетов и эксперимента. Для этого необходимо график сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации *c* при 2050 °С построить в виде линейного графика и сопоставить два получившихся графика – экспериментальный и расчетный. Такое сопоставление приведено на рисунке 48.

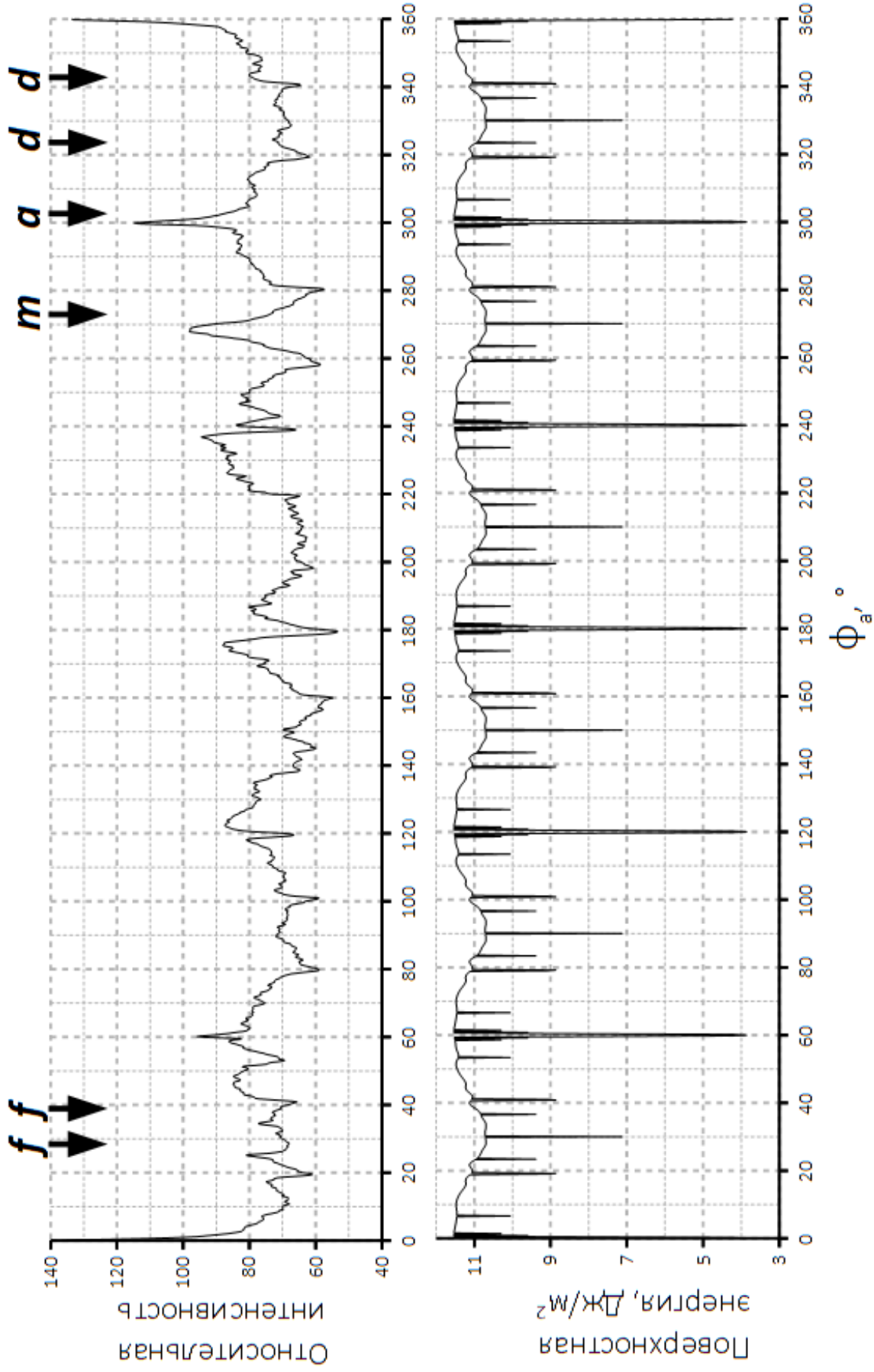


Рис. 48. Сопоставление экспериментально полученного графика (сверху) и графика полученного в результате расчета сечения указательной поверхности свободной поверхности энергии кристалла сапфира ориентации c при 2050°C

Графики, приведенные на рисунке 48, хорошо иллюстрируют, что всем особенностям на графике, полученном в результате расчета сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации c при 2050 °С, отвечают особенности на боковой поверхности профилированных монокристаллических стержней сапфира той же ориентации.

Сопоставление экспериментально выявленных граней с результатами расчетов показало, что на боковой поверхности монокристаллических стержней сапфира ориентации c проявляются все грани, которые были предсказаны расчетом, т.е. грани 6 простых форм.

Плоскость a проявляется, как уже говорилось ранее, в виде зеркальной грани и имеет значение поверхностной энергии менее 4 Дж/м².

Аналогичный анализ проведем для ориентаций a $\{11\bar{2}0\}$ и m $\{10\bar{1}0\}$.

5.3.4.2 Ориентация a $\{11\bar{2}0\}$

На рисунке 49 приведено сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a с нанесенной огибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации и линии пороговых значений.

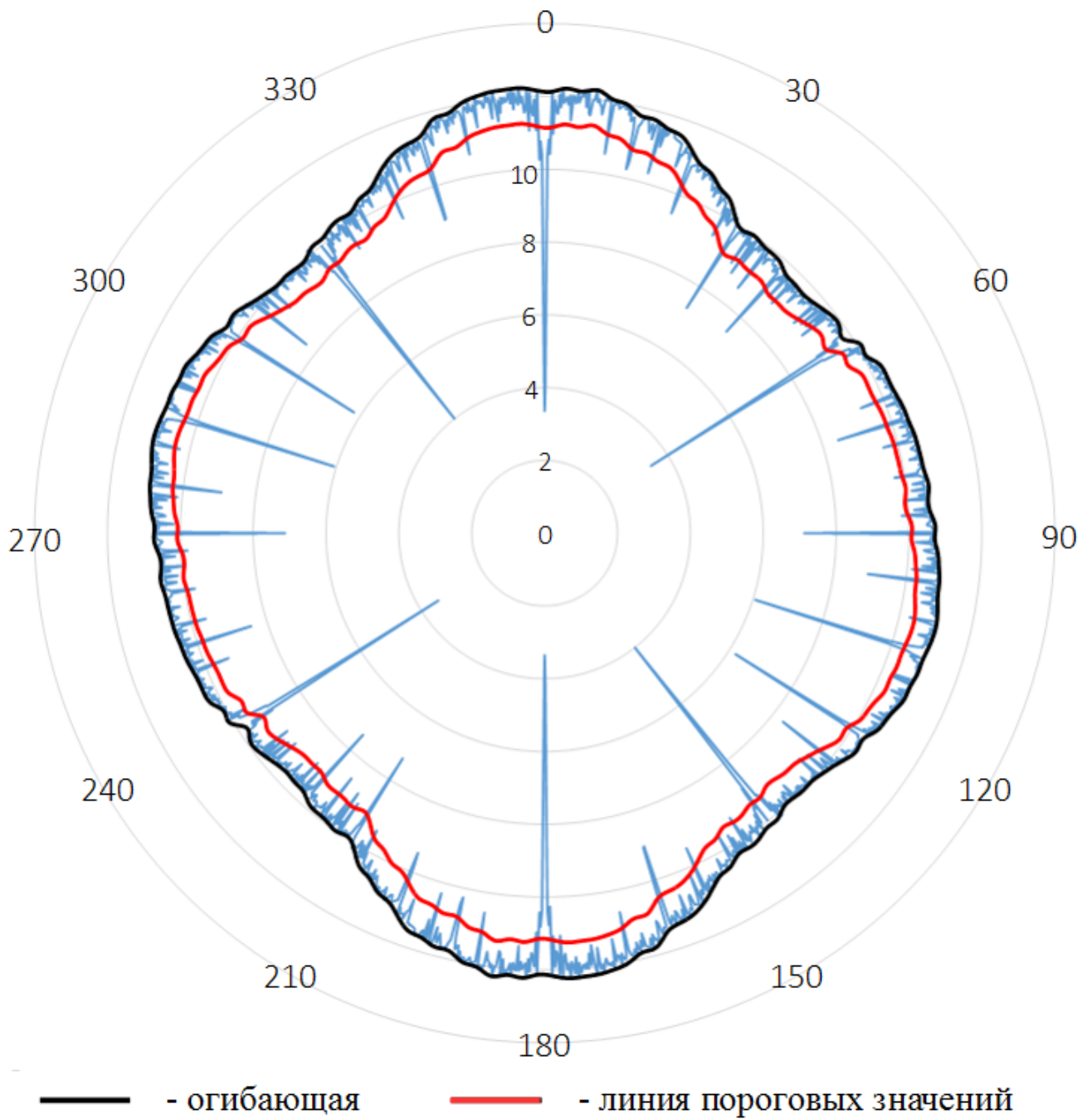


Рис. 49. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a с нанесенной оггибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации и линией пороговых значений

По алгоритму, изложенному в параграфе 5.3.4.1, отсеем те сингулярные минимумы, которые размоются тепловыми колебаниями при заданной температуре. В результате получим график сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при $2050\text{ }^{\circ}\text{C}$ (рисунок 50).

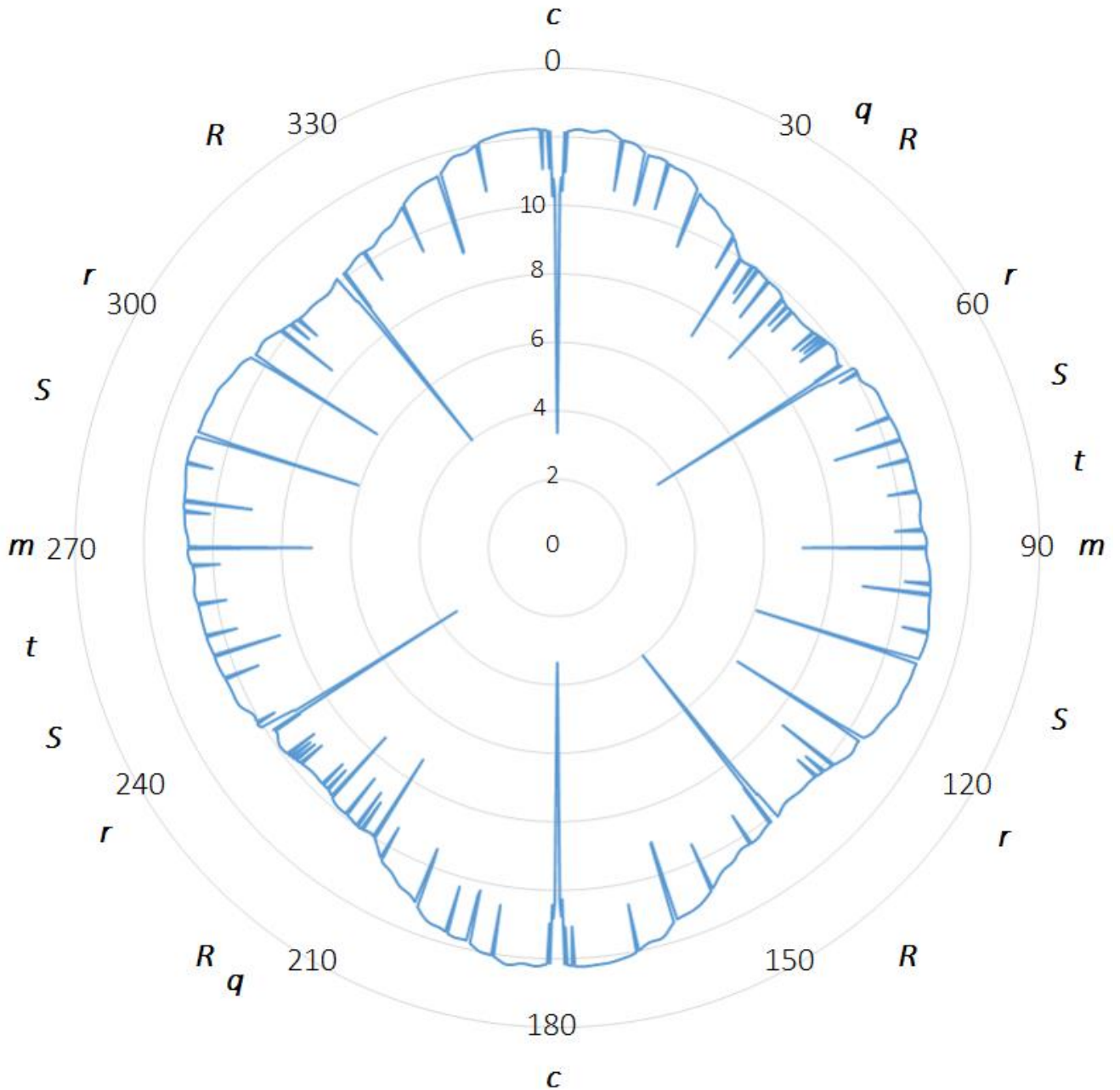


Рис. 50. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при $2050\text{ }^{\circ}\text{C}$

Список граней, которые могут проявиться на кристалле ориентации a при 2050 °С приведен в таблица 13.

Таблица 13 – Возможные сингулярные грани на боковой поверхности кристаллов сапфира ориентации a .

Обозначение	c	-	-	-	-	-	$+q$	-	-	$+R$
ϕ_c	0	9	12,6	16	21,6	29,4	32,2	34,6	35,6	38,2
Поверхностная энергия, Дж/м²	3,36	10,6	10,25	10,3	9,46	9,4	7,3	9	8,8	8,58
Обозначение	-	-	-	-	-	-	-	-	$+r$	-
ϕ_c	42	43,4	44,6	49,8	50,8	51,6	52,4	57	57,6	59,6
Поверхностная энергия, Дж/м²	7,46	8,95	8,88	8,96	9,1	9,15	8,92	8,91	3,46	9,51
Обозначение	-	$+S$	$+a$	$+t$	-	m	$-d$	$-p$	-	$-S$
ϕ_c	68,4	72,4	75,8	81	87,2	90	95,6	97,2	103	107,6
Поверхностная энергия, Дж/м²	9,34	8,45	9,6	9,72	9,82	7,13	10,13	8,94	10,28	6,1
Обозначение	$-r$	-	-	-	$-R$	-	-	-	-	-
ϕ_c	122,4	128,4	130,2	131,6	141,8	147	155,8	162,4	168,8	177,8
Поверхностная энергия, Дж/м²	6,2	8,35	9,44	9,34	4	9,34	9,5	9	10,62	11,1

На рисунке 51 приведены графики сопоставления сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при 2050 °С, построенный в виде линейного графика, и график относительной интенсивности отражений для той же ориентации.

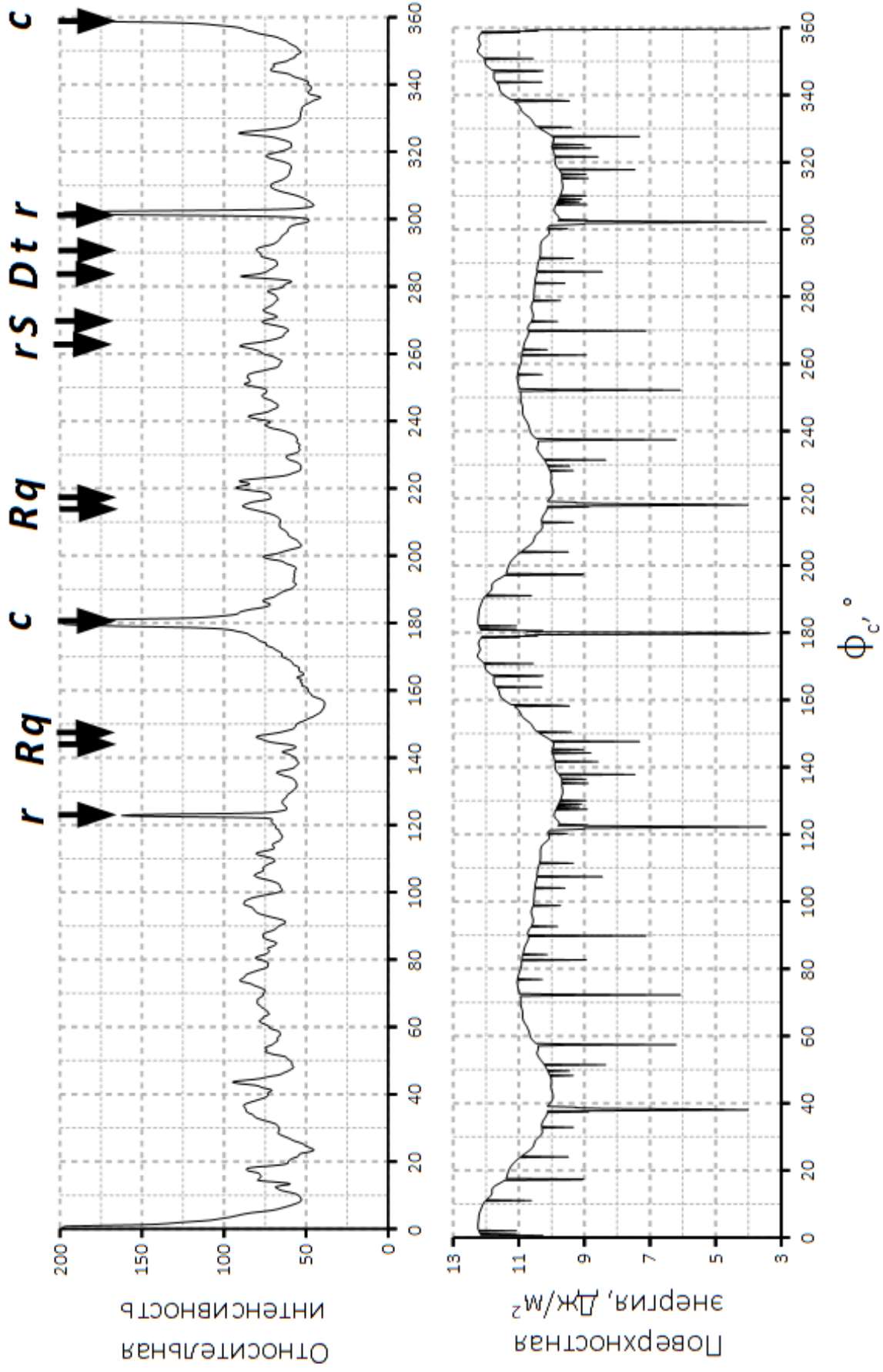


Рис. 51. Сопоставление экспериментально полученного графика (сверху) и графика полученного в результате расчета сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при 2050 °С

Графики, приведенные на рисунке 51, хорошо иллюстрируют, что всем особенностям на графике полученном в результате расчета сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при 2050 °С отвечают особенности на боковой поверхности профилированных монокристаллических стержней сапфира той же ориентации.

Сопоставление экспериментально выявленных граней с результатами расчетов показало, что на боковой поверхности монокристаллических стержней сапфира ориентации a проявляются 31 грань из 40, предсказанных расчетом.

Плоскости c и r проявляются, как уже говорилось, в виде зеркальных граней и имеют значение поверхностной энергии менее 4 Дж/м².

5.3.4.3 Ориентация m $\{10\bar{1}0\}$

На рисунке 52 приведено сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m с нанесенной огибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации и линии пороговых значений.

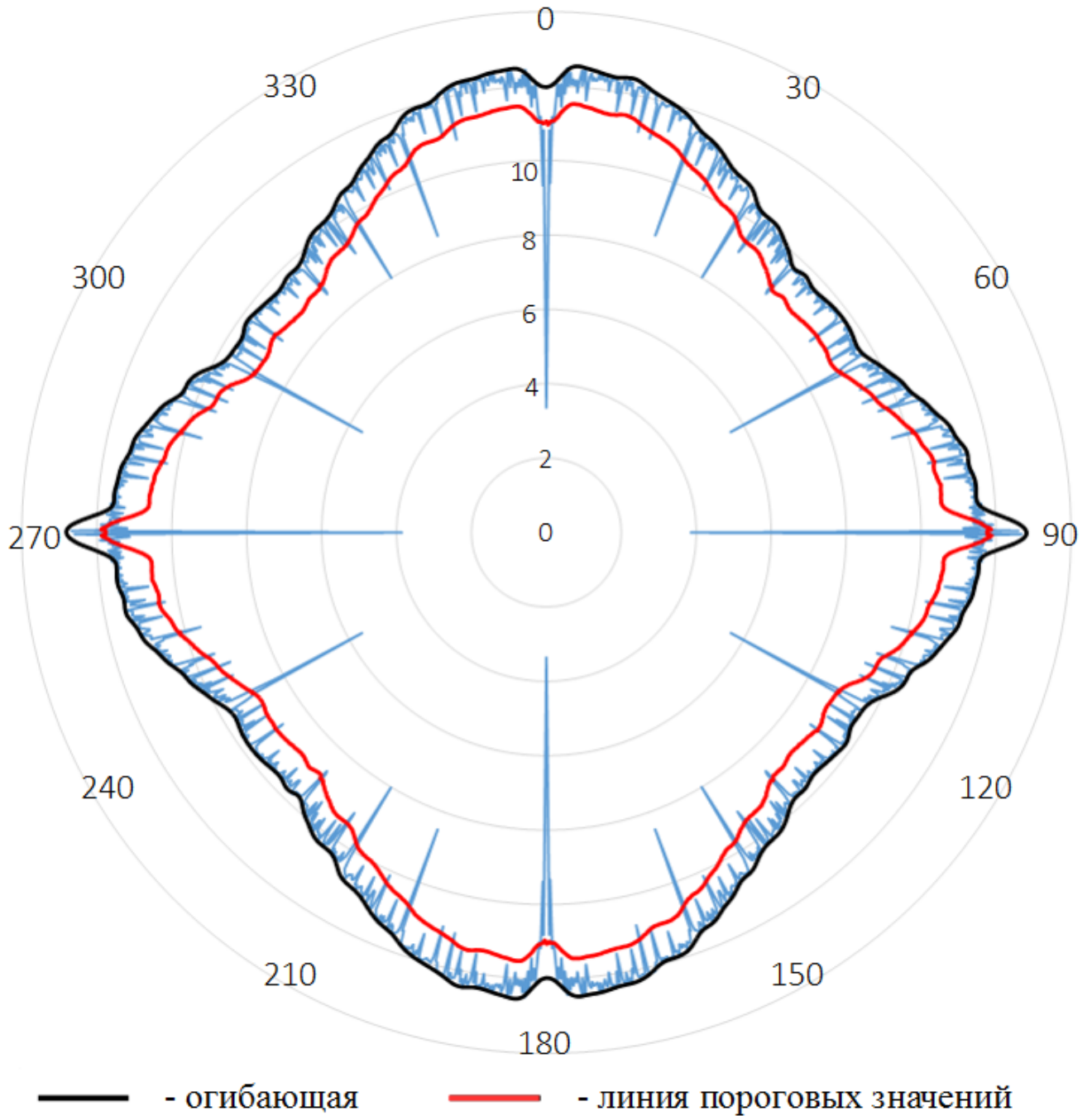


Рис. 52. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m с нанесенной огибающей, отвечающей свободной поверхностной энергии неограниченной поверхности кристалла сапфира данной ориентации и энергией тепловых колебаний

По алгоритму, изложенному в параграфе 5.3.4.1, отсеем те сингулярные минимумы, которые размоются тепловыми колебаниями при заданной температуре. В результате получим график сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации a при $2050\text{ }^\circ\text{C}$ (рисунок 53).

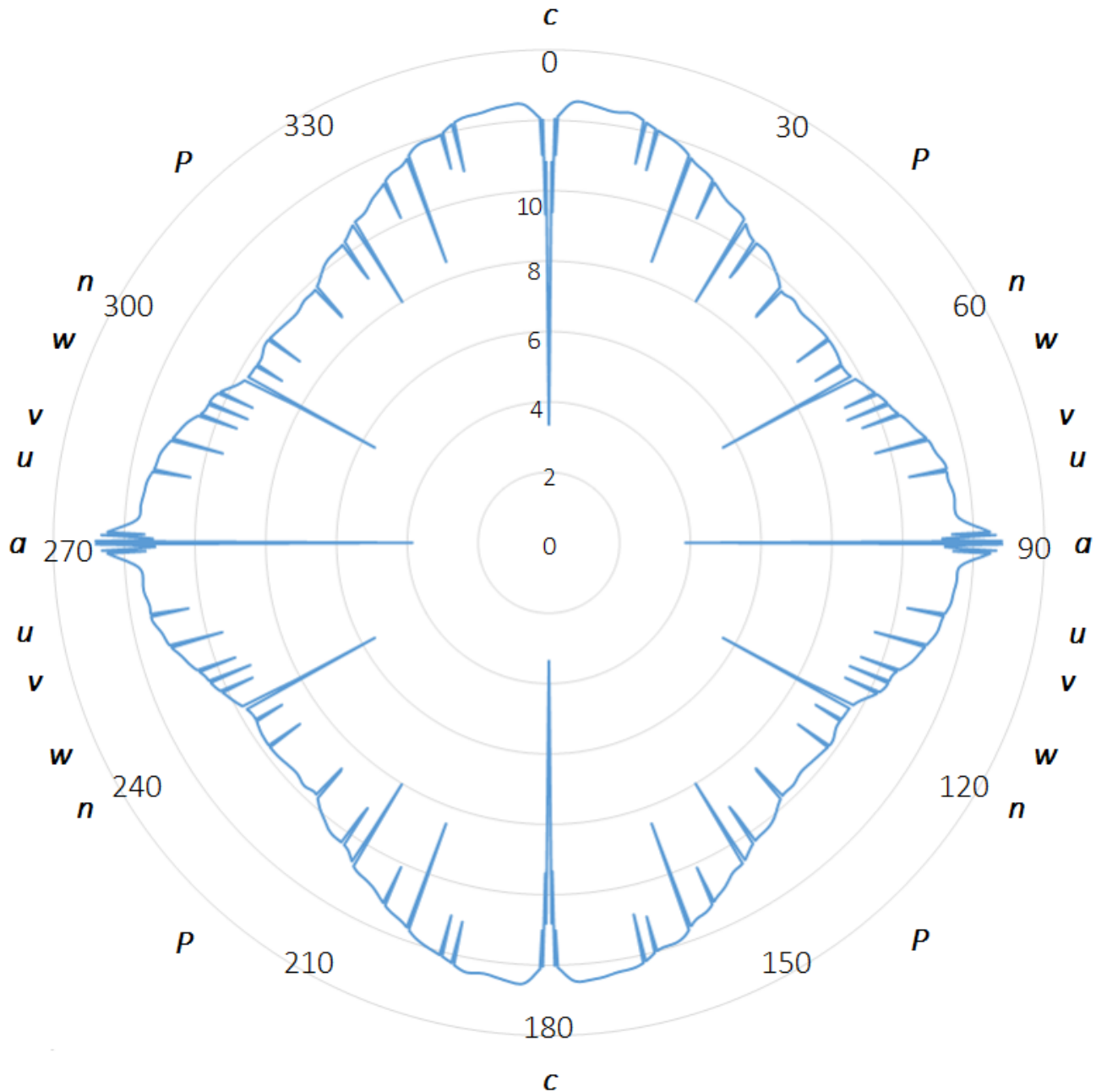


Рис. 53. Сечение указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m при $2050\text{ }^\circ\text{C}$

Список граней, которые могут проявиться на кристалле ориентации m при 2050 °С приведен в таблица 14.

Таблица 14 – Возможные сингулярные грани на боковой поверхности кристаллов сапфира ориентации m .

Обозначение	<i>c</i>	-	-	-	-	-	-	<i>P</i>	<i>M</i>	-
ϕ_c	0	12,8	14,6	20	24,4	31,2	34,4	42,4	53,8	58,6
Поверхностная энергия, Дж/м²	3,36	11	10,94	8,5	10,12	8	9,41	8,7	8,72	8,84
Обозначение	<i>n</i>	<i>B</i>	-	<i>w</i>	<i>v</i>	<i>u</i>	<i>a</i>			
ϕ_c	61,2	65,4	67,6	69,8	74,6	79,6	90			
Поверхностная энергия, Дж/м²	5,62	9,21	9,13	9,45	9,6	10,36	3,86			

На рисунке 54 приведены графики сопоставления сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m при 2050 °С, построенный в виде линейного графика, и графики относительной интенсивности отражений для той же ориентации.

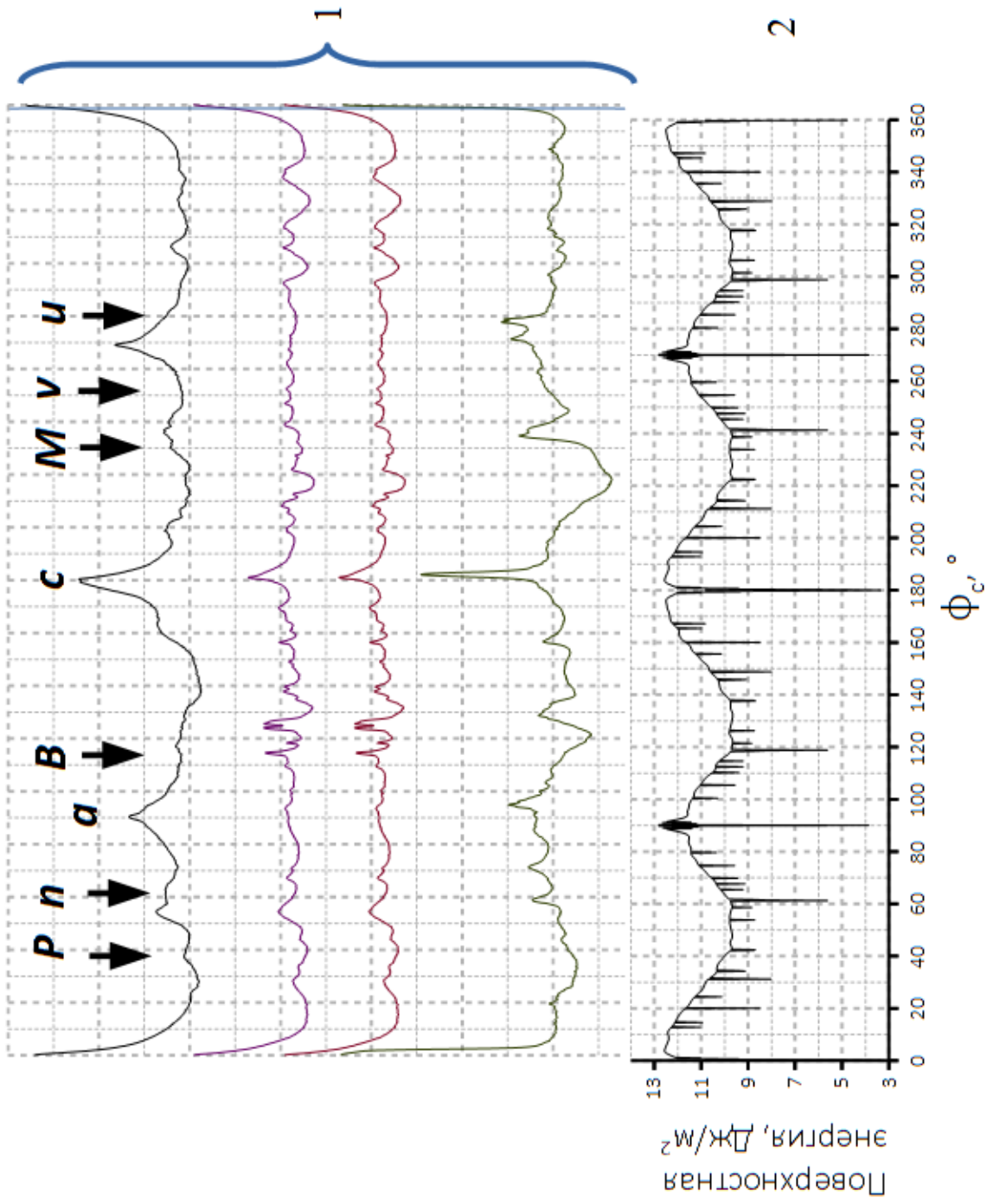


Рис. 54. Сопоставление экспериментально полученных графиков (1) и графика полученного в результате расчета сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m при 2050 °С (2)

Как уже говорилось ранее в главе 4, морфология боковой поверхности кристаллов данной ориентации плохо воспроизводится по длине, поэтому сравнению так же, как и в главе 4 следует подвергать несколько проанализированных уровня кристалла, что продемонстрировано на рисунке 54.

Графики, приведенные на рисунке 54, хорошо иллюстрируют, что всем особенностям на графике полученном в результате расчета сечения указательной поверхности свободной поверхностной энергии кристалла сапфира ориентации m при 2050 °С, отвечают особенности на боковой поверхности профилированных монокристаллических стержней сапфира той же ориентации.

Сопоставление экспериментально выявленных граней с результатами расчетов показало, что на боковой поверхности монокристаллических стержней сапфира ориентации m все грани, которые были предсказаны расчетом, т.е. грани 17 простых форм.

Плоскости c и a проявляются, как уже говорилось, в виде зеркальных граней и имеют значение поверхностной энергии менее 4 Дж/м².

5.4 Результаты и выводы

1. Был разработан новый метод расчета свободной поверхностной энергии, позволяющий рассчитывать как энергии отдельных кристаллографических плоскостей, так и полные сечения указательной поверхности свободной поверхностной энергии.
2. Был проведен расчет свободных поверхностных энергий разработанным методом для решетки сапфира, который показал, что:
 - рассчитанная последовательность ориентаций, имеющих минимальные значения поверхностной энергии, $E_c < E_r < E_a$ соответствуют экспериментально наблюдаемой последовательности размеров граней $h_c > h_r > h_a$;
 - минимальную поверхностную энергию имеют плоскости упакованные атомами кислорода;
 - графики сечений указательной поверхности свободной поверхностной энергии полностью соответствуют кристаллографии сапфира, т.е. на них можно выделить все элементы симметрии соответствующего сечения;
 - путём сопоставления полученных графиков со средней удельной энергией тепловых колебаний при температуре плавления определён набор кристаллографических направлений, в которых возможно образование граней на боковой поверхности кристаллов в форме стержней.

3. Сопоставление экспериментальных и расчетных данных показало, что:

- Для цилиндрических кристаллов сапфира, выращенных способом Степанова, предсказанный расчетом набор возможных граней на боковой поверхности хорошо совпадает с экспериментально наблюдаемым набором;
- в виде зеркальных проявляются грани, имеющие значение поверхностной энергии менее 4 Дж/м²;
- показано, что, в отличие от существовавших ранее представлений, боковая поверхность монокристаллических стержней сапфира, выращенных способом Степанова не является строго цилиндрической с отдельными плоскими гранями; ее морфология может отличаться от цилиндра, задаваемого формообразователем, как на огранённых, так и на неогранённых участках. Она полностью определяется анизотропией свободной поверхностной энергии, и согласуется с результатами расчетов, проведенных разработанным нами методом.

ЗАКЛЮЧЕНИЕ

Данная работа выполнялась на базе федерального государственного бюджетного учреждения науки Физико-технического института им. А.Ф. Иоффе Российской академии наук в лаборатории Физики профилированных кристаллов в период с 2011 по 2015 г.

Основные результаты работы состоят в следующем.

Были выращены профилированные монокристаллические стержни лейкосапфира диаметром 8 мм трех ориентаций *c*, *a* и *m* (по 4 кристалла каждой ориентации). Кристаллы выращивались в одной тепловой зоне с максимальным контролем условий роста.

В рамках данной работы разработан новый оптический прибор с цифровой регистрацией – видеогониограф. Создан действующий лабораторный макет прибора. Разработан метод количественного анализа морфологии боковой поверхности цилиндрических монокристаллов для обработки данных, получаемых с помощью видеогониографа.

В результате видеогониографического исследования морфологии боковой поверхности цилиндрических монокристаллов сапфира различных ориентаций, выращенных способом Степанова, было выявлено следующее:

1. На боковой поверхности цилиндрических кристаллов сапфира диагностировано следующее количество граней:
 - На кристаллах ориентации *a* $[11\bar{2}0]$ – 31 грань;
 - На кристаллах ориентации *c* $[0001]$ – 6 граней;
 - На кристаллах ориентации *m* $[10\bar{1}0]$ – 17 граней.

Полученный результат – более полный набор гранных форм на боковой поверхности профилированных кристаллов сапфира, чем тот, который давал предшествующий метод – метод фотогониографии. С полным перечнем граней и их угловыми значениями можно ознакомиться в главе 4 данной работы.

2. На боковой поверхности цилиндрических кристаллов сапфира установлено три морфологических единицы, влияющих на конечную форму растущего кристалла: плоская грань, ребро, уплощение.
3. Установлено, что наиболее стабильны во время роста кристаллы ориентации *a* и *c*, кристаллы ориентации *m* сильно чувствительны даже к незначительным изменениям условий роста, что необходимо учитывать во время планирования технологического процесса.

4. В виде зеркальных граней на боковой поверхности цилиндрических монокристаллов проявляются грани пинакоида c , положительного ромбоэдра r и гексагональной призмы a .
5. Установлена следующая последовательность граней на боковой поверхности цилиндрических монокристаллов сапфира по их ширине – $c > r > a >$ (все остальные). Те грани, которые не проявляются в виде зеркальных на боковой поверхности невозможно измерить, поэтому они приняты меньшими, чем три зеркальных.
6. Подтверждена главная последовательность граней на боковой поверхности профилированных цилиндрических монокристаллов сапфира $h_c:h_r:h_a$ с соотношением средней ширины этих граней как 8:4:1.

Таким образом, видеогониограф не только подтвердил и позволил на количественном уровне описать закономерности, установленные ранее на фотогониографе на качественном уровне, но и дал большое количество нового материала.

Для объяснения полученных результатов и прогнозирования проявления на боковой поверхности цилиндрических профилированных кристаллов граней возможных простых форм был разработан новый экспрессный метод расчета свободной поверхностной энергии кристаллов по сечениям указательной поверхности – метод разрыва связей. Этот метод позволил рассчитать, как значения энергий отдельных кристаллографических плоскостей, так и полные сечения указательной поверхности свободной поверхностной энергии выбранных ориентаций и спрогнозировать полный набор граней на боковой поверхности кристаллов. Полученный набор хорошо совпал с экспериментальными данными:

1. В результате расчетов свободных поверхностных энергий разработанным методом для решетки сапфира, было выявлено, что:
 - рассчитанные последовательности поверхностных энергий $E_c < E_r < E_a$ соответствуют экспериментально наблюдаемой последовательности размеров граней $h_c > h_r > h_a$;
 - графики сечений указательной поверхности свободной поверхностной энергии полностью соответствуют кристаллографии сапфира, т.е. на них можно выделить все элементы симметрии соответствующего сечения;
 - минимальную поверхностную энергию имеют плоскости, упакованные атомами кислорода.
2. Сопоставление экспериментальных и расчетных данных показало, что:
 - в виде зеркальных проявляются грани, имеющие значение поверхностной энергии менее 4 Дж/м²;

- сечение монокристаллических стержней сапфира, выращенных способом Степанова, отличается от формы, задаваемой формообразователем, из-за проявления отдельных плоских граней, ребер и уплощенных участков.

По результатам работы можно сделать следующие основные выводы:

1. Морфология боковой поверхности цилиндрических монокристаллов сапфира, в первую очередь, определяется анизотропией свободной поверхностной энергии. Условия роста влияют лишь на степень развития морфологических особенностей;
2. Боковая поверхность монокристаллических стержней сапфира, выращенных способом Степанова не является строго цилиндрической. Ее морфология определяется анизотропией поверхностной энергии и согласуется с результатами расчетов, проведенных разработанным нами методом.
3. Видеогониограф является уникальным инструментом изучения морфологии кристаллов, который может также служить устройством для неразрушающего экспрессного анализа разориентации цилиндрических кристаллов, что сейчас не позволяет делать ни один существующий метод;
4. Данные, полученные с помощью видеогониографа и разработанного метода расчета, вносят весомый вклад в понимание таких фундаментальных свойств твердого тела, как свободная поверхностная энергия и ее анизотропия, а также влияние этих свойств на реальную форму кристалла;
5. Полученные экспериментальные и расчетные результаты могут служить основой для дальнейшего изучения влияния ограничения на форму профилированных кристаллов и развития методов управления формой с целью оптимизации процесса роста. Исключительно важными с этой точки зрения являются полученные в работе полные сечения указательной поверхности свободной поверхностной энергии.

Таким образом, все поставленные в данной работе задачи решены. Выявлены закономерности в проявлении гранных форм и других морфологических особенностей боковой поверхности профилированных кристаллов сапфира и показано их влияние на окончательную форму кристалла.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ЗОС – программное обеспечение по захвату, обработке и сохранению полученных данных.

ПЦС – метод периодических цепей связей;

ПЭВМ – персональная электронно-вычислительная машина, используемая в данной работе как средство приема и записи сигнала с цифровой видеокамеры видеогониографа.

СЛОВАРЬ ТЕРМИНОВ

Фотограмма – снимок боковой поверхности цилиндрического монокристалла, сделанный на фотобумаге методом фотогониографии.

Видеограмма – снимок боковой поверхности цилиндрического монокристалла, сделанный методом видеогониографии и обработанный с помощью ЗОС, сопровождающийся графиком относительных интенсивностей отражений, количественно описывающим наблюдающиеся на снимке морфологические особенности.

Огибающая – кривая, которая в каждой своей точке касается хотя бы одной кривой семейства и каждым своим отрезком касается бесконечного множества этих кривых.

Указательная поверхность – поверхность, образованная концами отрезков, отложенных из начала координат, с длиной равной значению свойства в данном направлении.

СПИСОК ЛИТЕРАТУРЫ

1. Степанов А.В. Получение монокристаллов определенной формы. /А.В. Степанов// Сб. Материалы Первого совещания по получению профилированных полупроводниковых монокристаллов способом Степанова и перспектива их применения в приборостроении. – 1968. – С. 4.
2. Антонов П.И. Симметрия монокристаллов германия, выращиваемых из расплава по способу Степанова. /П.И. Антонов, Н.С. Григорьев, А.В. Степанов// Изв. АН СССР, сер. физ. – 1971. – Т35. – №3 – С. 447.
3. Антонов П.И. Фотогониография профилированных кристаллов германия. /П.И. Антонов, Н.С. Григорьев, Л.П. Вахмянин// Изв. АН СССР, сер. физ. – 1972. – Т36. – №3. – С 501.
4. Антонов П.И. Образование слоев роста на гранях и двойникование профилированных монокристаллов антимонида индия. /П.И. Антонов, С.И. Бахолдин, Ю.Г. Носов, Е.С. Калитина// Изв. АН СССР, сер. физ. – 1983. – Т47. – №2. – С 315.
5. Носов Ю.Г. Возникновение дефектов структуры в профилированных монокристаллах антимонида индия. /Ю.Г. Носов, П.И. Антонов// Изв. АН СССР, сер. физ. – 1973. – Т37. – №11. – С 2334.
6. Носов Ю.Г. Ограничение кристаллов сапфира, выращенных из расплава способом Степанова. /Ю.Г. Носов, С.И. Бахолдин, В.М. Крымов// ЖТФ. – 2009. – №79 (2). – С. 76.
7. Добровинская Е.Р. Энциклопедия сапфира. /Е.Р. Добровинская, Л.А. Литвинов, В.В. Пищик – Харьков: Институт монокристаллов, 2004. – 508 с.
8. Классен-Неклюдова М.В. Рубин и сапфир. /М.В. Классен-Неклюдова, Х.С. Багдасаров – Москва: «Наука», 1974. – 235 с.
9. С.И. Бахолдин. Применение метода периодических цепей связей для расчета простых форм сапфира. /С.И. Бахолдин, В.Н. Маслов// Кристаллография. – 2014. – Т59. – № 4. – С. 649.
10. С.И. Бахолдин. Моделирование габитусных форм кристаллов сапфира с использованием принципов подхода периодических цепей связей. /С.И. Бахолдин, В.Н. Маслов// Кристаллография. – 2015. – Т60. – № 2. – С. 336.
11. Степанов А.В. Будущее металлообработки. /А.В. Степанов – Л.: «Лениздат», 1963.
12. Антонов П.И. Получение профилированных монокристаллов и изделий способом Степанова. /П.И. Антонов, Л.М. Затуловский, А.С. Костыгов, Д.И. Левинзон, С.П. Никаноров, В.В. Пеллер, В.А. Татарченко, В.С. Юферев – Л.: «Наука», 1981. – 280 с.

13. Маслов В.Н. Выращивание профилированных полупроводниковых монокристаллов. /В.Н. Маслов – Москва: «Металлургия», 1977. – 327 с.
14. Степанов А.В. Новый способ получения изделий из расплавленного металла. Авт. свид. №429880. – Бюл. изобрет., 1974. – №20. – С. 27.
15. Ратников Д.Г. Исследование процесса бесконтактного выращивания профилированных кристаллов кремния с использованием электромагнитного формообразователя. /Д.Г. Ратников, В.И. Добровольская, Л.Р. Лев// Изв. АН СССР, сер. физ. – 1971. – Т35. – №3. – С 466.
16. Мошанов В.И. Исследование фазовых переходов на ступенчатых поверхностях кремния. /В.И. Мошанов, Б.З. Ольшанецкий, С.И. Стенин// Поверхность. – 1986. – №4. – С. 38.
17. Krolmer H. On The (110) orientation as the prepared orientation for the molecular beam epitaxial growth of GaAs on Ge, GaP on Si, and similar Zinblende-on-diamond systems. /H. Krolmer, K.J. Polasko, S.C. Wright// Appl. Phys. Lett. – 1980. – Vol. 36. – №9. – P. 763.
18. Pukite P.R. Supression of antiphase domains in the growth of GaAs on Ge (100) by molecular beam epitaxy. /P.R. Pukite, P.I. Cohen// J. Crystal Growth. – 1987. – Vol. 81. – iss. 1-4. – P. 214.
19. Наноминералогия. Ультра- и микродисперсное состояние минерального вещества /под ред. Асхабова А.М., Ракин В.А., Юшкин Н.П. – СПб.: «Наука», 2005. – 577 с.
20. Любалин М.Д. Рост кристаллов в расплаве. Кристаллографический анализ и эксперимент. /М.Д. Любалин – СПб.: «Наука», 2008. – 390 с.
21. Современная кристаллография. Т.3: Образование кристаллов /А.А. Чернов, Е.И. Гиваргизов, Х.С. Багдасаров и др. – М.: «Наука», 1980. – 408 с.
22. Хартман П. Зависимость морфологии кристалла от кристаллической структуры. /П. Хартман// Рост кристаллов. – 1967. – Т7. – С. 8024.
23. Hartman P. A theory of crystal morphology. /P. Hartman, W.G. Perdok// Proc. Koninkl. Nederland. Akad. Wetenschap. Ser. – 1952. – B55. – P. 134.
24. Hartman P. On the relation between structure and morphology of crystals I. /P. Hartman, W.G. Perdok// Acta. Cryst. – 1955. – №8. – P. 49.
25. Hartman P. On the relation between structure and morphology of crystals II. /P. Hartman, W.G. Perdok// Acta. Cryst. – 1955. – №8. – P. 521.
26. Hartman P. On the relation between structure and morphology of crystals III. /P. Hartman, W.G. Perdok// Acta. Cryst. – 1955. – №8. – P. 525.
27. Hartman P. Crystal form and crystal structure. /P. Hartman// Phys. and Chem. Of the organic Solid State. – 1963. – Vol. 1. – P. 369.

28. Браве О. Избранные научные труды. Кристаллографические этюды. /О. Браве – Л.: «Наука», 1974. – 420 с.
29. Donnay J.D.H. A new law of crystal morphology, extending the law of Bravais. /J.D.H. Donnay, D. Harker// *Am. Mineral.* – 1937. – Vol. 22 – P. 446.
30. Hartman P. An interpretation of the law of Donnay and Harker. /P. Hartman, W.G. Perdok// *Am. Mineral.* – 1956. – Vol. 41 – P. 449.
31. Watson G.W. Atomistic simulation of dislocations, surfaces and interfaces in MgO. /G.W. Watson, E.T. Kelsey, N.H. de Leeuw, D.J. Harris, S.C. Parker// *J. Chem. Soc., Faraday Trans.* – 1996. – Vol. **92** – P. 433.
32. Любалин М.Д. Анализ идеальных кристаллических граней. /М.Д. Любалин// *Изв. АН СССР, сер. физ.* – 1988. – Т 52 – № 10. – С. 1970.
33. Tasker P.W. Surfaces of magnesia and Alumina. /P.W. Tasker// *Am. Ceram. Soc.* – 1984. – Vol. 10. – P. 176.
34. Mackrodt W.C. The morphology of α -Al₂O₃ and α -Fe₂O₃: The importance of Surface relaxation. /W.C. Mackrodt, R.J. Davey, S.N. Black, R. Docherty// *J. Cryst. Growth.* – 1987. – Vol. 80 – № 2. – P. 441.
35. Causa M. Ab-Initio Characterization of the (0001) and (10-10) crystal Faces of α -Alumina. /M. Causa, R. Dovesi, C. Pisani, C. Roetti// *Surf. Sci.* – 1989. – Vol. 215 – № 1-2. – P. 259.
36. Ellis D.E. Cluster models of Bulk, Surface, and Impurity Structure in α -Alumina. /D.E. Ellis, J. Guo, D.J. Lam// *J. Am. Ceram Soc.* – 1994. – Vol. 77 – № 2. – P. 398.
37. Manassidis I. Structure and Energetics of Alumina Surfaces Calculated from First Principles. /I. Manassidis, M.J. Gillan// *J. Am. Ceram. Soc.* – 1994. – Vol. 77. – № 2. – P. 335.
38. Gay D.H. MARVIN: A New Computer Code for Studying Surfaces and Interfaces and Its Application to Calculating the Crystal Morphologies of Corundum and Zircon. /D.H. Gay, A.L. Rohl// *J. Chem. Soc., Faraday Trans.* – 1995. – Vol. 91 – № 5. – P. 925.
39. Blonski S. Molecular Dynamics simulations of α -Alumina and γ -Alumina surfaces. /S. Blonski, S.H. Garofalini// *Surf. Sci.* – 1993. – Vol. 295 – № 1-2 – P. 263.
40. Suzuki H. Simulation of Surface and Grain Boundary Properties of Alumina by Molecular Dynamics Method. /H. Suzuki, H. Matsubara, J. Kishino, T. Kondoh// *J. Ceram. Soc. Jpn.* – 1998. – Vol. 106. – № 12 – P. 1215.
41. Marmier A. Ab initio morphology and surface thermodynamics of α -Al₂O₃. /A. Marmier, S.C. Parker// *Phys. Rev. B.* – 2004. – Vol. 69. – P. 115409.
42. Kitayama M. The Wulff Shape of Alumina: III, Undoped Alumina. /M. Kitayama, A.M. Glaeser// *J. Am. Ceram. Soc.* – 2002. – Vol. 85. – № 3. – P. 611.

43. Хартман П. Структурная морфология корунда. /П. Хартман// Записки Всесоюзн. Минерал. Об-ва. – 1962. – Т. 91. – С. 672.
44. Шафрановский И.И. Рентгеновский способ определения морфологически важных граней кристаллов. /И.И. Шафрановский, В.И. Михеев// Записки Всесоюзн. Минерал. Об-ва. – 1949 – Т. 78. – С. 166.
45. Хонигман Б. Рост и форма кристаллов. /Б. Хонигман – М.: ИЛ, 1961. – 224 с.
46. Краснова Н.И. Генезис минеральных индивидов и агрегатов. /Н.И. Краснова, Т.Г. Петров – СПб.: Невский курьер, 1995. – 228 с.
47. Cahn J.W. Surface tensions of III-V compounds and their relationship to spontaneous bending of thin crystals. /J.W. Cahn, R.Z. Hanneman// Surf. Sci. – 1964. – Vol. 1. – № 4. – P. 387.
48. Chase A.B. Habit Modification of Corundum Crystals Grown from Molten $\text{PbF}_2\text{-Bi}_2\text{O}_3$. /A.B. Chase// J. Amer. Ceram. Soc. -1966. – Vol. 45 – № 5. – P. 233.
49. Тимофеева В.А. /В.А. Тимофеева, Н.И. Лукьянова// Кристаллография. – 1967. – Т. 12. – № 1. – С. 98.
50. Ландсберг Г.С. Оптика. /Г.С. Ландсберг – М.: Физматлит, 2003. – 848 с.
51. Бахолдин С.И. Метод видеогониографии для изучения огранения кристаллов сапфира, выращенных способом Степанова. /С.И. Бахолдин, В.Н. Маслов// ЖТФ. – 2015. – Т. 85. – № 9. – С. 132.
52. Шафрановский И.И. Кристаллы минералов. Ч.1: Плоскогранные формы. /И.И. Шафрановский – Л.: ЛГУ, 1957. – 222 с.
53. Бахолдин С.И. Моделирование поверхностных энергий кристаллов сапфира. /С.И. Бахолдин, В.Н. Маслов// ФТТ. – 2015. – Т. 57. – №. 6. – С. 1213.

ПРИЛОЖЕНИЕ 1.1

отражение от грани с переходными областями

Размеры кристалла и области граниРадиус кристалла, мм $R := 5$ Ширина грани $2a$, мм $a := 1.3$ Ширина переходной области b , мм $b := 1.3$ Половина центрального угла, опирающегося на грань и две переходные кривые, рад $\phi_b := \arcsin\left(\frac{a+b}{R}\right)$ Переводя в градусы $\phi_{bdeg} := \frac{180}{\pi} \phi_b$ $\phi_{bdeg} = 31.332$ Расстояние от центра до хорды, стягивающей концы переходных кривых p_b , мм $p_b := \frac{a+b}{\tan(\phi_b)}$ $p_b = 4.271$ Смещение в конце переходной кривой v_b , мм $v_b := 0.25$ Расстояние от центра до грани p_a , мм $p_a := p_b + v_b$ $p_a = 4.521$ Половина центрального угла, опирающегося на грань, рад $\phi_a := \arctan\left(\frac{a}{p_a}\right)$ Переводя в градусы $\phi_{adeg} := \frac{180}{\pi} \phi_a$ $\phi_{adeg} = 16.043$ Координаты датчика, мм $x_0 := \frac{R}{\sqrt{2}}$ $x_0 = 3.536$ $y_0 := 12$ $\Delta f_c := \frac{R \cdot \sqrt{2}}{4}$ Интенсивность при отражении от цилиндра $I_c := \frac{\Delta f_c}{y_0 - \Delta f_c}$ $I_c = 0.172765$ **Задание областей*****Переходная кривая выше плоской грани***Число точек $N_{p1} := 25$ $ip1 := 0..N_{p1} - 1$ Шаг по прямой, мм $dl := \frac{b}{N_{p1} + 1}$ $dl = 0.05$ Вектор координат u , мм $up1_{ip1} := (N_{p1} - ip1)dl$

(Не включая точки сопряжения с окружностью и плоской гранью.)

Задаём кривую $AA := \frac{vb}{b^3}$ $AA = 0.113792$ $Fv(u) := AA \cdot u^3$ $vp1_{ip1} := Fv(up1_{ip1})$ её наклон $BB := \frac{\tan(\phi_b)}{b^2}$ $BB = 0.360225$ $Fdv(u) := BB \cdot u^2$ $dvp1_{ip1} := Fdv(up1_{ip1})$ и кривизну $CC := \frac{1}{R \cdot b}$ $CC = 0.153846$ $Fkv(u) := CC \cdot u$ $kvp1_{ip1} := Fkv(up1_{ip1})$

Векторы декартовых (s и t, мм) и полярных (ρst, мм, θ, рад, θdeg, град) координат

$$sp1_{ip1} := pa - vp1_{ip1}$$

$$tp1_{ip1} := a + up1_{ip1}$$

$$\rho_{stp1} := \sqrt{sp1^2 + tp1^2}$$

$$\theta_{p1_{ip1}} := \operatorname{atan}\left(\frac{tp1_{ip1}}{sp1_{ip1}}\right)$$

$$\theta_{degp1} := \frac{180}{\pi} \theta_{p1}$$

Вектор положений квазифокуса, мм

$$\Delta fp1_{ip1} := \frac{\sqrt{2}}{4 \cdot kvp1_{ip1}}$$

Вектор углов нормалей к кривой с осью s, радианы

$$\phi_{Op1_{ip1}} := \operatorname{atan}(dvp1_{ip1})$$

Переводя в градусы

$$\phi_{Op1deg} := \frac{180}{\pi} \cdot \phi_{Op1}$$

Плоская грань

Число точек

$$Nf := 53$$

$$if := 0..Nf - 1$$

Векторы декартовых (s и t, мм) и полярных (ρst, мм, θ, рад, θdeg, град) координат

(Включая точки сопряжения с переходными кривыми.)

$$sf_{if} := pa$$

$$tf_{if} := \left(\frac{Nf - 1}{2} - if\right) \cdot dl$$

$$\rho_{stf} := \sqrt{sf^2 + tf^2}$$

$$\theta_{f_{if}} := \operatorname{atan}\left(\frac{tf_{if}}{sf_{if}}\right)$$

$$\theta_{degf} := \frac{180}{\pi} \theta_f$$

Вектор нормалей к грани с осью s рад и град

$$\phi_{Of_{if}} := 0$$

$$\phi_{Ofdeg_{if}} := 0$$

Векторы кривизны и положений квазифокуса
(задаются условно, не используются)

$$kf_{if} := 0.1$$

$$\Delta ff_{if} := 0$$

Переходная кривая ниже плоской грани

Число точек

$$Np2 := 25$$

$$ip2 := 0..Np2 - 1$$

Вектор координат u, мм

$$up2_{ip2} := (1 + ip2)dl$$

(Не включая точки сопряжения с окружностью и плоской гранью.)

Задаём кривую

$$vp2_{ip2} := Fv(up2_{ip2})$$

её наклон

$$dvp2_{ip2} := -Fdv(up2_{ip2})$$

и кривизну

$$kvp2_{ip2} := Fkv(up2_{ip2})$$

Векторы декартовых (s и t, мм) и полярных (ρst, мм, θ, рад, θdeg, град) координат

$$sp_{ip2}^2 := pa - vp_{ip2}^2 \quad tp_{ip2}^2 := -(a + up_{ip2}^2)$$

$$\rho_{stp2} := \sqrt{sp_{ip2}^2 + tp_{ip2}^2} \quad \theta_{p2ip2} := \operatorname{atan}\left(\frac{tp_{ip2}^2}{sp_{ip2}^2}\right) \quad \theta_{degp2} := \frac{180}{\pi} \theta_{p2}$$

Вектор положений квазифокуса, мм $\Delta fp_{ip2} := \frac{\sqrt{2}}{4 \cdot kvp_{ip2}^2}$

Вектор углов нормалей к кривой с осью s, радианы $\phi_{p2ip2} := \operatorname{atan}(dvp_{ip2}^2)$

Переводя в градусы $\phi_{p2deg} := \frac{180}{\pi} \cdot \phi_{p2}$

Формирование суммарных массивов

Общее число точек $N := Np1 + Nf + Np2 \quad N = 103 \quad i := 0..N - 1 \quad z_i := i$

Векторы декартовых (s и t, мм) и полярных (ρst, мм, θ, рад, θdeg, град) координат

$$s_{\Sigma} := \operatorname{stack}(sp1, sf, sp2) \quad t_{\Sigma} := \operatorname{stack}(tp1, tf, tp2)$$

$$\rho_{st\Sigma} := \operatorname{stack}(\rho_{stp1}, \rho_{stf}, \rho_{stp2}) \quad \theta_{\Sigma} := \operatorname{stack}(\theta_{p1}, \theta_f, \theta_{p2}) \quad \theta_{deg\Sigma} := \operatorname{stack}(\theta_{degp1}, \theta_{degf}, \theta_{degp2})$$

Вектор нормалей к суммарной кривой с осью s рад и град

$$\phi_{0\Sigma} := \operatorname{stack}(\phi_{0p1}, \phi_{0f}, \phi_{0p2}) \quad \phi_{0\Sigma deg} := \frac{180}{\pi} \cdot \phi_{0\Sigma}$$

Вектор кривизны, 1/мм $k := \operatorname{stack}(kvp1, kf, kvp2)$

Вектор положений квазифокуса, мм $\Delta f_{\Sigma} := \operatorname{stack}(\Delta fp1, \Delta ff, \Delta fp2)$

Решение системы уравнений для определения угла ψ

Вычисляем коэффициенты $a1 := t_{\Sigma} - y0 \quad a2 := t_{\Sigma} + y0 \quad c1_i := \tan(\phi_{0\Sigma_i})$

$$b1 := 1 + c1 \quad b2 := 1 - c1$$

$$c2 := s_{\Sigma} - x0 \quad c3 := s_{\Sigma} + x0$$

$$A1_i := -a2_i \cdot b1_i \cdot b2_i$$

$$B1_i := -2 \cdot \left[a2_i \cdot (b1_i)^2 - (a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b2_i \right]$$

$$C1_i := a2_i \cdot b1_i \cdot b2_i + 4 \cdot (a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b1_i + (a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b2_i$$

$$D1_i := -2 \cdot \left[(a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b2_i - (a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b1_i + (s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b2_i \right]$$

$$E1_i := - \left[(a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b2_i + 4 \cdot (s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b1_i + a1_i \cdot b1_i \cdot b2_i \right]$$

$$F1_i := 2 \cdot \left[(s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b2_i - a1_i \cdot (b1_i)^2 \right]$$

$$G1_i := a1_i \cdot b1_i \cdot b2_i$$

$$A2_i := -2 \cdot c1_i \cdot c3_i$$

$$B2_i := -4 \cdot \left[c1_i \cdot t\Sigma_i + \left[(c1_i)^2 - 1 \right] \cdot c3_i \right]$$

$$C2_i := 2 \cdot \left[c1_i \cdot c2_i - 4 \cdot \left[(c1_i)^2 - 1 \right] \cdot t\Sigma_i + 6 \cdot c1_i \cdot c3_i \right]$$

$$D2_i := 4 \cdot \left[\left[(c1_i)^2 - 1 \right] \cdot c2_i + 6 \cdot c1_i \cdot t\Sigma_i - \left[1 - (c1_i)^2 \right] \cdot c3_i \right]$$

$$E2_i := -2 \cdot \left[6 \cdot c1_i \cdot c2_i + 4 \cdot t\Sigma_i \cdot \left[1 - (c1_i)^2 \right] + c1_i \cdot c3_i \right]$$

$$F2_i := 4 \cdot \left[\left[1 - (c1_i)^2 \right] \cdot c2_i - t\Sigma_i \cdot c1_i \right]$$

$$G2_i := 2 \cdot c1_i \cdot c2_i$$

$$A := A1 - A2$$

$$B := B1 - B2$$

$$C := C1 - C2$$

$$D := D1 - D2$$

$$E := E1 - E2$$

$$F := F1 - F2$$

$$G := G1 - G2$$

Решением системы является вектор тангенсов половинного угла $\text{tg}\psi d2$
 Но полином 6-й степени имеет 6 корней. Обозначим массы корней
 $\text{root0} \dots \text{root1}$.

$$\begin{array}{l}
 F0\psi(n) := \left. \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_0 \end{array} \right\} \\
 F1\psi(n) := \left. \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_1 \end{array} \right\}
 \end{array}$$

$$\begin{array}{l}
 F2\psi(n) := \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\
 \text{roots} \leftarrow \text{polyroots}(\text{data}) \\
 \text{roots}_2
 \end{array}
 \qquad
 \begin{array}{l}
 F3\psi(n) := \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\
 \text{roots} \leftarrow \text{polyroots}(\text{data}) \\
 \text{roots}_3
 \end{array}$$

$$\begin{array}{l}
 F4\psi(n) := \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\
 \text{roots} \leftarrow \text{polyroots}(\text{data}) \\
 \text{roots}_4
 \end{array}
 \qquad
 \begin{array}{l}
 F5\psi(n) := \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\
 \text{roots} \leftarrow \text{polyroots}(\text{data}) \\
 \text{roots}_5
 \end{array}$$

Подходит root4- набор наименьших по модулю действительных корней.

$$\text{tg}\psi d2 := \text{root4}$$

Половина искомого угла ψ , рад

$$\psi d2 := \text{atan}(\text{tg}\psi d2)$$

Переводя в градусы

$$\psi d2\text{deg} := \frac{180}{\pi} \psi d2$$

Окончательно, вектор углов поворота для попадания на датчик ψ , рад

$$\psi := 2 \cdot \psi d2$$

Переводя в градусы

$$\psi\text{deg} := 2 \cdot \psi d2\text{deg}$$

Координаты точки отражения

$$x_1 := s \Sigma_i \cdot \cos(\psi_i) - t \Sigma_i \cdot \sin(\psi_i)$$

$$y_1 := (s \Sigma_i)_i \cdot \sin(\psi_i) + t \Sigma_i \cdot \cos(\psi_i)$$

Длина отражённого луча

$$L_i := \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Положение квазифокуса

$$\Delta f_i := \text{if} \left[i \leq (Np1 - 1) \vee i \geq (N - Np2), \frac{\sqrt{2}}{4 \cdot k_1}, 0 \right]$$

Относительная интенсивность отражённого луча

$$I_{r_i} := \text{if} \left[i \leq (Np1 - 1) \vee i \geq (N - Np2), \frac{\Delta f_i}{\Delta f_i + L_i}, 1 \right]$$

Относительная интенсивность сигнала датчика

$$I_i := I_{r_i} \cdot \sin^2[\phi_0 \Sigma_i + \psi_i]$$

$$n0 := 0..2 \quad \psi_{n0} := 5 \cdot n0$$

$$I0_{n0} := I_c$$

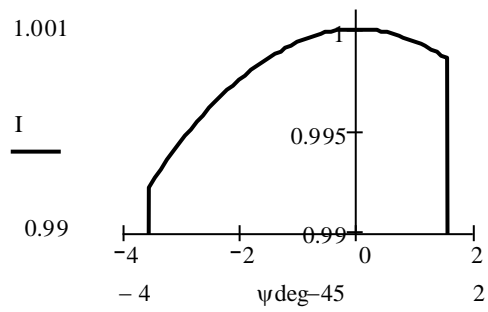
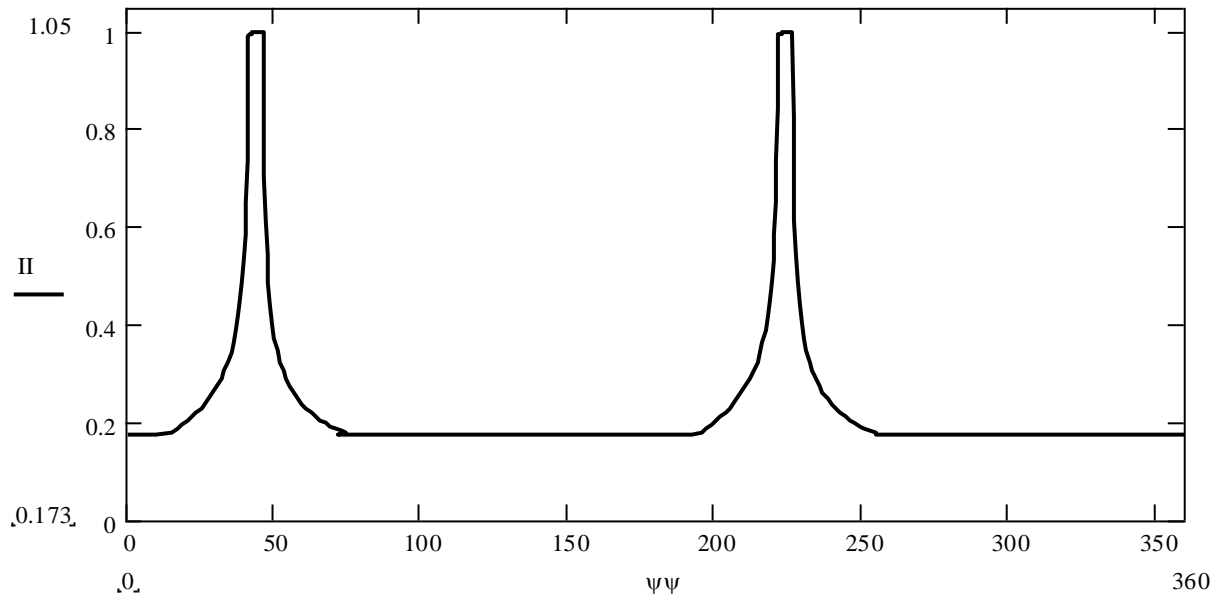
$$\psi_1 := \psi \text{deg} \quad I1 := I$$

$$n2 := 0..24 \quad \psi_{2n2} := 5 \cdot n2 + 72 \quad I2_{n2} := I_c$$

$$\psi_3 := \psi_1 + 180 \quad I3 := I1$$

$$n4 := 0..21 \quad \psi_{4n4} := 5 \cdot n4 + 255 \quad I4_{n4} := I_c$$

$$\psi\psi := \text{stack}(\psi_0, \psi_1, \psi_2, \psi_3, \psi_4) \quad \Pi := \text{stack}(I0, I1, I2, I3, I4)$$



ПРИЛОЖЕНИЕ 1.2

отражение от грани без переходных областей

Размеры кристалла и области граниРадиус кристалла, мм $R := 5$ Ширина грани $2a$, мм $a := 1.3$ Половина центрального угла, опирающегося на грань, рад $\phi_a := \arcsin\left(\frac{a}{R}\right)$ $\phi_a = 0.263$ Переводя в градусы $\phi_{a\text{deg}} := \frac{180}{\pi}\phi_a$ $\phi_{a\text{deg}} = 15.070062$ Расстояние от центра до грани r_a , мм $r_a := \frac{a}{\tan(\phi_a)}$ $r_a = 4.828$ Координаты датчика, мм $x_0 := \frac{R}{\sqrt{2}}$ $x_0 = 3.536$ $y_0 := 12$ Квазифокус при отражении от цилиндра, мм $\Delta f_c := \frac{R \cdot \sqrt{2}}{4}$ $\Delta f_c = 1.768$ Интенсивность при отражении от цилиндра $I_c := \frac{\Delta f_c}{y_0 - \Delta f_c}$ $I_c = 0.172765$ **Плоская грань**Число точек $N_f := 53$ $if := 0..N_f - 1$ Шаг по оси t , мм $dl := 2 \cdot \frac{a}{N_f - 1}$ $dl = 0.05$ Векторы декартовых (s и t , мм) и полярных (ρ_{st} , мм, θ , рад, θ_{deg} , град) координат

(Включая точки сопряжения с окружностью.)

$$\rho_{stf} := \sqrt{sf^2 + tf^2}$$

$$\theta_{f_{if}} := \operatorname{atan}\left(\frac{tf_{if}}{sf_{if}}\right)$$

$$sf_{if} := r_a$$

$$tf_{if} := \left(\frac{N_f - 1}{2} - if\right) \cdot dl$$

$$\theta_{\text{deg}f} := \frac{180}{\pi}\theta_f$$

Вектор нормали к грани с осью s рад и град

$$\phi_{0f_{if}} := 0$$

$$\phi_{0f\text{deg}_{if}} := 0$$

Векторы кривизны и положений квазифокуса (задаются условно, не используются)

$$kf_{if} := 0.1$$

$$\Delta ff_{if} := 0$$

Формирование суммарных массивов

Выполняется условно, чтобы не менять обозначений в дальнейших вычислениях

Общее число точек $N := Nf$ $N = 53$ $i := 0.. N - 1$ $z_i := i$

Векторы декартовых (s и t, мм) и полярных (ρst, мм, θ, рад, θdeg, град) координат

$s\Sigma := sf$ $t\Sigma := tf$

$\rho st\Sigma := \rho stf$ $\theta\Sigma := \theta f$ $\theta deg\Sigma := \theta degf$

Вектор нормалей к суммарной кривой с осью s рад и град

$\phi 0\Sigma := \phi 0f$ $\phi 0\Sigma deg := \frac{180}{\pi} \cdot \phi 0\Sigma$

Вектор кривизны, 1/мм $k := kf$

Вектор положений квазифокуса, мм $\Delta f\Sigma := \Delta ff$

Решение системы уравнений для определения угла ψ

Вычисляем коэффициенты $a1 := t\Sigma - y0$ $a2 := t\Sigma + y0$ $c1_i := \tan(\phi 0\Sigma_i)$
 $b1 := 1 + c1$ $b2 := 1 - c1$
 $c2 := s\Sigma - x0$ $c3 := s\Sigma + x0$

$$A1_i := -a2_i \cdot b1_i \cdot b2_i$$

$$B1_i := -2 \cdot \left[a2_i \cdot (b1_i)^2 - (a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b2_i \right]$$

$$C1_i := a2_i \cdot b1_i \cdot b2_i + 4 \cdot (a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b1_i + (a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b2_i$$

$$D1_i := -2 \cdot \left[(a2_i \cdot b2_i + s\Sigma_i \cdot b1_i) \cdot b2_i - (a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b1_i + (s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b2_i \right]$$

$$E1_i := - \left[(a2_i \cdot b1_i - 4 \cdot s\Sigma_i \cdot b2_i + a1_i \cdot b1_i) \cdot b2_i + 4 \cdot (s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b1_i + a1_i \cdot b1_i \cdot b2_i \right]$$

$$F1_i := 2 \cdot \left[(s\Sigma_i \cdot b1_i + a1_i \cdot b2_i) \cdot b2_i - a1_i \cdot (b1_i)^2 \right]$$

$$G1_i := a1_i \cdot b1_i \cdot b2_i$$

$$A2_i := -2 \cdot c1_i \cdot c3_i$$

$$B2_i := -4 \cdot \left[c1_i \cdot t\Sigma_i + \left[(c1_i)^2 - 1 \right] \cdot c3_i \right]$$

$$C2_i := 2 \cdot \left[c1_i \cdot c2_i - 4 \cdot \left[(c1_i)^2 - 1 \right] \cdot t\Sigma_i + 6 \cdot c1_i \cdot c3_i \right]$$

$$D2_i := 4 \cdot \left[\left[(c1_i)^2 - 1 \right] \cdot c2_i + 6 \cdot c1_i \cdot t\Sigma_i - \left[1 - (c1_i)^2 \right] \cdot c3_i \right]$$

$$E2_i := -2 \cdot \left[6 \cdot c1_i \cdot c2_i + 4 \cdot t\Sigma_i \cdot \left[1 - (c1_i)^2 \right] + c1_i \cdot c3_i \right]$$

$$F2_i := 4 \cdot \left[\left[1 - (c1_i)^2 \right] \cdot c2_i - t\Sigma_i \cdot c1_i \right]$$

$$G2_i := 2 \cdot c1_i \cdot c2_i$$

$$A := A1 - A2 \quad B := B1 - B2 \quad C := C1 - C2 \quad D := D1 - D2$$

$$E := E1 - E2 \quad F := F1 - F2 \quad G := G1 - G2$$

Решением системы является вектор ангенсов половинного угла $\text{tg}\psi/2$
 Но полином 6-й степени имеет 6 корней. Обозначим массивы корней
 $\text{root}_0 \dots \text{root}_5$.

$$\begin{array}{l}
 F0\psi(n) := \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_0 \end{array} \right. \quad \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_1 \end{array} \right.
 \end{array}$$

$$\begin{array}{l}
 F2\psi(n) := \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_2 \end{array} \right. \quad \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_3 \end{array} \right.
 \end{array}$$

$$\begin{array}{l}
 F4\psi(n) := \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_4 \end{array} \right. \quad \left| \begin{array}{l} \text{data} \leftarrow \begin{pmatrix} G_n \\ F_n \\ E_n \\ D_n \\ C_n \\ B_n \\ A_n \end{pmatrix} \\ \text{roots} \leftarrow \text{polyroots}(\text{data}) \\ \text{roots}_5 \end{array} \right.
 \end{array}$$

Подходит root4- набор наименьших по модулю действительных корней.

$$\text{tg}\psi d2 := \text{root4}$$

Половина искомого угла ψ , рад

$$\psi d2 := \text{atan}(\text{tg}\psi d2) \quad \text{Переводя в градусы} \quad \psi d2 \text{deg} := \frac{180}{\pi} \psi d2$$

Окончательно, вектор углов поворота для попадания на датчик ψ , рад $\psi := 2 \cdot \psi d2$

Переводя в градусы $\psi \text{deg} := 2 \cdot \psi d2 \text{deg}$

Определение точек отражения и расчёт интенсивности сигнала латчика

Координаты точки отражения

$$x_i := s\Sigma_i \cdot \cos(\psi_i) - t\Sigma_i \cdot \sin(\psi_i) \quad y_i := (s\Sigma_i) \cdot \sin(\psi_i) + t\Sigma_i \cdot \cos(\psi_i)$$

Построение суммарного графика интенсивности

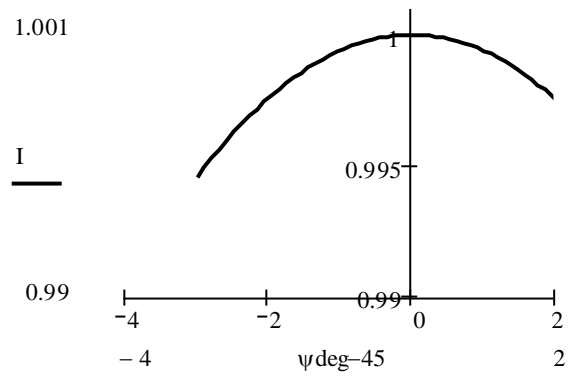
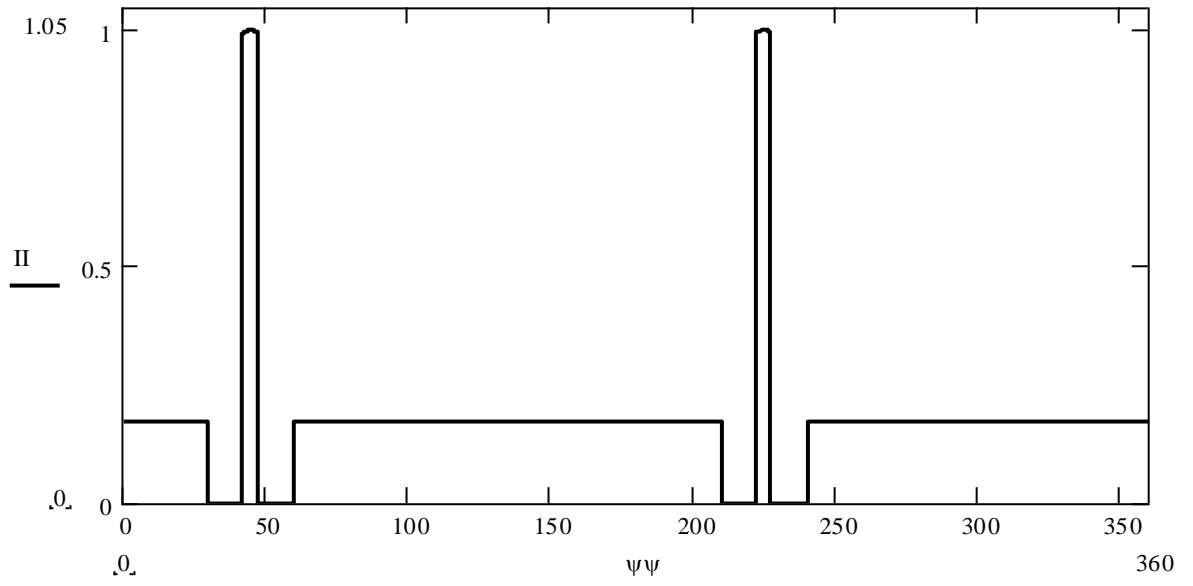
Граничные точки участков

Цилиндр в начале - первый провал	$45 - \phi \text{deg} = 29.93$
Первый провал - первая грань	$\psi \text{deg}_0 = 42.008$
Первая грань - второй провал	$\psi \text{deg}_{1\text{ast}}(\psi \text{deg}) = 47.146$
Второй провал - цилиндр между провалами	$45 + \phi \text{deg} = 60.07$
Цилиндр между провалами - третий провал	$225 - \phi \text{deg} = 209.93$
Третий провал - вторая грань	$\psi \text{deg}_0 + 180 = 222.008$
Вторая грань - четвёртый провал	$\psi \text{deg}_{1\text{ast}}(\psi \text{deg}) + 180 = 227.146$
Четвёртый провал - цилиндр в конце	$225 + \phi \text{deg} = 240.07$

$N0 := 7$	$n0 := 0.. N0 - 1$	$\psi_{0n0} := 5 \cdot n0$	$\psi_{01\text{ast}}(\psi_0) := 45 - \phi \text{deg}$	$I_{n0} := I_c$	
$N1 := 5$	$n1 := 0.. N1 - 1$	$\psi_{1n1} := 5 \cdot n1 + 25$	$\psi_{10} := 45 - \phi \text{deg}$	$\psi_{11\text{ast}}(\psi_1) := \psi \text{deg}(0)$	$I_{n1} := 0$
$N2 := N$		$\psi_2 := \psi \text{deg}$			$I_2 := I$
$N3 := 5$	$n3 := 0.. N3 - 1$	$\psi_{3n3} := 5 \cdot n3 + 45$	$\psi_{30} := \psi_{21\text{ast}}(\psi_2)$	$\psi_{31\text{ast}}(\psi_3) := \phi \text{deg} + 45$	$I_{n3} := 0$
$N4 := 31$	$n4 := 0.. N4 - 1$	$\psi_{4n4} := 5 \cdot n4 + 60$	$\psi_{40} := \phi \text{deg} + 45$	$\psi_{41\text{ast}}(\psi_4) := 225 - \phi \text{deg}$	$I_{n4} := I_c$
$N5 := 5$	$n5 := 0.. N5 - 1$	$\psi_{5n5} := 5 \cdot n5 + 205$	$\psi_{50} := 225 - \phi \text{deg}$	$\psi_{51\text{ast}}(\psi_5) := \psi \text{deg}_0 + 180$	$I_{n5} := 0$
$N6 := N$		$\psi_6 := \psi \text{deg} + 180$			$I_6 := I$
$N7 := 5$	$n7 := 0.. N7 - 1$	$\psi_{7n7} := 5 \cdot n7 + 225$	$\psi_{70} := \psi_{61\text{ast}}(\psi_6)$	$\psi_{71\text{ast}}(\psi_7) := 225 + \phi \text{deg}$	$I_{n7} := 0$
$N8 := 25$	$n8 := 0.. N8 - 1$	$\psi_{8n8} := 5 \cdot n8 + 240$	$\psi_{80} := 225 + \phi \text{deg}$		$I_{n8} := I_c$

$\psi\psi := \text{stack}(\psi_0, \psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7, \psi_8)$

$\Pi := \text{stack}(I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8)$

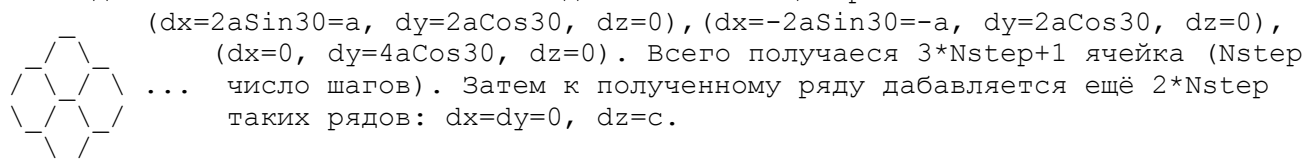


ПРИЛОЖЕНИЕ 2

текст программы для расчета поверхностных энергий кристаллов сапфира методом
Разрыва связей

Построение больших блоков решётки сапфира

{*****}
На каждом шаге к начальной ячейке добавляются ещё три:



{*****}

```
unit MainU;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Al2O3U, StdCtrls;
type
  TMainForm = class(TForm)
    MainPageControl: TPageControl;
    StartDataTabSheet: TTabSheet;
    Rombohedrons_1row_TabSheet: TTabSheet;
    AlStartDataLabel: TLabel;
    AlStartDataMemo: TMemo;
    OxigenStartDataLabel: TLabel;
    OxigenStartDataMemo: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    N_CellsToAddEdit: TEdit;
    FormRowButton: TButton;
    Al_1cellLabel: TLabel;
    Oxigen_1cellLabel: TLabel;
    Al_1row_Build_Memo: TMemo;
    Oxigen_1row_Build_Memo: TMemo;
    N_Al_1row_Label: TLabel;
    N_Oxigen_1row_Label: TLabel;
    Label3: TLabel;
    Al_1row_Build_Nil_Memo: TMemo;
    N_Al_1row_Clean_Label: TLabel;
    Al_1row_Clean_Memo: TMemo;
    Al_1row_Cuttet_Memo: TMemo;
    N_Oxigen_1row_Cuttet_Label: TLabel;
    Oxigen_1row_Cuttet_Memo: TMemo;
    N_Al_1row_Cuttet_Label: TLabel;
    Save_1row_Button: TButton;
    Rombohedrons_Block_TabSheet: TTabSheet;
    Label4: TLabel;
    Label5: TLabel;
    N_RowsToAddEdit: TEdit;
    FormBlockButton: TButton;
    Al_Block_Build_Memo: TMemo;
    Al_Block_Build_Nil_Memo: TMemo;
    Al_Block_Clean_Memo: TMemo;
    Oxigen_Block_Build_Memo: TMemo;
    N_Al_Block_Label: TLabel;
```

```
Label7: TLabel;
N_Al_Block_Clean_Label: TLabel;
N_Oxygen_Block_Label: TLabel;
Save_Block_Button: TButton;
Pyramids_1row_TabSheet1: TTabSheet;
Label6: TLabel;
Label8: TLabel;
N_CellsToAdd_P_Edit: TEdit;
FormRow_P_Button: TButton;
Save_1row_P_Button: TButton;
N_Al_1row_P_Label: TLabel;
Al_1row_Build_P_Memo: TMemo;
Label10: TLabel;
Al_1row_Build_Nil_P_Memo: TMemo;
N_Al_1row_Clean_P_Label: TLabel;
Al_1row_Clean_P_Memo: TMemo;
N_Oxygen_1row_P_Label: TLabel;
Oxygen_1row_Build_P_Memo: TMemo;
N_Al_1row_Cuttet_P_Label: TLabel;
Al_1row_Cuttet_P_Memo: TMemo;
N_Oxygen_1row_Cuttet_P_Label: TLabel;
Oxygen_1row_Cuttet_P_Memo: TMemo;
Pyramids_Block_TabSheet: TTabSheet;
Label9: TLabel;
Label11: TLabel;
N_RowsToAdd_P_Edit: TEdit;
FormBlock_P_Button: TButton;
Save_Block_P_Button: TButton;
N_Al_Block_P_Label: TLabel;
Label13: TLabel;
N_Al_Block_Clean_P_Label: TLabel;
N_Oxygen_Block_P_Label: TLabel;
Oxygen_Block_Build_P_Memo: TMemo;
Al_Block_Clean_P_Memo: TMemo;
Al_Block_Build_Nil_P_Memo: TMemo;
Al_Block_Build_P_Memo: TMemo;
Prisms_1row_TabSheet: TTabSheet;
Prisms_Block_TabSheet: TTabSheet;
Label12: TLabel;
Label14: TLabel;
N_CellsToAdd_Pr_Edit: TEdit;
FormRow_Pr_Button: TButton;
Save_1row_Pr_Button: TButton;
N_Al_1row_Pr_Label: TLabel;
Label16: TLabel;
N_Al_1row_Clean_Pr_Label: TLabel;
N_Oxygen_1row_Pr_Label: TLabel;
Al_1row_Build_Pr_Memo: TMemo;
Al_1row_Build_Nil_Pr_Memo: TMemo;
Al_1row_Clean_Pr_Memo: TMemo;
Oxygen_1row_Build_Pr_Memo: TMemo;
Label15: TLabel;
Label17: TLabel;
N_RowsToAdd_Pr_Edit: TEdit;
FormBlock_Pr_Button: TButton;
Save_Block_Pr_Button: TButton;
N_Al_Block_Pr_Label: TLabel;
Label19: TLabel;
N_Al_Block_Clean_Pr_Label: TLabel;
N_Oxygen_Block_Pr_Label: TLabel;
```

```

Al_Block_Build_Pr_Memo: TMemo;
Al_Block_Build_Nil_Pr_Memo: TMemo;
Al_Block_Clean_Pr_Memo: TMemo;
Oxigen_Block_Build_Pr_Memo: TMemo;
N_Al_Block_Cuttet_Pr_Label: TLabel;
Al_Block_Cuttet_Pr_Memo: TMemo;
N_Oxigen_Block_Cuttet_Pr_Label: TLabel;
Oxigen_Block_Cuttet_Pr_Memo: TMemo;
MoveBloskButton: TButton;
MoveBlosk_P_Button: TButton;
MoveBlosk_Pr_Button: TButton;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormRowButtonClick(Sender: TObject);
procedure Save_lrow_ButtonClick(Sender: TObject);
procedure FormBlockButtonClick(Sender: TObject);
procedure Save_Block_ButtonClick(Sender: TObject);
procedure FormRow_P_ButtonClick(Sender: TObject);
procedure Save_lrow_P_ButtonClick(Sender: TObject);
procedure FormBlock_P_ButtonClick(Sender: TObject);
procedure Save_Block_P_ButtonClick(Sender: TObject);
procedure FormRow_Pr_ButtonClick(Sender: TObject);
procedure Save_lrow_Pr_ButtonClick(Sender: TObject);
procedure FormBlock_Pr_ButtonClick(Sender: TObject);
procedure Save_Block_Pr_ButtonClick(Sender: TObject);
procedure MoveBloskButtonClick(Sender: TObject);
procedure MoveBlosk_P_ButtonClick(Sender: TObject);
procedure MoveBlosk_Pr_ButtonClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  MainForm: TMainForm;
{-----}
{Списки атомов и файлы исходных списков и результатов}
{-----}
type PAtomData = ^TAtomData; {тип данных для записей в списках атомов}
  TAtomData = record
    X,Y,Z: Double;
  end; {TAtomData = record}
var Al_lcell_List, Oxigen_lcell_List: TList; {списки исходных данных атомов Al
и O в пределах одной ячейки}
var Al_lrow_Build_List, Oxigen_lrow_Build_List: TList; {списки атомов Al и O в
пределах одного горизонтального ряда после добавления ячеек; указатели списков
хранят ссылки на все атомы ряда TAtomData; при повтором построении и при
выходе из программы эти ссылки нужно освобождать}
var Al_lrow_Clean_List, Oxigen_lrow_Clean_List: TList; {списки атомов Al и O в
пределах одного горизонтального ряда после удаления повторяющихся атомов; в
начале в них полностью копируются указатели из предыдущих списков, затем
указатели на повторяющиеся атомы заменяются на nil, и за тем проходом с конца
списка записи с nil удаляются; при повторном построении и выходе из программы
нужно освобождать только списки, но не ссылки на атомы (это дубли указателей)}
var Al_lrow_Cuttet_List, Oxigen_lrow_Cuttet_List: TList; {списки атомов Al и O
в пределах одного горизонтального ряда после обрезания до прямоугольника;
обрабатываются как и предыдущие списки}
var Al_Block_Build_List, Oxigen_Block_Build_List: TList; {списки атомов Al и O
в всего блока после добавления горизогтальных рядов; указатели списков хранят

```

```

ссылки на все атомы ряда TAtomData; при повтором построении и при выходе из
программы эти ссылки нужно освобождать}
var Al_Block_Clean_List, Oxigen_Block_Clean_List: TList; {списки атомов Al и O
в пределах всего блока после удаления повторяющихся атомов; в начале в них
полностью копируются указатели из предыдущих списков, затем указатели на
повторяющиеся атомы заменяются на nil, и за тем проходом с конца списка записи
с nil удаляются; при повторном построении и выходе из программы нужно
освобождать только списки, но не ссылки на атомы (это дубли указателей)}
var Al_Block_Cuttet_List, Oxigen_Block_Cuttet_List: TList; {списки атомов Al и
O в пределах блока после обрезания до прямоугольника; обрабатываются как и
предыдущие списки}
const Al_1cell_ShortFileName = 'Al_1cell.txt'; {короткие имена файлов}
      Oxigen_1cell_ShortFileName = 'Oxigen_1cell.txt';
      Al_1row_ShortFileName = 'Al_1row.txt';
      Oxigen_1row_ShortFileName = 'Oxigen_1row.txt';
      Al_Block_ShortFileName = 'Al_Block.txt';
      Oxigen_Block_ShortFileName = 'Oxigen_Block.txt';
var Al_1cell_FileName, Oxigen_1cell_FileName,
      Al_1row_FileName, Oxigen_1row_FileName,
      Al_Block_FileName, Oxigen_Block_FileName: String; {полные имена файлов}
      Al_1cell_File, Oxigen_1cell_File,
      Al_1row_File, Oxigen_1row_File,
      Al_Block_File, Oxigen_Block_File: TextFile; {файловые переменные}
      N_Al_1cell, N_Oxigen_1cell: Integer; {числа атомов AL и O в одной ячейке}
      N_CellsToAdd: Integer; {число троек ячеек, добавляемых в 1 горизонтальный
ряд}
      N_RowsToAdd: Integer; {число добавляемых рядов}
const YStep = 2*a*cR3; {шаг по Y при добавлении группы из 3 ячеек}
var XNew, YNew: Double; {координаты X и Y центра добавляемой ячейки}
      BlockLeft, BlockRight: Double; {размеры блока по Y}
      BlockFront, BlockBack: Double; {размеры блока по X}
{*****}
implementation
{$R *.dfm}
{=====}
      {Вспомогательные функции}
{=====}
function AtomsCoincide(Atom1, Atom2: PAtomData): Boolean; {даёт True если атомы
совпадают и ни один из них не nil}
begin
  if (Atom1 = nil) or (Atom2 = nil) then begin Result := False; Exit; end;
  if (Abs(Atom1.X-Atom2.X)<0.0001) and (Abs(Atom1.Y-Atom2.Y)<0.0001)
    and (Abs(Atom1.Z-Atom2.Z)<0.0001) then
    Result := True
  else Result := False;
end; {function AtomsCoincide}
{=====}
      {Процедуры создания и удаления формы}
{=====}
procedure TMainForm.FormCreate(Sender: TObject);
var PAtom: PAtomData;
begin
  Al_1cell_List := TList.Create; Oxigen_1cell_List := TList.Create;
  Al_1row_Build_List := TList.Create; Oxigen_1row_Build_List :=
TList.Create;
  Al_1row_Clean_List := TList.Create; Oxigen_1row_Clean_List :=
TList.Create;
  Al_1row_Cuttet_List := TList.Create; Oxigen_1row_Cuttet_List
:=TList.Create;

```

```

    Al_Block_Build_List := TList.Create; Oxigen_Block_Build_List
:=TList.Create;
    Al_Block_Clean_List := TList.Create; Oxigen_Block_Clean_List
:=TList.Create;
    Al_Block_Cuttet_List :=TList.Create;
Oxigen_Block_Cuttet_List:=TList.Create;
    {Загружаем файлы исходных данных об атомах Al и O в пределах ячейки}
    Al_1cell_FileName := ExtractFilePath(Application.ExeName)
                        +Al_1cell_ShortFileName;
    AssignFile(Al_1cell_File,Al_1cell_FileName);
    try
        Reset(Al_1cell_File);
        while not Eof(Al_1cell_File) do
            begin
                New(PAtom);
                ReadLn(Al_1cell_File,PAtom^.X,PAtom^.Y,PAtom^.Z);
                Al_1cell_List.Add(PAtom);
                AlStartDataMemo.Lines.Add(FloatToStr(PAtom^.X)
                    +' '+FloatToStr(PAtom^.Y)+' '+
FloatToStr(PAtom^.Z));
            end; {while not Eof(Al_1cell_File) do}
        finally CloseFile(Al_1cell_File);
        end; {try ... finally ...}
    N_Al_1cell := Al_1cell_List.Count;
    AlStartDataLabel.Caption := AlStartDataLabel.Caption+'. Файл '
        +Al_1cell_ShortFileName+' загружен в список Al_1cell_List';
    Al_1cellLabel.Caption := Al_1cellLabel.Caption+IntToStr(N_Al_1cell);
    Oxigen_1cell_FileName := ExtractFilePath(Application.ExeName)
                        +Oxigen_1cell_ShortFileName;
    AssignFile(Oxigen_1cell_File,Oxigen_1cell_FileName);
    try
        Reset(Oxigen_1cell_File);
        while not Eof(Oxigen_1cell_File) do
            begin
                New(PAtom);
                ReadLn(Oxigen_1cell_File,PAtom^.X,PAtom^.Y,PAtom^.Z);
                Oxigen_1cell_List.Add(PAtom);
                OxigenStartDataMemo.Lines.Add(FloatToStr(PAtom^.X)
                    +' '+FloatToStr(PAtom^.Y)+' '+
FloatToStr(PAtom^.Z));
            end; {while not Eof(Oxigen_1cell_File) do}
        finally CloseFile(Oxigen_1cell_File);
        end; {try ... finally ...}
    N_Oxigen_1cell := Oxigen_1cell_List.Count;
    OxigenStartDataLabel.Caption := OxigenStartDataLabel.Caption+'. Файл '
        +Oxigen_1cell_ShortFileName+' загружен в список Oxigen_1cell_List';
    Oxigen_1cellLabel.Caption := Oxigen_1cellLabel.Caption
        +IntToStr(N_Oxigen_1cell);
    end; {procedure TMainForm.FormCreate}
}-----}
procedure TMainForm.FormDestroy(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
    for I := 0 to Al_1cell_List.Count-1 do
        begin
            PAtom := Al_1cell_List[I];
            Dispose(PAtom);
        end; {for I := 0 to Al_1cell_List.Count-1 do ...}
    Al_1cell_List.Free;

```

```

for I := 0 to Oxigen_1cell_List.Count-1 do
begin
  PAtom := Oxigen_1cell_List[I];
  Dispose(PAtom);
end; {for I := 0 to Oxigen_1cell_List.Count-1 do ...}
Oxigen_1cell_List.Free;
{-----}
for I := 0 to Al_1row_Build_List.Count-1 do
begin
  PAtom := Al_1row_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
Al_1row_Build_List.Free;
for I := 0 to Oxigen_1row_Build_List.Count-1 do
begin
  PAtom := Oxigen_1row_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Oxigen_1row_Build_List.Count-1 do ...}
Oxigen_1row_Build_List.Free;
{-----}
Al_1row_Clean_List.Free;
Oxigen_1row_Clean_List.Free;
Al_1row_Cuttet_List.Free;
Oxigen_1row_Cuttet_List.Free;
{-----}
for I := 0 to Al_Block_Build_List.Count-1 do
begin
  PAtom := Al_Block_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
Al_Block_Build_List.Free;
for I := 0 to Oxigen_Block_Build_List.Count-1 do
begin
  PAtom := Oxigen_Block_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Oxigen_Block_Build_List.Count-1 do ...}
Oxigen_Block_Build_List.Free;
{-----}
Al_Block_Clean_List.Free;
Oxigen_Block_Clean_List.Free;
Al_Block_Cuttet_List.Free;
Oxigen_Block_Cuttet_List.Free;
end;
{=====}
  {Обработчики событий элементов управления. Страница Ромбоэдры 1 ряд.}
{=====}
procedure TMainForm.FormRowButtonClick(Sender: TObject);
var I,J: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  {Очищаем списки(они могли быть ранее использованы)}
  for I := 0 to Al_1row_Build_List.Count-1 do
  begin
    PAtom := Al_1row_Build_List[I];
    Dispose(PAtom);
  end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
  Al_1row_Build_List.Clear;
  for I := 0 to Oxigen_1row_Build_List.Count-1 do
  begin

```

```

    PAtom := Oxigen_1row_Build_List[I];
    Dispose(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
Oxigen_1row_Build_List.Clear;
Al_1row_Clean_List.Clear;
Oxigen_1row_Clean_List.Clear;
Al_1row_Cuttet_List.Clear;
Oxigen_1row_Cuttet_List.Clear;
Al_1row_Build_Memo.Clear;
Al_1row_Build_Nil_Memo.Clear;
Al_1row_Clean_Memo.Clear;
Oxigen_1row_Build_Memo.Clear;
Al_1row_Cuttet_Memo.Clear;
Oxigen_1row_Cuttet_Memo.Clear;
N_CellsToAdd := StrToInt(N_CellsToAddEdit.Text); {сколько раз повторять
добавление троек ячеек в горизонтальный ряд}
    {Копируем первую ячейку}
for J := 0 to N_Al_1cell-1 do
begin
    New(PAtom);
    PAtom^.X := PAtomData(Al_1cell_List[J]).X;
    PAtom^.Y := PAtomData(Al_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
    Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
    New(PAtom);
    PAtom^.X := PAtomData(Oxigen_1cell_List[J]).X;
    PAtom^.Y := PAtomData(Oxigen_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
    Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
    {Цикл добавления троек ячеек}
for I := 0 to N_CellsToAdd -1 do
begin
    YNew := YStep*(I+0.5); {первая из трёх ячеек}
    XNew := -a;
    for J := 0 to N_Al_1cell-1 do
begin
        New(PAtom);
        PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
        PAtom^.Y := YNew+PAtomData(Al_1cell_List[J]).Y;
        PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
        Al_1row_Build_List.Add(PAtom);
    end; {for J = 0 to N_Al_1cell-1 do ...}
    for J := 0 to N_Oxigen_1cell-1 do
begin
        New(PAtom);
        PAtom^.X := XNew+PAtomData(Oxigen_1cell_List[J]).X;
        PAtom^.Y := YNew+PAtomData(Oxigen_1cell_List[J]).Y;
        PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
        Oxigen_1row_Build_List.Add(PAtom);
    end; {for J = 0 to N_Al_1cell-1 do ...}
    XNew := a; {вторая из трёх ячеек}
    for J := 0 to N_Al_1cell-1 do
begin
        New(PAtom);
        PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
        PAtom^.Y := YNew+PAtomData(Al_1cell_List[J]).Y;

```

```

    PAtom^.Z := PAtomData(Al_1cell_List[J])^.Z;
    Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxygen_1cell-1 do
begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Oxygen_1cell_List[J])^.X;
    PAtom^.Y := YNew+PAtomData(Oxygen_1cell_List[J])^.Y;
    PAtom^.Z := PAtomData(Oxygen_1cell_List[J])^.Z;
    Oxygen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
YNew := YStep*(I+1); {третья из трёх ячеек}
XNew := 0;
for J := 0 to N_Al_1cell-1 do
begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Al_1cell_List[J])^.X;
    PAtom^.Y := YNew+PAtomData(Al_1cell_List[J])^.Y;
    PAtom^.Z := PAtomData(Al_1cell_List[J])^.Z;
    Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxygen_1cell-1 do
begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Oxygen_1cell_List[J])^.X;
    PAtom^.Y := YNew+PAtomData(Oxygen_1cell_List[J])^.Y;
    PAtom^.Z := PAtomData(Oxygen_1cell_List[J])^.Z;
    Oxygen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
end; {for I := 0 to N_CellsToAdd -1 do ...}
for I := 0 to Al_1row_Build_List.Count-1 do {Выводим результаты}
Al_1row_Build_Memo.Lines.Add(IntToStr(I)+' '
+FloatToStr(PAtomData(Al_1row_Build_List[I])^.X)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Y)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Z));
N_Al_1row_Label.Caption := 'N_Al_1row '
+IntToStr(Al_1row_Build_List.Count);
for I := 0 to Oxygen_1row_Build_List.Count-1 do
Oxygen_1row_Build_Memo.Lines.Add(IntToStr(I)+' '
+ FloatToStr(PAtomData(Oxygen_1row_Build_List[I])^.X)+' '
+ FloatToStr(PAtomData(Oxygen_1row_Build_List[I])^.Y)+' '
+ FloatToStr(PAtomData(Oxygen_1row_Build_List[I])^.Z));
N_Oxygen_1row_Label.Caption := 'N_Oxygen_1row '
+IntToStr(Oxygen_1row_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_1row_Build_List.Count-1 do
begin
    PAtom := Al_1row_Build_List[I];
    Al_1row_Clean_List.Add(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
for I := 0 to Oxygen_1row_Build_List.Count-1 do
begin
    PAtom := Oxygen_1row_Build_List[I];
    Oxygen_1row_Clean_List.Add(PAtom);
end; {for I := 0 to Oxygen_1row_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_1row_Clean_List.Count-2 do
for J := I+1 to Al_1row_Clean_List.Count-1 do
if AtomsCoincide(PAtomData(Al_1row_Clean_List[I]),

```



```

        PAtomData(Al_1row_Clean_List[J])) then
        Al_1row_Clean_List[J] := nil;
Al_1row_Build_Nil_Memo.Clear; {выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
begin
    if Al_1row_Clean_List[I] = nil then
        Al_1row_Build_Nil_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Build_Nil_Memo.Lines.Add(IntToStr(I)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
{!!! Атомы O повторяются не могут. Нечего и проверять !!!}
{Удаляем указатели с nil из списка}
for I := Al_1row_Clean_List.Count-1 downto 0 do
    if Al_1row_Clean_List[I] = Nil then Al_1row_Clean_List.Delete(I);
Al_1row_Clean_Memo.Clear; {выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
begin
    if Al_1row_Clean_List[I] = nil then
        Al_1row_Clean_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Clean_Memo.Lines.Add(IntToStr(I)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
        +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
N_Al_1row_Clean_Label.Caption := 'N_Al_1row_Clean '
        +IntToStr(Al_1row_Clean_List.Count);
{Обрезаем блок до прямоугольника. По толщине +-a, по ширине от BlockLeft
=
    rOpD2 до BlockRight=N_CellsToAdd*Ystep+rOpD2}
BlockLeft := -1.01*rOpD2;
BlockRight := 1.01*(N_CellsToAdd*Ystep+rOpD2);
BlockFront := 1.01*a;
BlockBack := -1.01*a;
{Заполняем новые списки записями атомов, лежащих внутри прямоугольника}
for I := 0 to Al_1row_Clean_List.Count-1 do
begin
    PAtom := Al_1row_Clean_List[I];
    if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
        (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
        begin
            Al_1row_Cuttet_List.Add(PAtom);
            Al_1row_Cuttet_Memo.Lines.Add(IntToStr(I)+' '
                +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
                +FloatToStr(PAtom^.Z));
        end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
end; {for I := 0 to Al_1row_Clean_List.Count-1 do}
N_Al_1row_Cuttet_Label.Caption := 'Al_1row_Cuttet '
    + IntToStr(Al_1row_Cuttet_List.Count);
for I := 0 to Oxigen_1row_Clean_List.Count-1 do
begin
    PAtom := Oxigen_1row_Clean_List[I];
    if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
        (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
        begin
            Oxigen_1row_Cuttet_List.Add(PAtom);
            Oxigen_1row_Cuttet_Memo.Lines.Add(IntToStr(I)+' '
                +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
                +FloatToStr(PAtom^.Z));
        end;
end;

```

```

        end; {for I := 0 to Oxigen_1row_Clean_List.Count-1 do ...}
    end; {for I := 0 to Oxigen_1row_Clean_List.Count-1 do}
    N_Oxigen_1row_Cuttet_Label.Caption := 'Oxigen_1row_Cuttet '
    + IntToStr(Oxigen_1row_Cuttet_List.Count);
    Screen.Cursor := crDefault;
end; {procedure TMainForm.FormRowButtonClick}
{-----}
procedure TMainForm.Save_1row_ButtonClick(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
    Screen.Cursor := crHourGlass;
    Al_1row_FileName := ExtractFilePath(Application.ExeName)
    +Al_1row_ShortFileName;
    AssignFile(Al_1row_File,Al_1row_FileName);
    try
        Rewrite(Al_1row_File);
        for I := 0 to Al_1row_Cuttet_List.Count-1 do
            begin
                PAtom := Al_1row_Cuttet_List[I];
                WriteLn(Al_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
            end; { for I := 0 to Al_1row_Cuttet_List.Count-1 do ...}
        finally CloseFile(Al_1row_File);
        end; {try ... finally ...}
    Oxigen_1row_FileName := ExtractFilePath(Application.ExeName)
    +Oxigen_1row_ShortFileName;
    AssignFile(Oxigen_1row_File,Oxigen_1row_FileName);
    try
        Rewrite(Oxigen_1row_File);
        for I := 0 to Oxigen_1row_Cuttet_List.Count-1 do
            begin
                PAtom := Oxigen_1row_Cuttet_List[I];
                WriteLn(Oxigen_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
            end; { for I := 0 to Oxigen_1row_Cuttet_List.Count-1 do ...}
        finally CloseFile(Oxigen_1row_File);
        end; {try ... finally ...}
    Screen.Cursor := crDefault;
end; {procedure TMainForm.Save_1row_ButtonClick}
{=====}
    {Обработчики событий элементов управления. Страница Ромбоэдры блок.}
{=====}
procedure TMainForm.FormBlockButtonClick(Sender: TObject);
var I,J: Integer;
    PAtom: PAtomData;
begin
    Screen.Cursor := crHourGlass;
    {Очищаем списки(они могли быть ранее использованы)}
    for I := 0 to Al_Block_Build_List.Count-1 do
        begin
            PAtom := Al_Block_Build_List[I];
            Dispose(PAtom);
        end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
    Al_Block_Build_List.Clear;
    for I := 0 to Oxigen_Block_Build_List.Count-1 do
        begin
            PAtom := Oxigen_Block_Build_List[I];
            Dispose(PAtom);
        end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
    Oxigen_Block_Build_List.Clear;
    Al_Block_Clean_List.Clear;

```

```

Oxygen_Block_Clean_List.Clear;
Al_Block_Build_Memo.Clear;
Al_Block_Build_Nil_Memo.Clear;
Al_Block_Clean_Memo.Clear;
Oxygen_Block_Build_Memo.Clear;
N_RowsToAdd := StrToInt(N_RowsToAddEdit.Text); {сколько рядов добавлять}
    {Копируем первый ряд}
for J := 0 to Al_1row_Cuttet_List.Count-1 do
begin
    New(PAtom);
    PAtom^.X := PAtomData(Al_1row_Cuttet_List[J]).X;
    PAtom^.Y := PAtomData(Al_1row_Cuttet_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1row_Cuttet_List[J]).Z;
    Al_Block_Build_List.Add(PAtom);
end; {for J := 0 to Al_1row_Cuttet_List.Count-1 do ...}
for J := 0 to Oxygen_1row_Cuttet_List.Count-1 do
begin
    New(PAtom);
    PAtom^.X := PAtomData(Oxygen_1row_Cuttet_List[J]).X;
    PAtom^.Y := PAtomData(Oxygen_1row_Cuttet_List[J]).Y;
    PAtom^.Z := PAtomData(Oxygen_1row_Cuttet_List[J]).Z;
    Oxygen_Block_Build_List.Add(PAtom);
end; {for J := 0 to Oxygen_1row_Cuttet_List.Count-1 do ...}
    {Добавляем заданное число рядов}
for I := 1 to N_RowsToAdd do
begin
    for J := 0 to Al_1row_Cuttet_List.Count-1 do
begin
        New(PAtom);
        PAtom^.X := PAtomData(Al_1row_Cuttet_List[J]).X;
        PAtom^.Y := PAtomData(Al_1row_Cuttet_List[J]).Y;
        PAtom^.Z := PAtomData(Al_1row_Cuttet_List[J]).Z+c*I;
        Al_Block_Build_List.Add(PAtom);
    end; {for J := 0 to Al_1row_Cuttet_List.Count-1 do ...}
    for J := 0 to Oxygen_1row_Cuttet_List.Count-1 do
begin
        New(PAtom);
        PAtom^.X := PAtomData(Oxygen_1row_Cuttet_List[J]).X;
        PAtom^.Y := PAtomData(Oxygen_1row_Cuttet_List[J]).Y;
        PAtom^.Z := PAtomData(Oxygen_1row_Cuttet_List[J]).Z+c*I;
        Oxygen_Block_Build_List.Add(PAtom);
    end; {for J := 0 to Oxygen_1row_Cuttet_List.Count-1 do ...}
end; {for I := 1 to N_RowsToAdd do ...}
for I := 0 to Al_Block_Build_List.Count-1 do {Выводим результаты}
    Al_Block_Build_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Build_List[I]).X)+' '
    + FloatToStr(PAtomData(Al_Block_Build_List[I]).Y)+' '
    + FloatToStr(PAtomData(Al_Block_Build_List[I]).Z));
N_Al_Block_Label.Caption := 'N_Al_Block '
    +IntToStr(Al_Block_Build_List.Count);
for I := 0 to Oxygen_Block_Build_List.Count-1 do
    Oxygen_Block_Build_Memo.Lines.Add(IntToStr(I)+' '
    + FloatToStr(PAtomData(Oxygen_Block_Build_List[I]).X)+' '
    + FloatToStr(PAtomData(Oxygen_Block_Build_List[I]).Y)+' '
    + FloatToStr(PAtomData(Oxygen_Block_Build_List[I]).Z));
N_Oxygen_Block_Label.Caption := 'N_Oxygen_Block '
    +IntToStr(Oxygen_Block_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_Block_Build_List.Count-1 do

```

```

begin
  PAtom := Al_Block_Build_List[I];
  Al_Block_Clean_List.Add(PAtom);
end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
for I := 0 to Oxigen_Block_Build_List.Count-1 do
begin
  PAtom := Oxigen_Block_Build_List[I];
  Oxigen_Block_Clean_List.Add(PAtom);
end; {for I := 0 to Oxigen_Block_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_Block_Clean_List.Count-2 do
  for J := I+1 to Al_Block_Clean_List.Count-1 do
    if AtomsCoincide(PAtomData(Al_Block_Clean_List[I]),
      PAtomData(Al_Block_Clean_List[J])) then
      Al_Block_Clean_List[J] := nil;
Al_Block_Build_Nil_Memo.Clear; {выводим результаты}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  if Al_Block_Clean_List[I] = nil then
    Al_Block_Build_Nil_Memo.Lines.Add(IntToStr(I)+' nil')
  else Al_Block_Build_Nil_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
end; {for I := 0 to Al_Block_Clean_List.Count-1 do ...}
{!!! Атомы O повторяться не могут. Нечего и проверять !!!}
{Удаляем указатели с nil из списка}
for I := Al_Block_Clean_List.Count-1 downto 0 do
  if Al_Block_Clean_List[I] = Nil then Al_Block_Clean_List.Delete(I);
Al_Block_Clean_Memo.Clear; {выводим результаты}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  if Al_Block_Clean_List[I] = nil then
    Al_Block_Clean_Memo.Lines.Add(IntToStr(I)+' nil')
  else Al_Block_Clean_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
N_Al_Block_Clean_Label.Caption := 'N_Al_Block_Clean '
    +IntToStr(Al_Block_Clean_List.Count);

Screen.Cursor := crDefault;
end; {procedure TMainForm.FormBlockButtonClick}
{-----}
procedure TMainForm.MoveBloskButtonClick(Sender: TObject);{перемещает начало
  координат блока в середину нижнего левого ребра сдвигая атомы по Y на
  rOpD2}
var I: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  Al_Block_Clean_Memo.Lines.Clear;
  Oxigen_Block_Build_Memo.Lines.Clear;
  for I := 0 to Al_Block_Clean_List.Count-1 do
    begin
      PAtom := Al_Block_Clean_List[I];
      PAtom^.Y := PAtom^.Y+rOpD2;
      Al_Block_Clean_Memo.Lines.Add(IntToStr(I)+' '
        +FloatToStr(PAtom^.X)+' ' +FloatToStr(PAtom^.Y)+' '
        +FloatToStr(PAtom^.Z));
    end;
  end;
end;

```

```

    end; { for I := 0 to .Al_Block_Clean_ListCount-1 do ...}
    N_Al_Block_Clean_Label.Caption :=N_Al_Block_Clean_Label.Caption+'
Shifted';
    for I := 0 to Oxigen_Block_Clean_List.Count-1 do
    begin
        PAtom := Oxigen_Block_Clean_List[I];
        PAtom^.Y := PAtom^.Y+rOpD2;
        Oxigen_Block_Build_Memo.Lines.Add(IntToStr(I)+' '
            + FloatToStr(PAtom^.X)+' ' + FloatToStr(PAtom^.Y)+' '
            + FloatToStr(PAtom^.Z));
        end; { for I := 0 to Oxigen_Block_Clean_List.Count-1 do ...}
    N_Oxigen_Block_Label.Caption :=N_Oxigen_Block_Label.Caption+' Shifted';
    Screen.Cursor := crDefault;
end; {procedure TMainForm.MoveBloskButtonClick}
{-----}
procedure TMainForm.Save_Block_ButtonClick(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
    Screen.Cursor := crHourGlass;
    Al_Block_FileName := ExtractFilePath(Application.ExeName)
        +Al_Block_ShortFileName;
    AssignFile(Al_Block_File,Al_Block_FileName);
    ShowMessage('Al_Block_FileName = '+#13#10+Al_Block_FileName);
    try
        Rewrite(Al_Block_File);
        for I := 0 to Al_Block_Clean_List.Count-1 do
            begin
                PAtom := Al_Block_Clean_List[I];
                WriteLn(Al_Block_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
            end; { for I := 0 to .Al_Block_Clean_ListCount-1 do ...}
        finally CloseFile(Al_Block_File);
    end; {try ... finally ...}
    Oxigen_Block_FileName := ExtractFilePath(Application.ExeName)
        +Oxigen_Block_ShortFileName;
    AssignFile(Oxigen_Block_File,Oxigen_Block_FileName);
    ShowMessage('Oxigen_Block_FileName = '+#13#10+Oxigen_Block_FileName);
    try
        Rewrite(Oxigen_Block_File);
        for I := 0 to Oxigen_Block_Clean_List.Count-1 do
            begin
                PAtom := Oxigen_Block_Clean_List[I];
                WriteLn(Oxigen_Block_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
            end; { for I := 0 to Oxigen_Block_Clean_List.Count-1 do ...}
        finally CloseFile(Oxigen_Block_File);
    end; {try ... finally ...}
    Screen.Cursor := crDefault;
end; {procedure TMainForm.Save_Block_ButtonClick}
{=====}
    {Обработчики событий элементов управления. Страница Пирамиды 1 ряд.}
{=====}
procedure TMainForm.FormRow_P_ButtonClick(Sender: TObject); {в процедуре
используются те же списки, что и в TMainForm.FormRowButtonClick при
построении ряда для блока ромбоэдров}
var I,J: Integer;
    PAtom: PAtomData;
begin
    Screen.Cursor := crHourGlass;
    {Очищаем списки(они могли быть ранее использованы)}
    for I := 0 to Al_lrow_Build_List.Count-1 do

```

```

begin
  PAtom := Al_1row_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
Al_1row_Build_List.Clear;
for I := 0 to Oxigen_1row_Build_List.Count-1 do
begin
  PAtom := Oxigen_1row_Build_List[I];
  Dispose(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
Oxigen_1row_Build_List.Clear;
Al_1row_Clean_List.Clear;
Oxigen_1row_Clean_List.Clear;
Al_1row_Cuttet_List.Clear;
Oxigen_1row_Cuttet_List.Clear;
Al_1row_Build_P_Memo.Clear;
Al_1row_Build_Nil_P_Memo.Clear;
Al_1row_Clean_P_Memo.Clear;
Oxigen_1row_Build_P_Memo.Clear;
Al_1row_Cuttet_P_Memo.Clear;
Oxigen_1row_Cuttet_P_Memo.Clear;
N_CellsToAdd := StrToInt(N_CellsToAdd_P_Edit.Text); {сколько раз повторять
добавление пар ячеек в горизонтальный ряд}
                {Копируем первую ячейку}
for J := 0 to N_Al_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := PAtomData(Al_1cell_List[J]).X;
  PAtom^.Y := PAtomData(Al_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
  Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := PAtomData(Oxigen_1cell_List[J]).X;
  PAtom^.Y := PAtomData(Oxigen_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
  Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
                {Добавляем 1 ячейку с центром в x=-a, y=a*cR3}
XNew := -a;
YNew := a*cR3;
for J := 0 to N_Al_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
  PAtom^.Y := YNew+PAtomData(Al_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
  Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := PAtomData(Oxigen_1cell_List[J]).X;
  PAtom^.Y := PAtomData(Oxigen_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
  Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
                {Цикл добавления пар ячеек}

```

```

for I := 1 to N_CellsToAdd do
begin
  XNew := 2*a*I; {первая ячейка в паре с центром в x=2*a*I, y=0}
  for J := 0 to N_Al_1cell-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
    PAtom^.Y := PAtomData(Al_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
    Al_1row_Build_List.Add(PAtom);
  end; {for J = 0 to N_Al_1cell-1 do ...}
  for J := 0 to N_Oxigen_1cell-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Oxigen_1cell_List[J]).X;
    PAtom^.Y := PAtomData(Oxigen_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
    Oxigen_1row_Build_List.Add(PAtom);
  end; {for J = 0 to N_Al_1cell-1 do ...}
  {Вторая ячейка в паре с центром в x=(2*I-1)*a, y=a*cR3}
  XNew := (2*I-1)*a;
  YNew := a*cR3;
  for J := 0 to N_Al_1cell-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Al_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
    Al_1row_Build_List.Add(PAtom);
  end; {for J = 0 to N_Al_1cell-1 do ...}
  for J := 0 to N_Oxigen_1cell-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Oxigen_1cell_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Oxigen_1cell_List[J]).Y;
    PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
    Oxigen_1row_Build_List.Add(PAtom);
  end; {for J = 0 to N_Al_1cell-1 do ...}
end; {for I := 1 to N_CellsToAdd do ...}
  {Добавляем 1 ячейку с центром в x=(2*N+1)*a, y=a*cR3}
XNew := (2*N_CellsToAdd+1)*a;
YNew := a*cR3;
for J := 0 to N_Al_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := XNew+PAtomData(Al_1cell_List[J]).X;
  PAtom^.Y := YNew+PAtomData(Al_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Al_1cell_List[J]).Z;
  Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
  New(PAtom);
  PAtom^.X := PAtomData(Oxigen_1cell_List[J]).X;
  PAtom^.Y := PAtomData(Oxigen_1cell_List[J]).Y;
  PAtom^.Z := PAtomData(Oxigen_1cell_List[J]).Z;
  Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for I := 0 to Al_1row_Build_List.Count-1 do {выводим результаты}
Al_1row_Build_P_Memo.Lines.Add(IntToStr(I)+' ')

```

```

+FloatToStr(PAtomData(Al_1row_Build_List[I])^.X)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Y)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Z));
N_Al_1row_P_Label.Caption := 'N_Al_1row '
                        +IntToStr(Al_1row_Build_List.Count);
for I := 0 to Oxigen_1row_Build_List.Count-1 do
  Oxigen_1row_Build_P_Memo.Lines.Add(IntToStr(I)+' '
    + FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.X)+' '
    + FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.Y)+' '
    + FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.Z));
N_Oxigen_1row_P_Label.Caption := 'N_Oxigen_1row '
                        +IntToStr(Oxigen_1row_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_1row_Build_List.Count-1 do
  begin
    PAtom := Al_1row_Build_List[I];
    Al_1row_Clean_List.Add(PAtom);
  end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
for I := 0 to Oxigen_1row_Build_List.Count-1 do
  begin
    PAtom := Oxigen_1row_Build_List[I];
    Oxigen_1row_Clean_List.Add(PAtom);
  end; {for I := 0 to Oxigen_1row_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_1row_Clean_List.Count-2 do
  for J := I+1 to Al_1row_Clean_List.Count-1 do
    if AtomsCoincide(PAtomData(Al_1row_Clean_List[I]),
      PAtomData(Al_1row_Clean_List[J])) then
      Al_1row_Clean_List[J] := nil;
Al_1row_Build_Nil_Memo.Clear; {Выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
  begin
    if Al_1row_Clean_List[I] = nil then
      Al_1row_Build_Nil_P_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Build_Nil_P_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
  end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
  {!!! Атомы O повторяться не могут. Нечего и проверять !!!}
  {Удаляем указатели с nil из списка}
for I := Al_1row_Clean_List.Count-1 downto 0 do
  if Al_1row_Clean_List[I] = Nil then Al_1row_Clean_List.Delete(I);
Al_1row_Clean_P_Memo.Clear; {Выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
  begin
    if Al_1row_Clean_List[I] = nil then
      Al_1row_Clean_P_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Clean_P_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
  end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
N_Al_1row_Clean_P_Label.Caption := 'N_Al_1row_Clean '
                        +IntToStr(Al_1row_Clean_List.Count);
{Обрезаем блок до прямоугольника. По толщине -rOpD2 < y < 2*rOr,
по ширине -a < x < 2*N*a}
BlockLeft := -1.01*rOpD2;
BlockRight := 1.01*2*rOp;

```



```

BlockFront := (*1.01*(2*N_CellsToAdd+1)*a*) 1.01*2*N_CellsToAdd*a;
BlockBack := -1.01*a;
{Заполняем новые списки записями атомов, лежащих внутри прямоугольника}
for I := 0 to Al_1row_Clean_List.Count-1 do
begin
  PAtom := Al_1row_Clean_List[I];
  if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
    (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
  begin
    Al_1row_Cuttet_List.Add(PAtom);
    Al_1row_Cuttet_P_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
      +FloatToStr(PAtom^.Z));
  end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
end; {for I := 0 to Al_1row_Clean_List.Count-1 do}
N_Al_1row_Cuttet_P_Label.Caption := 'Al_1row_Cuttet '
+ IntToStr(Al_1row_Cuttet_List.Count);
for I := 0 to Oxigen_1row_Clean_List.Count-1 do
begin
  PAtom := Oxigen_1row_Clean_List[I];
  if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
    (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
  begin
    Oxigen_1row_Cuttet_List.Add(PAtom);
    Oxigen_1row_Cuttet_P_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
      +FloatToStr(PAtom^.Z));
  end; {for I := 0 to Oxigen_1row_Clean_List.Count-1 do ...}
end; {for I := 0 to Oxigen_1row_Clean_List.Count-1 do}
N_Oxigen_1row_Cuttet_P_Label.Caption := 'Oxigen_1row_Cuttet '
+ IntToStr(Oxigen_1row_Cuttet_List.Count);
Screen.Cursor := crDefault;
end; {procedure TMainForm.FormRow_P_ButtonClick}
{-----}
procedure TMainForm.Save_1row_P_ButtonClick(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  Al_1row_FileName := ExtractFilePath(Application.ExeName)
    +Al_1row_ShortFileName;
  AssignFile(Al_1row_File,Al_1row_FileName);
  try
    Rewrite(Al_1row_File);
    for I := 0 to Al_1row_Cuttet_List.Count-1 do
    begin
      PAtom := Al_1row_Cuttet_List[I];
      WriteLn(Al_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
    end; { for I := 0 to Al_1row_Cuttet_List.Count-1 do ...}
  finally CloseFile(Al_1row_File);
  end; {try ... finally ...}
  Oxigen_1row_FileName := ExtractFilePath(Application.ExeName)
    +Oxigen_1row_ShortFileName;
  AssignFile(Oxigen_1row_File,Oxigen_1row_FileName);
  try
    Rewrite(Oxigen_1row_File);
    for I := 0 to Oxigen_1row_Cuttet_List.Count-1 do
    begin
      PAtom := Oxigen_1row_Cuttet_List[I];
      WriteLn(Oxigen_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
    end; { for I := 0 to Oxigen_1row_Cuttet_List.Count-1 do ...}
  finally CloseFile(Oxigen_1row_File);
  end; {try ... finally ...}
end;

```

```

        end; { for I := 0 to Oxigen_1row_Cuttet_List.Count-1 do ...}
    finally CloseFile(Oxigen_1row_File);
end; {try ... finally ...}
Screen.Cursor := crDefault;
end; {procedure TMainForm.Save_1row_P_ButtonClick}
{=====}
    {Обработчики событий элементов управления. Страница Пирамиды блок.}
{=====}
procedure TMainForm.FormBlock_P_ButtonClick(Sender: TObject);{в процедуре
используются те же списки, что и в TMainForm.FormBlockButtonClick при
построении блока ромбоэдров}
    var I,J: Integer;
        PAtom: PAtomData;
    begin
        Screen.Cursor := crHourGlass;
        {Очищаем списки(они могли быть ранее использованы)}
        for I := 0 to Al_Block_Build_List.Count-1 do
            begin
                PAtom := Al_Block_Build_List[I];
                Dispose(PAtom);
            end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
        Al_Block_Build_List.Clear;
        for I := 0 to Oxigen_Block_Build_List.Count-1 do
            begin
                PAtom := Oxigen_Block_Build_List[I];
                Dispose(PAtom);
            end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
        Oxigen_Block_Build_List.Clear;
        Al_Block_Build_List.Clear;
        Oxigen_Block_Build_List.Clear;
        Al_Block_Build_P_Memo.Clear;
        Al_Block_Build_Nil_P_Memo.Clear;
        Al_Block_Build_P_Memo.Clear;
        Oxigen_Block_Build_P_Memo.Clear;
        N_RowsToAdd := StrToInt(N_RowsToAdd_P_Edit.Text); {сколько рядов
        добавлять}
        {Копируем первый ряд}
        for J := 0 to Al_1row_Cuttet_List.Count-1 do
            begin
                New(PAtom);
                PAtom^.X := PAtomData(Al_1row_Cuttet_List[J]).X;
                PAtom^.Y := PAtomData(Al_1row_Cuttet_List[J]).Y;
                PAtom^.Z := PAtomData(Al_1row_Cuttet_List[J]).Z;
                Al_Block_Build_List.Add(PAtom);
            end; {for J := 0 to Al_1row_Cuttet_List.Count-1 do ...}
        for J := 0 to Oxigen_1row_Cuttet_List.Count-1 do
            begin
                New(PAtom);
                PAtom^.X := PAtomData(Oxigen_1row_Cuttet_List[J]).X;
                PAtom^.Y := PAtomData(Oxigen_1row_Cuttet_List[J]).Y;
                PAtom^.Z := PAtomData(Oxigen_1row_Cuttet_List[J]).Z;
                Oxigen_Block_Build_List.Add(PAtom);
            end; {for J := 0 to Oxigen_1row_Cuttet_List.Count-1 do ...}
            {Добавляем заданное число рядов}
        for I := 1 to N_RowsToAdd do
            begin
                for J := 0 to Al_1row_Cuttet_List.Count-1 do
                    begin
                        New(PAtom);
                        PAtom^.X := PAtomData(Al_1row_Cuttet_List[J]).X;

```

```

    PAtom^.Y := PAtomData(Al_1row_Cuttet_List[J])^.Y;
    PAtom^.Z := PAtomData(Al_1row_Cuttet_List[J])^.Z+c*I;
    Al_Block_Build_List.Add(PAtom);
end; {for J := 0 to Al_1row_Cuttet_List.Count-1 do ...}
for J := 0 to Oxigen_1row_Cuttet_List.Count-1 do
begin
    New(PAtom);
    PAtom^.X := PAtomData(Oxigen_1row_Cuttet_List[J])^.X;
    PAtom^.Y := PAtomData(Oxigen_1row_Cuttet_List[J])^.Y;
    PAtom^.Z := PAtomData(Oxigen_1row_Cuttet_List[J])^.Z+c*I;
    Oxigen_Block_Build_List.Add(PAtom);
end; {for J := 0 to Oxigen_1row_Cuttet_List.Count-1 do ...}
end; {for I := 1 to N_RowsToAdd do ...}
for I := 0 to Al_Block_Build_List.Count-1 do {выводим результаты}
    Al_Block_Build_P_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Build_List[I])^.X)+' '
    + FloatToStr(PAtomData(Al_Block_Build_List[I])^.Y)+' '
    + FloatToStr(PAtomData(Al_Block_Build_List[I])^.Z));
N_Al_Block_P_Label.Caption := 'N_Al_Block '
    +IntToStr(Al_Block_Build_List.Count);
for I := 0 to Oxigen_Block_Build_List.Count-1 do
    Oxigen_Block_Build_P_Memo.Lines.Add(IntToStr(I)+' '
    + FloatToStr(PAtomData(Oxigen_Block_Build_List[I])^.X)+' '
    + FloatToStr(PAtomData(Oxigen_Block_Build_List[I])^.Y)+' '
    + FloatToStr(PAtomData(Oxigen_Block_Build_List[I])^.Z));
N_Oxigen_Block_P_Label.Caption := 'N_Oxigen_Block '
    +IntToStr(Oxigen_Block_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_Block_Build_List.Count-1 do
begin
    PAtom := Al_Block_Build_List[I];
    Al_Block_Clean_List.Add(PAtom);
end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
for I := 0 to Oxigen_Block_Build_List.Count-1 do
begin
    PAtom := Oxigen_Block_Build_List[I];
    Oxigen_Block_Clean_List.Add(PAtom);
end; {for I := 0 to Oxigen_Block_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_Block_Clean_List.Count-2 do
for J := I+1 to Al_Block_Clean_List.Count-1 do
    if AtomsCoincide(PAtomData(Al_Block_Clean_List[I]),
        PAtomData(Al_Block_Clean_List[J])) then
        Al_Block_Clean_List[J] := nil;
Al_Block_Build_Nil_P_Memo.Clear; {выводим результаты}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
    if Al_Block_Clean_List[I] = nil then
        Al_Block_Build_Nil_P_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_Block_Build_Nil_P_Memo.Lines.Add(IntToStr(I)+' '
        +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
        +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
        +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
end; {for I := 0 to Al_Block_Clean_List.Count-1 do ...}
{!!! Атомы O повторяться не могут. Нечего и проверять !!!}
{Удаляем указатели с nil из списка}
for I := Al_Block_Clean_List.Count-1 downto 0 do
    if Al_Block_Clean_List[I] = Nil then Al_Block_Clean_List.Delete(I);
Al_Block_Clean_P_Memo.Clear; {выводим результаты}

```

```

for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  if Al_Block_Clean_List[I] = nil then
    Al_Block_Clean_P_Memo.Lines.Add(IntToStr(I)+' nil')
  else Al_Block_Clean_P_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
  end; {for I := 0 to Al_lrow_Clean_List.Count-1 do ...}
N_Al_Block_Clean_P_Label.Caption := 'N_Al_Block_Clean '
    +IntToStr(Al_Block_Clean_List.Count);

  Screen.Cursor := crDefault;
end; {procedure TMainForm.FormBlock_P_ButtonClick}
{-----}
procedure TMainForm.MoveBlosk_P_ButtonClick(Sender: TObject);{поворачивает
блок на 90 град., меняя у на х и х на -у и одновременно перемещает начало
координат блока в середину нижнего левого ребра сдвигая атомы по новому у на
a}
var I: Integer;
    PAtom: PAtomData;
    Buff: Double;
begin
  Screen.Cursor := crHourGlass;
  Al_Block_Clean_P_Memo.Lines.Clear;
  Oxigen_Block_Build_P_Memo.Lines.Clear;
  for I := 0 to Al_Block_Clean_List.Count-1 do
  begin
    PAtom := Al_Block_Clean_List[I];
    Buff := PAtom^.Y;
    PAtom^.Y := PAtom^.X+a;
    PAtom^.X := -Buff;
    Al_Block_Clean_P_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
      +FloatToStr(PAtom^.Z));
    end; { for I := 0 to .Al_Block_Clean_ListCount-1 do ...}
  N_Al_Block_Clean_P_Label.Caption :=
    N_Al_Block_Clean_P_Label.Caption+' Shifted';
  for I := 0 to Oxigen_Block_Clean_List.Count-1 do
  begin
    PAtom := Oxigen_Block_Clean_List[I];
    Buff := PAtom^.Y;
    PAtom^.Y := PAtom^.X+a;
    PAtom^.X := -Buff;
    Oxigen_Block_Build_P_Memo.Lines.Add(IntToStr(I)+' '
      + FloatToStr(PAtom^.X)+' '+ FloatToStr(PAtom^.Y)+' '
      + FloatToStr(PAtom^.Z));
    end; { for I := 0 to Oxigen_Block_Clean_List.Count-1 do ...}
  N_Oxigen_Block_P_Label.Caption :=N_Oxigen_Block_P_Label.Caption+'
  Shifted';
  Screen.Cursor := crDefault;
end; {procedure TMainForm.MoveBlosk_P_ButtonClick}
{-----}
procedure TMainForm.Save_Block_P_ButtonClick(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  Al_Block_FileName := ExtractFilePath(Application.ExeName)
    +Al_Block_ShortFileName;
  AssignFile(Al_Block_File,Al_Block_FileName);

```

```

try
  Rewrite(Al_Block_File);
  for I := 0 to Al_Block_Clean_List.Count-1 do
  begin
    PAtom := Al_Block_Clean_List[I];
    WriteLn(Al_Block_File, PAtom^.X, ' ', PAtom^.Y, ' ', PAtom^.Z);
  end; { for I := 0 to .Al_Block_Clean_ListCount-1 do ...}
finally CloseFile(Al_Block_File);
end; {try ... finally ...}
Oxigen_Block_FileName := ExtractFilePath(Application.ExeName)
                        +Oxigen_Block_ShortFileName;
AssignFile(Oxigen_Block_File, Oxigen_Block_FileName);
try
  Rewrite(Oxigen_Block_File);
  for I := 0 to Oxigen_Block_Clean_List.Count-1 do
  begin
    PAtom := Oxigen_Block_Clean_List[I];
    WriteLn(Oxigen_Block_File, PAtom^.X, ' ', PAtom^.Y, ' ', PAtom^.Z);
  end; { for I := 0 to Oxigen_Block_Clean_List.Count-1 do ...}
finally CloseFile(Oxigen_Block_File);
end; {try ... finally ...}
Screen.Cursor := crDefault;
end; {procedure TMainForm.Save_Block_P_ButtonClick}
{=====}
  {Обработчики событий элементов управления. Страница Призмы 1 ряд.}
{=====}
procedure TMainForm.FormRow_Pr_ButtonClick(Sender: TObject); {в процедуре
используются те же списки, что и в TMainForm.FormRowButtonClick при построении
ряда для блока ромбоэдров; обрезание ряда не производится: обрезать будем блок
в целом}
var I, J: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  {Очищаем списки(они могли быть ранее использованы)}
  for I := 0 to Al_1row_Build_List.Count-1 do
  begin
    PAtom := Al_1row_Build_List[I];
    Dispose(PAtom);
  end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
  Al_1row_Build_List.Clear;
  for I := 0 to Oxigen_1row_Build_List.Count-1 do
  begin
    PAtom := Oxigen_1row_Build_List[I];
    Dispose(PAtom);
  end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
  Oxigen_1row_Build_List.Clear;
  Al_1row_Clean_List.Clear;
  Oxigen_1row_Clean_List.Clear;
  Al_1row_Cuttet_List.Clear;
  Oxigen_1row_Cuttet_List.Clear;
  Al_1row_Build_Pr_Memo.Clear;
  Al_1row_Build_Nil_Pr_Memo.Clear;
  Al_1row_Clean_Pr_Memo.Clear;
  Oxigen_1row_Build_Pr_Memo.Clear;
  N_CellsToAdd := StrToInt(N_CellsToAdd_Pr_Edit.Text); {сколько раз
повторять добавление ячейки в первый ряд}
  {Копируем первую ячейку}
  for J := 0 to N_Al_1cell-1 do
  begin

```

```

New(PAtom);
PAtom^.X := PAtomData(Al_1cell_List[J])^.X;
PAtom^.Y := PAtomData(Al_1cell_List[J])^.Y;
PAtom^.Z := PAtomData(Al_1cell_List[J])^.Z;
Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
New(PAtom);
PAtom^.X := PAtomData(Oxigen_1cell_List[J])^.X;
PAtom^.Y := PAtomData(Oxigen_1cell_List[J])^.Y;
PAtom^.Z := PAtomData(Oxigen_1cell_List[J])^.Z;
Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
{Цикл добавления ячеек в первый ряд}
for I := 1 to N_CellsToAdd do
begin
XNew := 2*a*I; {ячейка с центром в x=2*a*I, y=0}
for J := 0 to N_Al_1cell-1 do
begin
New(PAtom);
PAtom^.X := XNew+PAtomData(Al_1cell_List[J])^.X;
PAtom^.Y := PAtomData(Al_1cell_List[J])^.Y;
PAtom^.Z := PAtomData(Al_1cell_List[J])^.Z;
Al_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
for J := 0 to N_Oxigen_1cell-1 do
begin
New(PAtom);
PAtom^.X := XNew+PAtomData(Oxigen_1cell_List[J])^.X;
PAtom^.Y := PAtomData(Oxigen_1cell_List[J])^.Y;
PAtom^.Z := PAtomData(Oxigen_1cell_List[J])^.Z;
Oxigen_1row_Build_List.Add(PAtom);
end; {for J = 0 to N_Al_1cell-1 do ...}
end; {for I := 1 to N_CellsToAdd do ...}
for I := 0 to Al_1row_Build_List.Count-1 do {выводим результаты}
Al_1row_Build_Pr_Memo.Lines.Add(IntToStr(I)+' '
+FloatToStr(PAtomData(Al_1row_Build_List[I])^.X)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Y)+' '
+ FloatToStr(PAtomData(Al_1row_Build_List[I])^.Z));
N_Al_1row_Pr_Label.Caption := 'N_Al_1row '
+IntToStr(Al_1row_Build_List.Count);
for I := 0 to Oxigen_1row_Build_List.Count-1 do
Oxigen_1row_Build_Pr_Memo.Lines.Add(IntToStr(I)+' '
+ FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.X)+' '
+ FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.Y)+' '
+ FloatToStr(PAtomData(Oxigen_1row_Build_List[I])^.Z));
N_Oxigen_1row_Pr_Label.Caption := 'N_Oxigen_1row '
+IntToStr(Oxigen_1row_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_1row_Build_List.Count-1 do
begin
PAtom := Al_1row_Build_List[I];
Al_1row_Clean_List.Add(PAtom);
end; {for I := 0 to Al_1row_Build_List.Count-1 do ...}
for I := 0 to Oxigen_1row_Build_List.Count-1 do
begin
PAtom := Oxigen_1row_Build_List[I];
Oxigen_1row_Clean_List.Add(PAtom);

```

```

    end; {for I := 0 to Oxigen_1row_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_1row_Clean_List.Count-2 do
  for J := I+1 to Al_1row_Clean_List.Count-1 do
    if AtomsCoincide(PAtomData(Al_1row_Clean_List[I]),
      PAtomData(Al_1row_Clean_List[J])) then
      Al_1row_Clean_List[J] := nil;
      {Выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
  begin
    if Al_1row_Clean_List[I] = nil then
      Al_1row_Build_Nil_Pr_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Build_Nil_Pr_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
  end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
  {!!! Атомы O повторяются не могут. Нечего и проверять !!!}
  {Удаляем указатели с nil из списка}
for I := Al_1row_Clean_List.Count-1 downto 0 do
  if Al_1row_Clean_List[I] = Nil then Al_1row_Clean_List.Delete(I);
  {Выводим результаты}
for I := 0 to Al_1row_Clean_List.Count-1 do
  begin
    if Al_1row_Clean_List[I] = nil then
      Al_1row_Clean_Pr_Memo.Lines.Add(IntToStr(I)+' nil')
    else Al_1row_Clean_Pr_Memo.Lines.Add(IntToStr(I)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.X)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Y)+' '
      +FloatToStr(PAtomData(Al_1row_Clean_List[I])^.Z));
  end; {for I := 0 to Al_1row_Clean_List.Count-1 do ...}
  N_Al_1row_Clean_Pr_Label.Caption := 'N_Al_1row_Clean '
    +IntToStr(Al_1row_Clean_List.Count);

  Screen.Cursor := crDefault;
end; {procedure TMainForm.FormRow_Pr_ButtonClick}
{-----}
procedure TMainForm.Save_1row_Pr_ButtonClick(Sender: TObject);
var I: Integer;
    PAtom: PAtomData;
begin
  Screen.Cursor := crHourGlass;
  Al_1row_FileName := ExtractFilePath(Application.ExeName)
    +Al_1row_ShortFileName;
  AssignFile(Al_1row_File,Al_1row_FileName);
  try
    Rewrite(Al_1row_File);
    for I := 0 to Al_1row_Clean_List.Count-1 do
      begin
        PAtom := Al_1row_Clean_List[I];
        WriteLn(Al_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
      end; { for I := 0 to Al_1row_Clean_List.Count-1 do ...}
  finally CloseFile(Al_1row_File);
  end; {try ... finally ...}
  Oxigen_1row_FileName := ExtractFilePath(Application.ExeName)
    +Oxigen_1row_ShortFileName;
  AssignFile(Oxigen_1row_File,Oxigen_1row_FileName);
  try
    Rewrite(Oxigen_1row_File);
    for I := 0 to Oxigen_1row_Clean_List.Count-1 do
      begin

```

```

        PAtom := Oxigen_1row_Clean_List[I];
        WriteLn(Oxigen_1row_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
    end; { for I := 0 to Oxigen_1row_Clean_List.Count-1 do ...}
    finally CloseFile(Oxigen_1row_File);
end; {try ... finally ...}
Screen.Cursor := crDefault;
end; {procedure TMainForm.Save_1row_Pr_ButtonClick}
{=====}
    {Обработчики событий элементов управления. Страница Призмы блок.}
{=====}
procedure TMainForm.FormBlock_Pr_ButtonClick(Sender: TObject); {в процедуре
используются те же списки, что и в TMainForm.FormBlockButtonClick при
построении блока ромбоэдров; в отличии от других блоков добавляются пары
рядов, а обрезание производится в конце построения}
    var I,J: Integer;
        PAtom: PAtomData;
    begin
        Screen.Cursor := crHourGlass;
        {Очищаем списки(они могли быть ранее использованы)}
        for I := 0 to Al_Block_Build_List.Count-1 do
            begin
                PAtom := Al_Block_Build_List[I];
                Dispose(PAtom);
            end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
        Al_Block_Build_List.Clear;
        for I := 0 to Oxigen_Block_Build_List.Count-1 do
            begin
                PAtom := Oxigen_Block_Build_List[I];
                Dispose(PAtom);
            end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
        Oxigen_Block_Build_List.Clear;
        Al_Block_Clean_List.Clear;
        Oxigen_Block_Clean_List.Clear;
        Al_Block_Cuttet_List.Clear;
        Oxigen_Block_Cuttet_List.Clear;
        Al_Block_Build_Pr_Memo.Clear;
        Al_Block_Build_Nil_Pr_Memo.Clear;
        Al_Block_Clean_Pr_Memo.Clear;
        Oxigen_Block_Build_Pr_Memo.Clear;
        Al_Block_Cuttet_Pr_Memo.Clear;
        Oxigen_Block_Cuttet_Pr_Memo.Clear;
        N_RowsToAdd := StrToInt(N_RowsToAdd_Pr_Edit.Text); {сколько пар рядов
добавлять}
        {Копируем первый ряд}
        for J := 0 to Al_1row_Clean_List.Count-1 do
            begin
                New(PAtom);
                PAtom^.X := PAtomData(Al_1row_Clean_List[J]).X;
                PAtom^.Y := PAtomData(Al_1row_Clean_List[J]).Y;
                PAtom^.Z := PAtomData(Al_1row_Clean_List[J]).Z;
                Al_Block_Build_List.Add(PAtom);
            end; {for J := 0 to Al_1row_Clean_List.Count-1 do ...}
        for J := 0 to Oxigen_1row_Clean_List.Count-1 do
            begin
                New(PAtom);
                PAtom^.X := PAtomData(Oxigen_1row_Clean_List[J]).X;
                PAtom^.Y := PAtomData(Oxigen_1row_Clean_List[J]).Y;
                PAtom^.Z := PAtomData(Oxigen_1row_Clean_List[J]).Z;
                Oxigen_Block_Build_List.Add(PAtom);
            end; {for J := 0 to Oxigen_1row_Clean_List.Count-1 do ...}

```



```

for I := 1 to N_RowsToAdd do {добавляем заданное число пар}
begin
  {Добавляем первый ряд пары. Смещение по x=-a, по y=a*cR3*(2*I-1)}
  XNew := -a;
  YNew := a*cR3*(2*I-1);
  for J := 0 to Al_1row_Clean_List.Count-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Al_1row_Clean_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Al_1row_Clean_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1row_Clean_List[J]).Z;
    Al_Block_Build_List.Add(PAtom);
  end; {for J := 0 to Al_1row_Clean_List.Count-1 do ...}
  for J := 0 to Oxigen_1row_Clean_List.Count-1 do
  begin
    New(PAtom);
    PAtom^.X := XNew+PAtomData(Oxigen_1row_Clean_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Oxigen_1row_Clean_List[J]).Y;
    PAtom^.Z := PAtomData(Oxigen_1row_Clean_List[J]).Z;
    Oxigen_Block_Build_List.Add(PAtom);
  end; {for J := 0 to Oxigen_1row_Clean_List.Count-1 do ...}
  {Добавляем второй ряд пары. Смещение по x=0, по y=a*cR3*(2*I)}
  YNew := a*cR3*2*I;
  for J := 0 to Al_1row_Clean_List.Count-1 do
  begin
    New(PAtom);
    PAtom^.X := PAtomData(Al_1row_Clean_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Al_1row_Clean_List[J]).Y;
    PAtom^.Z := PAtomData(Al_1row_Clean_List[J]).Z;
    Al_Block_Build_List.Add(PAtom);
  end; {for J := 0 to Al_1row_Clean_List.Count-1 do ...}
  for J := 0 to Oxigen_1row_Clean_List.Count-1 do
  begin
    New(PAtom);
    PAtom^.X := PAtomData(Oxigen_1row_Clean_List[J]).X;
    PAtom^.Y := YNew+PAtomData(Oxigen_1row_Clean_List[J]).Y;
    PAtom^.Z := PAtomData(Oxigen_1row_Clean_List[J]).Z;
    Oxigen_Block_Build_List.Add(PAtom);
  end; {for J := 0 to Oxigen_1row_Clean_List.Count-1 do ...}
end; {for I := 1 to N_RowsToAdd do ...}
for I := 0 to Al_Block_Build_List.Count-1 do {выводим результаты}
  Al_Block_Build_Pr_Memo.Lines.Add(IntToStr(I)+' '
  +FloatToStr(PAtomData(Al_Block_Build_List[I]).X)+' '
  + FloatToStr(PAtomData(Al_Block_Build_List[I]).Y)+' '
  + FloatToStr(PAtomData(Al_Block_Build_List[I]).Z));
  N_Al_Block_Pr_Label.Caption := 'N_Al_Block '
  +IntToStr(Al_Block_Build_List.Count);
for I := 0 to Oxigen_Block_Build_List.Count-1 do
  Oxigen_Block_Build_Pr_Memo.Lines.Add(IntToStr(I)+' '
  + FloatToStr(PAtomData(Oxigen_Block_Build_List[I]).X)+' '
  + FloatToStr(PAtomData(Oxigen_Block_Build_List[I]).Y)+' '
  + FloatToStr(PAtomData(Oxigen_Block_Build_List[I]).Z));
  N_Oxigen_Block_Pr_Label.Caption := 'N_Oxigen_Block '
  +IntToStr(Oxigen_Block_Build_List.Count);
{Удаление повторяющихся атомов. Сначала копируем списки, создавая повторные
указатели, но не новые записи атомов}
for I := 0 to Al_Block_Build_List.Count-1 do
begin
  PAtom := Al_Block_Build_List[I];
  Al_Block_Clean_List.Add(PAtom);

```

```

end; {for I := 0 to Al_Block_Build_List.Count-1 do ...}
for I := 0 to Oxigen_Block_Build_List.Count-1 do
begin
  PAtom := Oxigen_Block_Build_List[I];
  Oxigen_Block_Clean_List.Add(PAtom);
end; {for I := 0 to Oxigen_Block_Build_List.Count-1 do ...}
{Далее работаем с дублями указателей. Ищем повторы и заменяем их на nil.}
for I := 0 to Al_Block_Clean_List.Count-2 do
  for J := I+1 to Al_Block_Clean_List.Count-1 do
    if AtomsCoincide(PAtomData(Al_Block_Clean_List[I]),
      PAtomData(Al_Block_Clean_List[J])) then
      Al_Block_Clean_List[J] := nil;
      {Выводим результаты}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  if Al_Block_Clean_List[I] = nil then
    Al_Block_Build_Nil_Pr_Memo.Lines.Add(IntToStr(I)+' nil')
  else Al_Block_Build_Nil_Pr_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
end; {for I := 0 to Al_Block_Clean_List.Count-1 do ...}
{!!! Атомы O повторяться не могут. Нечего и проверять !!!}
{Удаляем указатели с nil из списка}
for I := Al_Block_Clean_List.Count-1 downto 0 do
  if Al_Block_Clean_List[I] = Nil then Al_Block_Clean_List.Delete(I);
  {Выводим результаты}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  if Al_Block_Clean_List[I] = nil then
    Al_Block_Clean_Pr_Memo.Lines.Add(IntToStr(I)+' nil')
  else Al_Block_Clean_Pr_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.X)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Y)+' '
    +FloatToStr(PAtomData(Al_Block_Clean_List[I])^.Z));
end; {for I := 0 to Al_lrow_Clean_List.Count-1 do ...}
N_Al_Block_Clean_Pr_Label.Caption := 'N_Al_Block_Clean '
  +IntToStr(Al_Block_Clean_List.Count);
{Обрезаем блок до прямоугольника. -rOpD2 < y < (3*Nr+0.5)*rOp,
  -a < x < 2*Nc*a}
BlockLeft := -1.01*rOpD2;
BlockRight := 1.01*(3*N_RowsToAdd+0.5)*rOp;
BlockFront := 1.01*2*N_CellsToAdd*a;
BlockBack := -1.01*a;
{Заполняем новые списки записями атомов, лежащих внутри прямоугольника}
for I := 0 to Al_Block_Clean_List.Count-1 do
begin
  PAtom := Al_Block_Clean_List[I];
  if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
    (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
begin
  Al_Block_Cuttet_List.Add(PAtom);
  Al_Block_Cuttet_Pr_Memo.Lines.Add(IntToStr(I)+' '
    +FloatToStr(PAtom^.X)+' ' +FloatToStr(PAtom^.Y)+' '
    +FloatToStr(PAtom^.Z));
end; {if (PAtom^.Y > BlockLeft) and ... then ...}
end; {for I := 0 to Al_Block_Clean_List.Count-1 do}
N_Al_Block_Cuttet_Pr_Label.Caption := 'Al_Block_Cuttet '
  + IntToStr(Al_Block_Cuttet_List.Count);
for I := 0 to Oxigen_Block_Clean_List.Count-1 do

```

```

begin
  PAtom := Oxigen_Block_Clean_List[I];
  if (PAtom^.Y > BlockLeft) and (PAtom^.Y < BlockRight) and
    (PAtom^.X > BlockBack) and (PAtom^.X < BlockFront) then
    begin
      Oxigen_Block_Cuttet_List.Add(PAtom);
      Oxigen_Block_Cuttet_Pr_Memo.Lines.Add(IntToStr(I)+' '
        +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
        +FloatToStr(PAtom^.Z));
      end; {if (PAtom^.Y > BlockLeft) and ... then ...}
    end; {for I := 0 to Oxigen_Block_Clean_List.Count-1 do}
  N_Oxigen_Block_Cuttet_Pr_Label.Caption := 'Oxigen_Block_Cuttet '
    + IntToStr(Oxigen_Block_Cuttet_List.Count);
  Screen.Cursor := crDefault;
end; {procedure TMainForm.FormBlock_Pr_ButtonClick}
{-----}
procedure TMainForm.MoveBlosk_Pr_ButtonClick(Sender: TObject);{перемещает
начало координат в середину левого заднего ребра блока (на высоту c/2) и
поворачивает блок вокруг Y на -90 градусов (против часовой стрелки)}
  var I: Integer;
      PAtom: PAtomData;
      Buff: Double;
  begin
    Screen.Cursor := crHourGlass;
    Al_Block_Cuttet_Pr_Memo.Lines.Clear;
    Oxigen_Block_Cuttet_Pr_Memo.Lines.Clear;
    for I := 0 to Al_Block_Cuttet_List.Count-1 do
      begin
        PAtom := Al_Block_Cuttet_List[I];
        Buff := PAtom^.X+a;
        PAtom^.Y := PAtom^.Y+rOpD2;
        PAtom^.X := c/2-PAtom^.Z;
        PAtom^.Z := Buff;
        Al_Block_Cuttet_Pr_Memo.Lines.Add(IntToStr(I)+' '
          +FloatToStr(PAtom^.X)+' '+FloatToStr(PAtom^.Y)+' '
          +FloatToStr(PAtom^.Z));
        end; { for I := 0 to .Al_Block_Clean_ListCount-1 do ...}
    N_Al_Block_Cuttet_Pr_Label.Caption :=
      N_Al_Block_Cuttet_Pr_Label.Caption+' Shifted';
    for I := 0 to Oxigen_Block_Cuttet_List.Count-1 do
      begin
        PAtom := Oxigen_Block_Cuttet_List[I];
        Buff := PAtom^.X+a;
        PAtom^.Y := PAtom^.Y+rOpD2;
        PAtom^.X := c/2-PAtom^.Z;
        PAtom^.Z := Buff;
        Oxigen_Block_Cuttet_Pr_Memo.Lines.Add(IntToStr(I)+' '
          + FloatToStr(PAtom^.X)+' ' + FloatToStr(PAtom^.Y)+' '
          + FloatToStr(PAtom^.Z));
        end; { for I := 0 to Oxigen_Block_Clean_List.Count-1 do ...}
    N_Oxigen_Block_Cuttet_Pr_Label.Caption :=
      N_Oxigen_Block_Cuttet_Pr_Label.Caption+' Shifted';
    Screen.Cursor := crDefault;
  end; {procedure TMainForm.MoveBlosk_Pr_ButtonClick}
{-----}
procedure TMainForm.Save_Block_Pr_ButtonClick(Sender: TObject);
  var I: Integer;
      PAtom: PAtomData;
  begin
    Screen.Cursor := crHourGlass;

```

```

Al_Block_FileName := ExtractFilePath(Application.ExeName)
                    +Al_Block_ShortFileName;
AssignFile(Al_Block_File,Al_Block_FileName);
try
  Rewrite(Al_Block_File);
  for I := 0 to Al_Block_Cuttet_List.Count-1 do
    begin
      PAtom := Al_Block_Cuttet_List[I];
      WriteLn(Al_Block_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
    end; { for I := 0 to Al_Block_Cuttet_List.Count-1 do ...}
finally CloseFile(Al_Block_File);
end; {try ... finally ...}
Oxigen_Block_FileName := ExtractFilePath(Application.ExeName)
                    +Oxigen_Block_ShortFileName;
AssignFile(Oxigen_Block_File,Oxigen_Block_FileName);
try
  Rewrite(Oxigen_Block_File);
  for I := 0 to Oxigen_Block_Cuttet_List.Count-1 do
    begin
      PAtom := Oxigen_Block_Cuttet_List[I];
      WriteLn(Oxigen_Block_File,PAtom^.X,' ',PAtom^.Y,' ',PAtom^.Z);
    end; { for I := 0 to Oxigen_Block_Cuttet_List.Count-1 do ...}
finally CloseFile(Oxigen_Block_File);
end; {try ... finally ...}
Screen.Cursor := crDefault;

end; {procedure TMainForm.Save_Block_Pr_ButtonClick}

end.

```

Определение связей Al-O в пределах блока ячеек сапфира

```

unit MainU;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  QDialogs, StdCtrls, Grids, Dialogs, Math;
type
  TMainForm = class(TForm)
    BackButton: TButton;
    ForwardButton: TButton;
    FileOpenDialog: TOpenDialog;
    AlStringGrid: TStringGrid;
    Al_Atoms_Label: TLabel;
    O_Atoms_Label: TLabel;
    OStringGrid: TStringGrid;
    LinksNumLabel: TLabel;
    LinksStringGrid: TStringGrid;
    Label3: TLabel;
    FileSaveDialog: TSaveDialog;
    LinksFileNameLabel: TLabel;
    AngleControlButton: TButton;
    AngleControlMemo: TMemo;
    FillStringGridsCheckBox: TCheckBox;
    procedure ForwardButtonClick(Sender: TObject);
    procedure BackButtonClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  end;

```

```

    procedure AngleControlButtonClick(Sender: TObject);
private
    procedure LoadAlRun;
    procedure LoadORun;
    procedure CalcRun;
    procedure SaveLinksRun;
    procedure FinishRun;
    procedure CliauLinks;
    procedure CliauO;
    procedure CliauAl;
public
    { Public declarations }
end;
var
    MainForm: TMainForm;
    AFile: TextFile; {переменная для работы с файлами}
    DCBuff: Char;
{-----}
        {Стадии работы программы}
{-----}
type TProgramStages = (LoadAl, LoadO, Calc, SaveLinks, Finish);
var ProgramStage: TProgramStages = LoadAl;
{-----}
        {Списки данных и записи для них}
{-----}
var AlDataList, ODataList, LinksDataList: TList;
type PAlRec = ^TAlRec;
    TAlRec = record
        X, Y, Z: Double;
    end; {TAlRec = record ...}
    POREc = ^TORec;
    TORec = record
        X, Y, Z: Double;
    end; {TORec = record ...}
    PLinkRec = ^TLinkRec;
    TLinkRec = record
        AlNum, ONum: Integer;
        Length: Double;
        LongLinc: Boolean;
    end; {TLinkRec = record ...}
const D_AlO_Long = 1.97;
    D_AlO_Short = 1.87;
    Eps = 0.02;
{*****}
implementation
{$R *.dfm}
{=====}
        {Процедуры выполнения стадий программы}
{=====}
procedure TMainForm.LoadAlRun;
    var PAl: PAlRec;
        I: Integer;
    begin
        FileOpenDialog.Title := 'Загрузка данных об Al';
        if FileOpenDialog.Execute then
            begin
                AssignFile(AFile, FileOpenDialog.FileName);
                try
                    Reset(AFile);
                    while not Eof(AFile) do

```

```

begin
  New(PAL);
  ReadLn(AFile, PAL^.X, PAL^.Y, PAL^.Z);
  AlDataList.Add(PAL);
end; {while not Eof(AFile) do ...}
finally CloseFile(AFile);
end; {try ... finally ...}
Al_Atoms_Label.Caption := Al_Atoms_Label.Caption+ '  '
                          +IntToStr(AlDataList.Count);
if FillStringGridsCheckBox.Checked then
begin
  AlStringGrid.RowCount := AlDataList.Count+1;
  for I := 0 to AlDataList.Count-1 do
  begin
    AlStringGrid.Cells[0, I+1] := IntToStr(I);
    PAL := AlDataList[I];
    AlStringGrid.Cells[1, I+1] := FloatToStrF(PAL^.X, ffFixed, 6, 4);
    AlStringGrid.Cells[2, I+1] := FloatToStrF(PAL^.Y, ffFixed, 6, 4);
    AlStringGrid.Cells[3, I+1] := FloatToStrF(PAL^.Z, ffFixed, 6, 4);
  end; {for I := 1 to AlDataList.Count do ...}
end; {if FillStringGridsCheckBox.Checked then ...} (* *)
end; {if FileOpenDialog.Execute then ...}
end; {procedure LoadAlRun}
{-----}
procedure TMainForm.LoadORun;
var
  PO: POREc;
  I: Integer;
begin
  FileOpenDialog.Title := 'Загрузка данных об O';
  if FileOpenDialog.Execute then
  begin
    AssignFile(AFile, FileOpenDialog.FileName);
    try
      Reset(AFile);
      while not Eof(AFile) do
      begin
        New(PO);
        ReadLn(AFile, PO^.X, PO^.Y, PO^.Z);
        ODataList.Add(PO);
      end; {while not Eof(AFile) do ...}
    finally CloseFile(AFile);
    end; {try ... finally ...}
    O_Atoms_Label.Caption := O_Atoms_Label.Caption+ '  '
                          +IntToStr(ODataList.Count);
    if FillStringGridsCheckBox.Checked then
    begin
      OStringGrid.RowCount := ODataList.Count+1;
      for I := 0 to ODataList.Count-1 do
      begin
        OStringGrid.Cells[0, I+1] := IntToStr(I);
        PO := ODataList[I];
        OStringGrid.Cells[1, I+1] := FloatToStrF(PO^.X, ffFixed, 6, 4);
        OStringGrid.Cells[2, I+1] := FloatToStrF(PO^.Y, ffFixed, 6, 4);
        OStringGrid.Cells[3, I+1] := FloatToStrF(PO^.Z, ffFixed, 6, 4);
      end; {for I := 1 to ODataList.Count do ...}
    end; {if FillStringGridsCheckBox.Checked then ...} (* *)
  end; {if FileOpenDialog.Execute then ...}
end; {procedure LoadORun}
{-----}

```

```

procedure TMainForm.CalcRun;
  var I,J,K: Integer; {I -номер Al, J - номер O, K - счётчик длинных связей}
      D: Double; {расстояние Al-O}
      PAL: PALRec;
      PO: POREc;
      PLink: PLinkRec;
begin
  Screen.Cursor := crHourGlass;
  K := 0;
  for I := 0 to AlDataList.Count-1 do {перебираем атомы Al}
    begin
      PAL := AlDataList[I];
      for J := 0 to ODataList.Count-1 do {перебираем атомы O}
        begin
          PO := ODataList[J];
          D := Sqrt(Sqr(PAL^.X-PO^.X)+Sqr(PAL^.Y-PO^.Y)+Sqr(PAL^.Z-PO^.Z));
          if D <= (D_AlO_Long+Eps) then {если это связь}
            begin
              New(PLink); {получаем память}
              PLink^.AlNum := I; {заполняем запись}
              PLink^.ONum := J;
              PLink^.Length := D;
              PLink^.LongLinc := D > (D_AlO_Short+Eps);
              LinksDataList.Add(PLink); {вносим в список связей}
              if PLink^.LongLinc then Inc(K);
            end; {if D <= D_AlO_Long=Eps then ...}
          end; {for J := 0 to ODataList.Count-1 do ...}
        end; {for I := 0 to AlDataList.Count-1 do ...}
      LinksNumLabel.Caption := 'Найдено '+IntToStr(LinksDataList.Count)+'
связей, '
          +'#13#10+'длинных '+IntToStr(K)+' , коротких '
          +IntToStr(LinksDataList.Count-K);
    end;
  if FillStringGridsCheckBox.Checked then
    begin
      LinksStringGrid.RowCount := LinksDataList.Count+1;
      for I := 0 to LinksDataList.Count-1 do
        begin
          LinksStringGrid.Cells[0,I+1] := IntToStr(I);
          PLink := LinksDataList[I];
          LinksStringGrid.Cells[1,I+1] := IntToStr(PLink^.AlNum);
          LinksStringGrid.Cells[2,I+1] := IntToStr(PLink^.ONum);
          LinksStringGrid.Cells[3,I+1] :=
            FloatToStrF(PLink^.Length,ffFixed,6,4);
          if PLink^.LongLinc then LinksStringGrid.Cells[4,I+1] := 'Long'
            else LinksStringGrid.Cells[4,I+1] := 'Short';
          end; {for I := 1 to LinksDataList.Count do ...}
        end; {if FillStringGridsCheckBox.Checked then ...} (**)
      Screen.Cursor := crDefault;
    end; {procedure LoadAlRun}
  {-----}
procedure TMainForm.SaveLinksRun;
  var I,J: Integer;
      PLink: PLinkRec;
begin
  FileSaveDialog.Title := 'Сохранение данных о связях';
  if FileSaveDialog.Execute then
    begin
      AssignFile(AFile,FileSaveDialog.FileName);
      try
        Rewrite(AFile);
      finally
        CloseFile(AFile);
      end;
    end;
end;

```

```

    for I := 0 to LinksDataList.Count-1 do
    begin
        PLink := LinksDataList[I];
        WriteLn(AFile, PLink^.AlNum, ' ', PLink^.ONum, ' ',
            PLink^.Length, ' ', PLink^.LongLinc);
        end; {for I := 0 to LinksDataList.Count-1 do ...}
    finally CloseFile(AFile);
    end; {try ... finally ...}
    LinksFileNameLabel.Caption := 'Данные сохранены в файле'+#13#10
        +FileSaveDialog.FileName;
    end; {if FileSaveDialog.Execute then ...}
end; {procedure LoadAlRun}
{-----}
procedure TMainForm.FinishRun;
begin
    end; {procedure LoadAlRun}
{=====}
    {Процедуры обратных переходов}
{=====}
procedure TMainForm.ClianLinks;
var I,J: Integer;
    PLink: PLinkRec;
begin
    for I := 0 to LinksDataList.Count-1 do
    begin
        PLink := LinksDataList[I];
        Dispose(PLink);
        end; {for I := 0 to LinksDataList.Count-1 do ...}
    LinksDataList.Clear;
    LinksStringGrid.RowCount := 30;
    for I := 1 to 29 do
        for J := 0 to 4 do
            LinksStringGrid.Cells[J,I] := '';
        end; {procedure CliauLinks}
    }-----}
procedure TMainForm.CliauO;
var I,J: Integer;
    PO: POREc;
begin
    for I := 0 to ODataList.Count-1 do
    begin
        PO := ODataList[I];
        Dispose(PO);
        end; {for I := 0 to ODataList.Count-1 do ...}
    ODataList.Clear;
    OStringGrid.RowCount := 30;
    for I := 1 to 29 do
        for J := 0 to 3 do
            OStringGrid.Cells[J,I] := '';
        end; {procedure CliauO}
    }-----}
procedure TMainForm.CliauAl;
var I,J: Integer;
    PAL: PALRec;
begin
    for I := 0 to AlDataList.Count-1 do
    begin
        PAL := AlDataList[I];
        Dispose(PAL);
    }

```



```

    end; {for I := 0 to AlDataList.Count-1 do ...}
    AlDataList.Clear;
    AlStringGrid.RowCount := 30;
    for I := 1 to 29 do
        for J := 0 to 3 do
            AlStringGrid.Cells[J,I] := '';
        end; {procedure CliaAl}
    end; {procedure CliaAl}
{=====}
    {Процедуры кнопок управления}
{=====}
procedure TMainForm.ForwardButtonClick(Sender: TObject);
begin
    case ProgramStage of
        LoadAl: begin
            LoadAlRun;
            BackButton.Enabled := True;
            ForwardButton.Caption := 'Загрузить данные об O';
            ProgramStage := LoadO;
        end; {LoadAl: ...}
        LoadO: begin
            LoadORun;
            ForwardButton.Caption := 'Выполнить расчёт';
            ProgramStage := Calc;
        end; {LoadO: ...}
        Calc: begin
            CalcRun;
            ForwardButton.Caption := 'Сохранить данные о связях';
            ProgramStage := SaveLinks;
        end; {Calc: ...}
        SaveLinks: begin
            SaveLinksRun;
            ForwardButton.Caption := 'Выход';
            ProgramStage := Finish;
        end; {SaveLinks: ...}
        Finish: Close;
    end; {case ProgramStage of ...}
end; {procedure TMainForm.ForwardButtonClick}
{-----}
procedure TMainForm.BackButtonClick(Sender: TObject);
begin
    case ProgramStage of
        LoadO: begin
            CliaAl;
            BackButton.Enabled := False;
            ForwardButton.Caption := 'Загрузить данные об Al';
            ProgramStage := LoadAl;
        end; {LoadO: ...}
        Calc: begin
            CliaO;
            ForwardButton.Caption := 'Загрузить данные об O';
            ProgramStage := LoadO;
        end; {Calc: ...}
        SaveLinks: begin
            CliaLinks;
            ForwardButton.Caption := 'Выполнить расчёт';
            ProgramStage := Calc;
        end; {SaveLinks: ...}
        Finish: begin
            ForwardButton.Caption := 'Сохранить данные о связях';
            ProgramStage := SaveLinks;
        end;
    end;
end;

```

```

        end; {Finish: ...}
    end; {case ProgramStage of ...}
end; {procedure TMainForm.BackButtonClick}
{=====}
    {Обработчики событий создания и удаления формы}
{=====}
procedure TMainForm.FormCreate(Sender: TObject);
begin
    DCBuff := DecimalSeparator;
    DecimalSeparator := '.';
    AlDataList := TList.Create;
    ODataList := TList.Create;
    LinksDataList := TList.Create;
    FileOpenDialog.InitialDir := ExtractFileDir(Application.ExeName);
    FileSaveDialog.InitialDir := ExtractFileDir(Application.ExeName);
    AlStringGrid.Cells[1,0] := 'X';
    AlStringGrid.Cells[2,0] := 'Y';
    AlStringGrid.Cells[3,0] := 'Z';
    OStringGrid.Cells[1,0] := 'X';
    OStringGrid.Cells[2,0] := 'Y';
    OStringGrid.Cells[3,0] := 'Z';
    LinksStringGrid.Cells[1,0] := 'Атом А1';
    LinksStringGrid.Cells[2,0] := 'Атом О';
    LinksStringGrid.Cells[3,0] := 'Длина связи';
    LinksStringGrid.Cells[4,0] := 'Long/Short';
end; {procedure TMainForm.FormCreate}
{-----}
procedure TMainForm.FormDestroy(Sender: TObject);
var I: Integer;
    PA1: PA1Rec;
    PO: PA1Rec;
    PLink: PLinkRec;
begin
    for I := 0 to AlDataList.Count-1 do
        begin
            PA1 := AlDataList[I];
            Dispose(PA1);
        end; {for I := 0 to AlDataList.Count-1 do ...}
    AlDataList.Free;
    for I := 0 to ODataList.Count-1 do
        begin
            PO := ODataList[I];
            Dispose(PO);
        end; {for I := 0 to ODataList.Count-1 do ...}
    ODataList.Free;
    for I := 0 to LinksDataList.Count-1 do
        begin
            PLink := LinksDataList[I];
            Dispose(PLink);
        end; {for I := 0 to LinksDataList.Count-1 do ...}
    LinksDataList.Free;
    DecimalSeparator := DCBuff;
end; {procedure TMainForm.FormDestroy}
{-----}
procedure TMainForm.AngleControlButtonClick(Sender: TObject);
var PA1: PA1Rec;
    PO: PA1Rec;
    PLink: PLinkRec;
    NLink: Integer; {номер анализируемой связи в списке}
    S: String;

```

```

Al,O: array[1..3] of Double; {координаты атомов Al и O}
Lx,Ly,Lz,ML: Double; {проекции вектора связи на оси и его модуль}
CosX,CosY,CosZ: Double; {направляющие косинусы вектора связи}
FiX,FiY,FiZ: Double; {углы вектора связи с осями}
begin
  AngleControlMemo.Clear;
  S := InputBox('Выбор связи','Введите номер связи в списке','0');
  NLinc := StrToInt(S);
  PLink := LinksDataList[NLinc];
  (* ShowMessage(IntToStr(PLink^.AlNum *)
  PAL := AlDataList[PLink^.AlNum];
  Al[1] := PAL^.X; Al[2] := PAL^.Y; Al[3] := PAL^.Z;
  AngleControlMemo.Lines.Add('Атом Al X = '+FloatToStrF(Al[1],ffFixed,6,4)
                               +' Y = '+FloatToStrF(Al[2],ffFixed,6,4)
                               +' Z = '+FloatToStrF(Al[3],ffFixed,6,4));

  PO := ODataList[PLink^.ONum];
  O[1] := PO^.X; O[2] := PO^.Y; O[3] := PO^.Z;
  AngleControlMemo.Lines.Add('Атом O: X = '+FloatToStrF(O[1],ffFixed,6,4)
                               +' Y = '+FloatToStrF(O[2],ffFixed,6,4)
                               +' Z = '+FloatToStrF(O[3],ffFixed,6,4));

  Lx := Al[1]-O[1]; Ly := Al[2]-O[2]; Lz := Al[3]-O[3];
  AngleControlMemo.Lines.Add('Прекции связи: X =
'+FloatToStrF(Lx,ffFixed,6,4)
                               +' Y = '+FloatToStrF(Ly,ffFixed,6,4)
                               +' Z =
'+FloatToStrF(Lz,ffFixed,6,4));
  ML := Sqrt(Sqr(Lx)+Sqr(Ly)+Sqr(Lz));
  AngleControlMemo.Lines.Add('Длина связи: '+FloatToStrF(ML,ffFixed,6,4));
  CosX := Lx/ML; CosY := Ly/ML; CosZ := Lz/ML;
  AngleControlMemo.Lines.Add('Напр cos связи: X = '
                               +'FloatToStrF(CosX,ffFixed,6,4)
                               +' Y = '+FloatToStrF(CosY,ffFixed,6,4)
                               +' Z =
'+FloatToStrF(CosZ,ffFixed,6,4));
  FiX := ArcCos(CosX); FiY := ArcCos(CosY); FiZ := ArcCos(CosZ);
  AngleControlMemo.Lines.Add('Углы связи: X = '
                               +'FloatToStrF(RadToDeg(FiX),ffFixed,6,4)
                               +' Y = '+FloatToStrF(RadToDeg(FiY),ffFixed,6,4)
                               +' Z =
'+FloatToStrF(RadToDeg(FiZ),ffFixed,6,4));
  end; {procedure TMainForm.AngleControlButtonClick}

end.

```

Расчет поверхностных энергий кристаллов сапфира для большого блока ячеек

```

unit MainU;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Al2O3U, AtomsAndLinksU, PlaneU, ExtCtrls,
  Buttons, Math, CalcU, DateUtils, BPProgonU, TeEngine, Series, TeeProcs,
  Chart;
type
  TMainForm = class(TForm)
    MainPageControl: TPageControl;
    StartTabSheet: TTabSheet;
    LoadDataButton: TButton;

```

```

AlAtomsLabel: TLabel;
OxigenAtomsLabel: TLabel;
LinksAtomsLabel: TLabel;
BlockDimsLabel: TLabel;
AlfaResultsLabel: TLabel;
DataFileOpenDialog: TOpenDialog;
ParamsMemo: TMemo;
OnePlaneGroupBox: TGroupBox;
Label1: TLabel;
AlfaLabeledEdit: TLabeledEdit;
BetaLabeledEdit: TLabeledEdit;
GammaLabeledEdit: TLabeledEdit;
Label2: TLabel;
BPXLabeledEdit: TLabeledEdit;
BPYLabeledEdit: TLabeledEdit;
BPZLabeledEdit: TLabeledEdit;
CalcOnePlaneButton: TBitBtn;
ResultsMemo: TMemo;
ProgonGroupBox: TGroupBox;
BPNLabeledEdit: TLabeledEdit;
ZBPProgonGrafButton: TButton;
ZBPProgonButton: TButton;
YBPProgonGrafButton: TButton;
YBPProgonButton: TButton;
ZProgonTabSheet: TTabSheet;
ZBPProgonLinksChart: TChart;
ZBPProgonLinksSeries: TLineSeries;
ZBPProgonLongLinksSeries: TLineSeries;
ZBPProgonShortLinksSeries: TLineSeries;
ZBPProgonAtomsChart: TChart;
ZBPProgonAllLongSeries: TLineSeries;
ZBPProgonOxLongSeries: TLineSeries;
ZBPProgonEChart: TChart;
ZBPProgonESeries: TLineSeries;
ZBProgonAlShortSeries: TLineSeries;
ZBPProgonOxShortSeries: TLineSeries;
procedure LoadDataButtonClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure CalcOnePlaneButtonClick(Sender: TObject);
procedure ZBPProgonGrafButtonClick(Sender: TObject);
procedure ZBPProgonButtonClick(Sender: TObject);
procedure YBPProgonGrafButtonClick(Sender: TObject);
procedure YBPProgonButtonClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  MainForm: TMainForm;
{*****}
implementation
{$R *.dfm}
var DSBuff: char;
{=====}
                {Обработчики событий создания и удаления формы.}
{=====}
procedure TMainForm.FormCreate(Sender: TObject);
begin

```

```

DSBuff := DecimalSeparator;
DecimalSeparator := '.';
ParamsMemo.Lines.Add('rO = '+FloatToStrF(rO, ffFixed, 6, 4));
ParamsMemo.Lines.Add('rAl = '+FloatToStrF(rAl, ffFixed, 6, 4));
ParamsMemo.Lines.Add('qAl/qOx = '+FloatToStrF(qAl_D_qOx, ffFixed, 6, 4));
ParamsMemo.Lines.Add('lLong = '+FloatToStrF(lLong, ffFixed, 6, 4));
ParamsMemo.Lines.Add('lShort = '+FloatToStrF(lShort, ffFixed, 6, 4));
ParamsMemo.Lines.Add('Emid = '+FloatToStr(Emid));
ParamsMemo.Lines.Add('eAlLong = '+FloatToStr(eAlLong));
ParamsMemo.Lines.Add('eALShort = '+FloatToStr(eALShort));
ParamsMemo.Lines.Add('eOxLong = '+FloatToStr(eOxLong));
ParamsMemo.Lines.Add('eOxShort = '+FloatToStr(eOxShort));
end; {procedure TMainForm.FormCreate}
{-----}
procedure TMainForm.FormDestroy(Sender: TObject);
begin
    DecimalSeparator := DSBuff;
end; {procedure TMainForm.FormDestroy}
{=====}
    {Обработчики событий элементов управления, страница StartTabSheet.}
{=====}
procedure TMainForm.LoadDataButtonClick(Sender: TObject); {для каждого файла
вызывается процедура загрузки из модуля AtomsAndLinksU; параметры: диалог и
метка для отображения результата}
begin
    DataFileOpenDialog.InitialDir := ExtractFileDir(Application.ExeName);
    LoadAlFile(DataFileOpenDialog, AlAtomsLabel);           {4 процедуры}
    LoadOxigenFile(DataFileOpenDialog, OxigenAtomsLabel); {из модуля}
    LoadLinksFile(DataFileOpenDialog, LinksAtomsLabel);   {AtomsAndLinksU}
    LoadDimsFile(DataFileOpenDialog, BlockDimsLabel);
    OnePlaneGroupBox.Enabled := True;
end; {procedure TMainForm.LoadDataButtonClick}
{-----}
procedure TMainForm.CalcOnePlaneButtonClick(Sender: TObject); {Для одной
плоскости, перпендикулярной плоскости блока и наклонённой на угол Alfa к оси Z
(Beta=Gamma=0). Задаются Alfa и координаты Z и Y базовой точкой (полагается
X=0).}
begin
    Screen.Cursor := crHourGlass;
    Alfa := StrToFloat(AlfaLabeledEdit.Text); {принимаем параметры}
    Beta := 0; {Beta=Gamma=0 независимо от ввода}
    Gamma := 0;
    PlaneNCos[3] := Cos(DegToRad(Alfa));
    PlaneNCos[2] := -Sqrt(1-Sqr(PlaneNCos[3])); {не зависит от Beta и Gamma}
    PlaneNCos[1] := 0; {не зависит от Beta и Gamma}
    BasePoint[3] := StrToFloat(BPZLabeledEdit.Text);
    BasePoint[2] := StrToFloat(BPYLabeledEdit.Text);
    BasePoint[1] := 0;
    SectCalc; {процедура из модуля CalcU, основные вычисления}
    ResultsMemo.Clear; {вывод результатов}
    ResultsMemo.Lines.Add('Результаты для одной плоскости');
    ResultsMemo.Lines.Add('Вектор нормали ('
        +FloatToStrF(PlaneNCos[1], ffGeneral, 6, 4)+' , '
        +FloatToStrF(PlaneNCos[2], ffGeneral, 6, 4)+' , '
        +FloatToStrF(PlaneNCos[3], ffGeneral, 6, 4)+' )');
    ResultsMemo.Lines.Add('Базовая точка ('
        +FloatToStrF(BasePoint[1], ffGeneral, 6, 4)+' , '
        +FloatToStrF(BasePoint[2], ffGeneral, 6, 4)+' , '
        +FloatToStrF(BasePoint[3], ffGeneral, 6, 4)+' )');
    ResultsMemo.Lines.Add('Площадь сечения, м^2 '+FloatToStr(PlaneS));

```

```

ResultsMemo.Lines.Add('Параметр плоскости P, A '
  +FloatToStrF(PlaneP,ffFixed,6,4));
ResultsMemo.Lines.Add('Число разорванных связей');
ResultsMemo.Lines.Add(' ненормированное '+IntToStr(NSectLinks)
  +', нормированное '+FloatToStr(NSectLinksDS));
ResultsMemo.Lines.Add('Число разорванных длинных связей');
ResultsMemo.Lines.Add(' ненормированное '+IntToStr(NLongSectLinks)
  +', нормированное '+FloatToStr(NLongSectLinksDS));
ResultsMemo.Lines.Add('Число разорванных коротких связей');
ResultsMemo.Lines.Add(' ненормированное '+IntToStr(NShortSectLinks)
  +', нормированное '+FloatToStr(NShortSectLinksDS));
ResultsMemo.Lines.Add('Число разорванных связей от атомов AL');
ResultsMemo.Lines.Add(' ненормированное '+IntToStr(NAL)
  +', нормированное '+FloatToStr(NALDS));
ResultsMemo.Lines.Add('Число разорванных связей от атомов O');
ResultsMemo.Lines.Add(' ненормированное '+IntToStr(NOx)
  +', нормированное '+FloatToStr(NOxDs));
ResultsMemo.Lines.Add('Поверхностная энергия');
ResultsMemo.Lines.Add(' ненормированная '+FloatToStr(E0)
  +', нормированная '+FloatToStr(E));
(*ResultsMemo.Lines.Add('Время работы SectCalc '
  +IntToStr(MilliSecondsBetween(CalcStop,CalcStart))); {!!! Использовалось
при определении времени работы процедуры !!!}*)
Screen.Cursor := crDefault;
end; {procedure TMainForm.CalcOnePlaneButtonClick}
{=====}
procedure TMainForm.ZBPProgonGrafButtonClick(Sender: TObject);
begin
Screen.Cursor := crHourGlass;
Alfa := StrToFloat(AlfaLabeledEdit.Text);{принимаем параметры}
Beta := 0; {Beta=Gamma=0 независимо от ввода}
Gamma := 0;
PlaneNCos[3] := Cos(DegToRad(Alfa));
PlaneNCos[2] := -Sqrt(1-Sqr(PlaneNCos[3])); {не зависит от Beta и Gamma}
PlaneNCos[1] := 0; {не зависит от Beta и Gamma}
{BasePoint[3] не принимаем - будем прогонять}
BasePoint[2] := StrToFloat(BPYLabeledEdit.Text);
BasePoint[1] := 0;
BPN := StrToInt(BPNLabeledEdit.Text);
ZBPProgonPrep; {процедура из модуля VPProgonU}
ZBPProgonGraf; {процедура из модуля VPProgonU - основные вычисления}

Screen.Cursor := crDefault;
end; {procedure TMainForm.ZBPProgonGrafButtonClick}
{-----}
procedure TMainForm.ZBPProgonButtonClick(Sender: TObject);
begin
Screen.Cursor := crHourGlass;
Alfa := StrToFloat(AlfaLabeledEdit.Text);{принимаем параметры}
Beta := 0; {Beta=Gamma=0 независимо от ввода}
Gamma := 0;
PlaneNCos[3] := Cos(DegToRad(Alfa));
PlaneNCos[2] := -Sqrt(1-Sqr(PlaneNCos[3])); {не зависит от Beta и Gamma}
PlaneNCos[1] := 0; {не зависит от Beta и Gamma}
{BasePoint[3] не принимаем - будем прогонять}
BasePoint[2] := StrToFloat(BPYLabeledEdit.Text);
BasePoint[1] := 0;
BPN := StrToInt(BPNLabeledEdit.Text);
BPStep := c/BPN; {шага прогонки}
BPBase := (DZ-c)/2; {ниже середины блока на c/2}

```

```

ZBPProgon; {процедура из модуля BPProgonU - основные вычисления}

Screen.Cursor := crDefault;
end; {procedure TMainForm.ZBPProgonButtonClick}
{-----}
procedure TMainForm.YBPProgonGrafButtonClick(Sender: TObject);
begin
Screen.Cursor := crHourGlass;
Alfa := StrToFloat(AlfaLabeledEdit.Text); {принимаем параметры}
Beta := 0; {Beta=Gamma=0 независимо от ввода}
Gamma := 0;
PlaneNCos[3] := Cos(DegToRad(Alfa));
PlaneNCos[2] := -Sqrt(1-Sqr(PlaneNCos[3])); {не зависит от Beta и Gamma}
PlaneNCos[1] := 0; {не зависит от Beta и Gamma}
BasePoint[3] := StrToFloat(BPZLabeledEdit.Text);
{BasePoint[2] не принимаем - будем прогонять}
BasePoint[1] := 0;
BPN := StrToInt(BPNLabeledEdit.Text);
YBPProgonPrep; {процедура из модуля BPProgonU}
YBPProgonGraf; {процедура из модуля BPProgonU - основные вычисления}

Screen.Cursor := crDefault;
end; {procedure TMainForm.YBPProgonGrafButtonClick}
{-----}
procedure TMainForm.YBPProgonButtonClick(Sender: TObject);
begin
Screen.Cursor := crHourGlass;
Alfa := StrToFloat(AlfaLabeledEdit.Text); {принимаем параметры}
Beta := 0; {Beta=Gamma=0 независимо от ввода}
Gamma := 0;
PlaneNCos[3] := Cos(DegToRad(Alfa));
PlaneNCos[2] := -Sqrt(1-Sqr(PlaneNCos[3])); {не зависит от Beta и Gamma}
PlaneNCos[1] := 0; {не зависит от Beta и Gamma}
BasePoint[3] := StrToFloat(BPZLabeledEdit.Text);
{BasePoint[2] не принимаем - будем прогонять}
BasePoint[1] := 0;
BPN := StrToInt(BPNLabeledEdit.Text);
BPStep := c/BPN; {шага прогонки}
BPBase := (DY-c)/2; {левее середины блока на c/2}
YBPProgon; {процедура из модуля BPProgonU - основные вычисления}

Screen.Cursor := crDefault;
end; {procedure TMainForm.YBPProgonButtonClick}
{-----}
end.

{*****}
{Версия модуля для программы BigBlocksCalc_V2. Содержит кристаллографические
константы и матрицы Al2O3 и константы для расчёта поверхностной энергии.}
{*****}
unit Al2O3U;
interface
uses SysUtils, Dialogs, Math;
{Определение векторов и матриц}
type TInd3 = 1..3;
TInd4 = 1..4;
TIntVect3 = array[TInd3] of Integer;
TIntVect4 = array[TInd4] of Integer;
PRealVect3 = ^TRealVect3;
TRealVect3 = array[TInd3] of Double;

```

```

TMatrix3x3 = array[TInd3,TInd3] of Double;
{=====}
const cR3 = 1.7320508075688772935274463415059; {корень из 3}
{=====}
    {Параметры решётки, длины и энергии связей}
const a = 4.75; c = 12.97(*6.485*); cda = c/a; {cda = 2.7305}
    ainv = 2/a/cR3; cinv = 1/c; {ainv = 0.2431, cinv = 0.0771}
    rOp = 2*a/cR3; rOpD2 = a/cR3; {радиус окружности, описанной вокруг
        шестиугольника основания (он же - сторона основания), rOp = 5.4848
        и его половина rOpD2 = 2.7424}
    s0 = 2*cR3*a*a; {площадь основания ячейки 78.1588 А^2}
    dZAl = 0.243; {смещение атомов Al от Z0}
    rO = 1.4; rAl = 0.57; {радиусы ионов O и Al}
    r3O = 1.452; {радиус на котором расположены 3 сближенных атома O}
    cd6 = c/6; {расстояние между слоями кислорода}
    cd12 = c/12; {смещение первого слоя от основания ячейки}
    lLong = 1.9649; lShort = 1.8531; {длины связей ("длинной" и "короткой") Al-
O}
    Emid = 8.33e-19; {средняя энергия связи}
{-----}
const qAl_D_qOx = 1; {отношение зарядов Al/Oxygen}
{-----}
    {Энергии оборванных длинных и коротких связей от атомов Al и O}
const lLong_D_lShort = lLong/lShort; {отношение длин связей}
    sqr_lLong_D_lShort = lLong_D_lShort*lLong_D_lShort; {то же в квадрате}
    e_Denominator =
(1+sqr_lLong_D_lShort)*(qAl_D_qOx*rO*rO+rAl*rAl);{знаменатель в выражениях для
констант энергий связи}
    eAllong = 2*Emid*qAl_D_qOx*rO*rO/e_Denominator; {длинная связь от Al}
    eAlShort = eAllong*sqr_lLong_D_lShort; {короткая связь от Al}
    eOxLong = 2*Emid*rAl*rAl/e_Denominator; {длинная связь от O}
    eOxShort = eOxLong*sqr_lLong_D_lShort; {короткая связь от O}
{=====}
    {Кристаллографические матрицы}
const mA: TMatrix3x3 = ((a,0,0), (-a/2,a*cR3/2,0), (0,0,c));
    mE: TMatrix3x3 = ((ainv*cR3/2,0,0), (ainv/2,ainv,0), (0,0,cinv));
    {
        |0.2105  0  0 |}
    {mE = |0.1215 0.2431  0 |}
    {
        |0  0  0.0771|}
    mG: TMatrix3x3 = ((a*a,-a*a/2,0), (-a*a/2,a*a,0), (0,0,c*c));
    mGinv: TMatrix3x3 = ((ainv*ainv,ainv*ainv/2,0), (ainv*ainv/2,ainv*ainv,0),
        (0,0,cinv*cinв));
    {
        |0.0591 0.0295  0 |}
    {mGinv = |0.1215 0.0591  0 |}
    {
        | 0  0  0.0059|}
{=====}
    {Вспомогательные величины}
const z0 = 7*c/2; {середина столбика из 7 ячеек}
    {Межплоскостные расстояния, ромбоэдры}
    dq = 2.1942; dRR = 2.5465; dr = 3.4737; dS = 3.9211; dt = 2.0314;
    {Межплоскостные расстояния, пирамиды}
    dP = 1.5986; dn = 2.0816; dw = 2.2301; du = 2.3362;
{=====}
    {Вспомогательные функции}
function PlaneDist(Bukva: String): Double; {возвращает межплоскостное
расстояние по буквенному обозначению плоскости}
function PlaneDistCalc(MillerInd: TIntVect3): Double; {вычисляет
межплоскостное расстояние по индексам Миллера плоскости}
function MulVectConst(Vect: TRealVect3; CC: Double): TRealVect3;
function MulMatrixVect(Matrix: TMatrix3x3; Vect: TRealVect3): TRealVect3;

```



```

overload;
function MulMatrixVect(Matrix: TMatrix3x3; Vect: TIntVect3): TRealVect3;
overload;
(*procedure DivEmid; {вычисляет OFactor, AlFactor, ELong, EShort по
первоначальным
или изменённым QAl_D_QOx и N_rAl_D_rOx} *)
{=====}
{Функции преобразования векторов}
function MillerPlaneToCos(MillerInd: TIntVect3): TRealVect3; {переводит
индексы Миллера нормали к плоскости в её направляющие косинусы}
function PlaneNCosToMiller(NCos: TRealVect3): TIntVect3; {переводит
направляющие косинусы нормали к плоскости в её индексы Миллера}
function BravePlaneToMiller(BraveInd: TIntVect4): TIntVect3; {переводит
индексы Бравэ нормали к плоскости в её индексы Миллера}
function MillerPlaneToBrave(MillerInd: TIntVect3): TIntVect4; {переводит
индексы Миллера нормали к плоскости в её индексы Бравэ}
function BravePlaneToCos(BraveInd: TIntVect4): TRealVect3; {переводит индексы
Бравэ нормали к плоскости в её направляющие косинусы}
function PlaneNCosToBrave(NCos: TRealVect3): TIntVect4; {переводит
направляющие косинусы нормали к плоскости в её индексы Бравэ}
{*****}
implementation
{=====}
{Вспомогательные функции}
{=====}
function PlaneDist(Bukva: String): Double; {возвращает межплоскостное
расстояние по буквенному обозначению плоскости}
begin
if Bukva = 'c.' then Result := c
else if (Bukva = 'q+') or (Bukva = 'q-') then Result := dq
else if (Bukva = 'R+') or (Bukva = 'R-') then Result := dRR
else if (Bukva = 'r+') or (Bukva = 'r-') then Result := dr
else if (Bukva = 'S+') or (Bukva = 'S-') then Result := dS
else if (Bukva = 't+') or (Bukva = 't-') then Result := dt
else if Bukva = 'P.' then Result := dP
else if Bukva = 'n.' then Result := dn
else if Bukva = 'w.' then Result := dw
else if Bukva = 'u.' then Result := du;
end; {function PlaneDist}
{-----}
function PlaneDistCalc(MillerInd: TIntVect3): Double; {вычисляет
межплоскостное расстояние по индексам Миллера плоскости}
var I, J: Integer;
D: Double;
begin
D := 0;
for I := 1 to 3 do
for J := 1 to 3 do
D := D+mGinv[I, J]*MillerInd[I]*MillerInd[J];
Result := 1/Sqrt(D);
end; {function PlaneDistCalc}
{-----}
function MulVectConst(Vect: TRealVect3; CC: Double): TRealVect3;
begin
Result[1] := CC*Vect[1];
Result[2] := CC*Vect[2];
Result[3] := CC*Vect[3];
end; {function MulVectConst}
{-----}
function MulMatrixVect(Matrix: TMatrix3x3; Vect: TRealVect3): TRealVect3;

```

```

overload;
var I,J: TInd3;
begin
  for I := 1 to 3 do
    begin
      Result[I] := 0;
      for J := 1 to 3 do
        Result[I] := Result[I]+Matrix[I,J]*Vect[J];
      end; {for I := 1 to 3 do ...}
    end; {function MulMatrixVect}
  }-----}
function MulMatrixVect(Matrix: TMatrix3x3; Vect: TIntVect3): TRealVect3;
overload;
var I,J: TInd3;
begin
  for I := 1 to 3 do
    begin
      Result[I] := 0;
      for J := 1 to 3 do
        Result[I] := Result[I]+Matrix[I,J]*Vect[J];
      end; {for I := 1 to 3 do ...}
    end; {function MulMatrixVect}
  }=====}
  {Функции преобразования векторов}
  }=====}
function MillerPlaneToCos(MillerInd: TIntVect3): TRealVect3; {переводит
индексы Миллера нормали к плоскости в её направляющие косинусы}
var I,J: TInd3;
  M: Double;
begin
  (* ShowMessage('cda = '+FloatToStrF(cda, ffFixed, 6, 4)+#13#10
    +'ainv = '+FloatToStrF(ainv, ffFixed, 6, 4)+#13#10
    +'cinv = '+FloatToStrF(cinv, ffFixed, 6, 4)); *)
  (* ShowMessage('mGinv'+#13#10+FloatToStrF(mGinv[1,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[1,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[1,3], ffFixed, 6, 4)+' '+#13#10
    +FloatToStrF(mGinv[2,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[2,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[2,3], ffFixed, 6, 4)+' '+#13#10
    +FloatToStrF(mGinv[3,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[3,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mGinv[3,3], ffFixed, 6, 4)); *)
  (* ShowMessage('mE'+#13#10+FloatToStrF(mE[1,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[1,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[1,3], ffFixed, 6, 4)+' '+#13#10
    +FloatToStrF(mE[2,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[2,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[2,3], ffFixed, 6, 4)+' '+#13#10
    +FloatToStrF(mE[3,1], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[3,2], ffFixed, 6, 4)+' '
    +FloatToStrF(mE[3,3], ffFixed, 6, 4)); *)
  M := 0;
  for I := 1 to 3 do
    for J := 1 to 3 do
      begin
        M := M+mGinv[I,J]*MillerInd[I]*MillerInd[J];
        (* ShowMessage(IntToStr(I)+' '+IntToStr(J)+' '+FloatToStr(M)); *)
      end;
    end;
  M := Sqrt(M);
  (* ShowMessage('M = '+FloatToStr(M)); *)

```

```

Result := MulMatrixVect(mE,MillerInd);
(* ShowMessage('Cos*M = '+FloatToStrF(Result[1],ffFixed,6,4)+' '
              +FloatToStrF(Result[2],ffFixed,6,4)+' '
              +FloatToStrF(Result[3],ffFixed,6,4)); *)
Result := MulVectConst(Result,1/M);
(* ShowMessage('Cos = '      +FloatToStrF(Result[1],ffFixed,6,4)+' '
              +FloatToStrF(Result[2],ffFixed,6,4)+' '
              +FloatToStrF(Result[3],ffFixed,6,4)); *)
end; {function MillerPlaneToCos}
{-----}
-}
function PlaneNCosToMiller(NCos: TRealVect3): TIntVect3; {переводит
направляющие косинусы нормали к плоскости в её индексы Миллера}
var I: Integer;
    N: TRealVect3;
    M: Double;
begin
N := MulMatrixVect(mA,NCos);
if N[1] <> 0 then
begin
I := 1;
M := Abs(N[1]);
if (N[2] <> 0) and (Abs(N[2])<M) then
begin
I := 2;
M := Abs(N[2]);
if (N[3] <> 0) and (Abs(N[3])<M) then
begin
I := 3;
M := Abs(N[3]);
end; {if (N[3] <> 0) and (Abs(N[3])<M) then ...}
end; {if (N[2] <> 0) and (Abs(N[2])<M) then ...}
end {if N[1] <> 0 then ...}
else
if N[2] <> 0 then
begin
I := 2;
M := Abs(N[2]);
if (N[3] <> 0) and (Abs(N[3])<M) then
begin
I := 3;
M := Abs(N[3]);
end; { if (N[3] <> 0) and (Abs(N[3])<M) then ...}
end {if N[2] <> 0 then}
else
begin
I := 3;
M := Abs(N[3]);
end; {if N[2] <> 0 then ... else ...}
{В этом месте M - минимальное по модулю ненулевое значение в векторе N}
Result[1] := Round(N[1]/M);
Result[2] := Round(N[2]/M);
Result[3] := Round(N[3]/M);
end; {function MillerPlaneToCos}
{-----}
function BravePlaneToMiller(BraveInd: TIntVect4): TIntVect3; {переводит
индексы Бравэ нормали к плоскости в её индексы Миллера}
begin
Result[1] := BraveInd[1];
Result[2] := BraveInd[2];

```

```

    Result[3] := BraveInd[4];
  end; {function BravePlaneToMiller}
}-----}
function MillerPlaneToBrave(MillerInd: TIntVect3): TIntVect4; {переводит
индексы Миллера нормали к плоскости в её индексы Бравэ}
begin
  Result[1] := MillerInd[1];
  Result[2] := MillerInd[2];
  Result[3] := -MillerInd[1]-MillerInd[2];
  Result[4] := MillerInd[3];
end; {function MillerPlaneToBrave}
}-----}
function BravePlaneToCos(BraveInd: TIntVect4): TRealVect3; {переводит индексы
Бравэ нормали к плоскости в её направляющие косинусы}
var MillerInd: TIntVect3;
begin
  MillerInd := BravePlaneToMiller(BraveInd);
  Result := MillerPlaneToCos(MillerInd);
end; {function BravePlaneToCos}
}-----}
function PlaneNCosToBrave(NCos: TRealVect3): TIntVect4; {переводит
направляющие косинусы нормали к плоскости в её индексы Бравэ}
var MillerInd: TIntVect3;
begin
  MillerInd := PlaneNCosToMiller(NCos);
  Result := MillerPlaneToBrave(MillerInd);
end; {function PlaneNCosToBrave}
end.

{*****}
{Модуль содержит параметры, относящиеся к секущей плоскости: направляющие
косинусы нормали, координаты базовой точки, угол между нормалью и осью Z=c,
параметр р и площадь сечения S блока плоскостью. Модуль также содержит
процедуры вычисления S и р по нормали и базовой точке. Далее модуль вводит все
параметры, используемые при накоплении E/ В начале все вектора задаются
нулевыми. Значения Alfa и BasePoint изменяются в зависимости от варианта
расчёта.}
{*****}
unit PlaneU;
interface
uses StdCtrls, SysUtils, Dialogs, Math, AtomsAndLinksU, Al2O3U;
var PlaneNCos: TRealVect3; {направляющие косинусы нормали секущей плоскости}
    BasePoint: TRealVect3; {базовая точка секущей плоскости}
    Alfa, Beta, Gamma: Double; {углы между нормалью к плоскости сечения и осями
Z, Y, X соответственно, градусы}
    PlaneS: Double; {площадь сечения блока плоскостью}
    PlaneP: Double; {параметр р плоскости}
procedure CalcPlaneS; {процедура вычисления площади}
procedure CalcPlaneP; {процедура вычисления параметра P}
var NSectLinks, NLongSectLinks, NShortSectLinks: Integer; {число разрываемых
связей, в том числе длинных и коротких}
    NSectLinksDS, NLongSectLinksDS, NShortSectLinksDS: Double; {тоже,
нормированные на площадь (шт/м^2)}
    NAl, NOx: Integer; {число связей от атомов Al и O на поверхности сечения}
    NAlDS, NOxDS: Double; {тоже, нормированные на площадь (шт/м^2)}
    NAlLong, NAlShort, NOxLong, NOxShort: Integer;
    NAlLongDS, NAlShortDS, NOxLongDS, NOxShortDS: Double; {тоже, нормированные на
площадь (шт/м^2)}
    E, E0: Double; {поверхностная энергия нормированная (E) и ненормированная
(E0) на площадь}

```

```

    {Внутренние переменные процедурыCalcPlaneS. Выведены в interface для
    проверки работы программы}
    var Beta0,Fi1,Fi2: Double; {Beta = 180-Alfa, используется при Alfa>90, и
    углы наклона линий, проведённых из опорной точки в углы блока к оси Z=c,
    градусы}
    S1,S2: Double; {участки линии сечения блока плоскостью до и после
    опорнойточки}
    {*****}
implementation
procedure CalcPlaneS; {процедура вычисления площади}
(* var Beta0,Fi1,Fi2: Double; {Brta0= 180-Alfa, используется при Alfa>90, и
углы наклона линий, проведённых из опорной точки в углы блока к оси Z=c,
градусы}
S1,S2: Double; {участки линии сечения блока плоскостью до и после опорной
точки} Выведены в interface для проверки работы программы*)
begin
    if (Alfa=0) or (Alfa=180) then {выделяем горизонтальную плоскость}
    begin
        PlaneS := DX*DY*1E-20; {перевод из ангстрем^2 в м^2}
        Exit; {больше считать нечего}
    end; {if (Alfa=0) or (Alfa=180) then ...}
    if Alfa=90 then {выделяем вертикальную плоскость}
    begin
        PlaneS := DX*DZ*1E-20; {перевод из ангстрем^2 в м^2}
        Exit; {больше считать нечего}
    end; {if (Alfa=0) then ...}
    {Не вертикальная и не горизонтальная плоскость}
    if Alfa < 90 then
    begin
        if BasePoint[2]>0 then {чтобы не возникло деление на 0}
            Fi1 := RadToDeg(ArcTan(BasePoint[3]/BasePoint[2]))
        else Fi1 := 90;
        if Alfa <= Fi1 then S1 := BasePoint[2]/Cos(DegToRad(Alfa))
            else S1 := BasePoint[3]/Sin(DegToRad(Alfa));
        if BasePoint[2] < DY then
            Fi2 := RadToDeg(ArcTan((DZ-BasePoint[3])/(DY-BasePoint[2])))
        else Fi2 := 90; {чтобы не возникло деление на 0}
        if Alfa <= Fi2 then S2 := (DY-BasePoint[2])/Cos(DegToRad(Alfa))
            else S2 := (DZ-BasePoint[3])/Sin(DegToRad(Alfa));
        end {if Alfa < 90 then ...}
    else {Alfa>90, т.к. Alfa=90 уже обработано}
    begin
        Beta0 := 180 - Alfa;
        Fi1 := RadToDeg(ArcTan((DZ-BasePoint[3])/BasePoint[2]));
        if Beta0 <= Fi1 then S1 := BasePoint[2]/Cos(DegToRad(Beta0))
            else S1 := (DZ-BasePoint[3])/Sin(DegToRad(Beta0));
        Fi2 := RadToDeg(ArcTan(BasePoint[3]/(DY-BasePoint[2])));
        if Beta0 <= Fi2 then S2 := (DY-BasePoint[2])/Cos(DegToRad(Beta0))
            else S2 := BasePoint[3]/Sin(DegToRad(Beta0));
        end; {if Alfa < 90 then ... else ...}
        PlaneS := DX*(S1+S2)*1E-20; {полная площадь, перевод из ангстрем^2 в м^2}
    end; {procedure CalcPlaneS}
}-----}
procedure CalcPlaneP; {процедура вычисления параметра P}
begin
    PlaneP := PlaneNCos[1]*BasePoint[1]+PlaneNCos[2]*BasePoint[2]+
        +PlaneNCos[3]*BasePoint[3];
end; {procedure CalcPlaneP}

end.

```

```

{*****}
{Модуль содержит переменные и подпрограммы, связанные с прогонкой базовой
точки по Z и Y. Прогонка проводится на величину +-с/2. Для прогонки с
построением графиков берётся запас +-5% (). Эти процедуры также накапливают в
динамических массивах значения всех параметров во всех точках для возможности
сохранения. Процедуры без построения графиков () предназначены для
использования при прогонке по углам. В этом случае запас не берётся, массивы
не заполняются и результатом их работы является только значение минимальной
энергии и координата точки, в которой оно получено.}
{*****}
unit BPProgonU;
interface
uses Math, Al2O3U, AtomsAndLinksU, PlaneU, CalcU;
var BPN: Integer; {на единицу меньше числа точек при прогонке, т.к. нумерация
в массивах с 0}
  BPStep: Double; {шаг при прогонке = с/BPN}
  BPBase: Double; {начальная координата прогонки (Z или Y)}
  BPLinksAr, BPLongLinksAr, BPShortLinksAr, BPAlAr, BPOxAr: array of Double;
  BPAlLongAr, BPAlShortAr, BPOxLongAr, BPOxShortAr: array of Double;
  BPEAr: array of Double; {массивы для накопления результатов}
  EminI, EminIk: Integer; {номер точки с Emin и число точек с таким значением}
  Emin, EminPoz: Double; {значение в этой точке и её положение}
  Links_min, LongLinks_min, ShortLinks_min, Al_min, Ox_min: Double; {другие}
  AlLong_min, AlShort_min, OxLong_min, OxShort_min: Double; {результаты в точке
с Emin}
procedure ZBPProgonPrep; {вычисляет шаг, задаёт начальную координату,
устанавливает длины массивов}
procedure ZBPProgonGraf;
procedure ZBPProgon;
procedure YBPProgonPrep; {вычисляет шаг, задаёт начальную координату,
устанавливает длины массивов}
procedure YBPProgonGraf;
procedure YBPProgon;
{*****}
implementation
{=====}
procedure ZBPProgonPrep; {вычисляет шаг, задаёт начальную координату,
устанавливает длины массивов}
begin
  BPStep := с/BPN; {при вычислении шага берём число точек без запаса}
  BPN := Round(1.1*BPStep); {вводим запас 10%}
  BPBase := (DZ-BPStep*BPN)/2; {ниже середины блока на с/2 с запасом}
  SetLength(BPLinksAr, BPN+1); {длины массивов}
  SetLength(BPLongLinksAr, BPN+1);
  SetLength(BPShortLinksAr, BPN+1);
  SetLength(BPAlAr, BPN+1);
  SetLength(BPOxAr, BPN+1);
  SetLength(BPAlLongAr, BPN+1);
  SetLength(BPAlShortAr, BPN+1);
  SetLength(BPOxLongAr, BPN+1);
  SetLength(BPOxShortAr, BPN+1);
  SetLength(BPEAr, BPN+1);
end; {procedure ZBPProgonPrep}
{=====}
procedure ZBPProgonGraf;
var I: Integer;
begin
  EminI := 0; {обнуляем переменные, связанные с Emin}
  EminIk := 0; EminPoz := 0;
  Emin := MaxDouble; {максимально возможное значение; любое E будет меньше}

```

```

for I := 0 to BPN do {цикл прогонки}
begin
  BasePoint[3] := VPBase+BPStep*I; {новое положение базовой точки}
  CalcPlaneS;
  CalcPlaneP;
  SectCalc; {основные вычисления вызовом процедуры из модуля CalcU}
  BPLinksAr[I] := NSectLinksDS; {фиксируем результаты в массивах для
возможности сохранения}
  BPLongLinksAr[I] := NLongSectLinksDS;
  BPShortLinksAr[I] := NShortSectLinksDS;
  BPA1Ar[I] := NA1DS;
  BPOxAr[I] := NOxDS;
  BPA1LongAr[I] := NA1LongDS;
  BPA1ShortAr[I] := NA1ShortDS;
  BPOxLongAr[I] := NOxLongDS;
  BPOxShortrAr[I] := NOxShortDS;
  BPEAr[I] := E;
  if E < Emin then {проверяем, не найдено ли новое меньшее значение E}
  begin
    EminI := I; {фиксируем все параметры в точке с новым найденным
Emin}
    EminPoz := BasePoint[3];
    Emin := E;
    EminIk := 1;
    Links_min := NSectLinksDS;
    LongLinks_min := NLongSectLinksDS;
    ShortLinks_min := NShortSectLinksDS;
    A1_min := NA1DS;
    Ox_min := NOxDS;
    A1Long_min := NA1LongDS;
    A1Short_min := NA1ShortDS;
    OxLong_min := NOxLongDS;
    OxShortr_min := NOxShortDS;
  end {if E < Emin then ...}
  else if E = Emin then EMinIk := EMinIk+1; {при повторе увеличиваем
кратность}
  end; {for I := 0 to BPN do ...}
  (*
  *)
end; {procedure ZBPProgonGraf;}
{=====}
procedure ZBPProgon;
var I: Integer;
begin
  EminI := 0; {обнуляем переменные, связанные с Emin}
  EMinIk := 0; EminPoz := 0;
  Emin := MaxDouble; {максимально возможное значение; любое E будет меньше}
  for I := 0 to BPN do {цикл прогонки}
  begin
    BasePoint[3] := VPBase+BPStep*I; {новое положение базовой точки}
    CalcPlaneS;
    CalcPlaneP;
    SectCalc; {основные вычисления вызовом процедуры из модуля CalcU}
    if E < Emin then {проверяем, не найдено ли новое меньшее значение E}
    begin
      EminI := I; {фиксируем все параметры в точке с новым найденным
Emin}
      EminPoz := BasePoint[3];
      Emin := E;
      EMinIk := 1;

```

```

Links_min := NSectLinksDS;
LongLinks_min := NLongSectLinksDS;
ShortLinks_min := NShortSectLinksDS;
Al_min := NAlDS;
Ox_min := NOxDS;
AlLong_min := NAlLongDS;
AlShort_min := NAlShortDS;
OxLong_min := NOxLongDS;
OxShortr_min := NOxShortDS;
end {if E < Emin then ...}
else if E = Emin then EMinIk := EMinIk+1; {при повторе увеличиваем
кратность}
end; {I := 0 to BPN do ...}
end; {procedure ZBPProgon;}
{=====}
procedure YBPProgonPrep; {вычисляет шаг, задаёт начальную координату,
устанавливает длины массивов}
begin
BPStep := c/BPN; {при вычислении шага берём число точек без запаса}
BPN := Round(1.1*BPN); {вводим запас 10%}
BPBase := (DY-BPStep*BPN)/2; {левее середины блока на c/2 с запасом}
SetLength(BPLinksAr,BPN+1);
SetLength(BPLongLinksAr,BPN+1);
SetLength(BPShortLinksAr,BPN+1);
SetLength(BPAlAr,BPN+1);
SetLength(BPOxAr,BPN+1);
SetLength(BPAlLongAr,BPN+1);
SetLength(BPAlShortAr,BPN+1);
SetLength(BPOxLongAr,BPN+1);
SetLength(BPOxShortrAr,BPN+1);
SetLength(BPEAr,BPN+1);
end; {procedure YBPProgonPrep;}
{=====}
procedure YBPProgonGraf;
var I: Integer;
begin
EMinI := 0; {обнуляем переменные, связанные с Emin}
EMinIk := 0; EMinPoz := 0;
EMin := MaxDouble; {максимально возможное значение; любое E будет меньше}
for I := 0 to BPN do {цикл прогонки}
begin
BasePoint[2] := BPBase+BPStep*I; {новое положение базовой точки}
CalcPlaneS;
CalcPlaneP;
SectCalc; {основные вычисления вызовом процедуры из модуля CalcU}
BPLinksAr[I] := NSectLinksDS; {фиксируем результаты в массивах для
возможности сохранения}
BPLongLinksAr[I] := NLongSectLinksDS;
BPShortLinksAr[I] := NShortSectLinksDS;
BPAlAr[I] := NAlDS;
BPOxAr[I] := NOxDS;
BPAlLongAr[I] := NAlLongDS;
BPAlShortAr[I] := NAlShortDS;
BPOxLongAr[I] := NOxLongDS;
BPOxShortrAr[I] := NOxShortDS;
BPEAr[I] := E;
if E < EMin then {проверяем, не найдено ли новое меньшее значение E}
begin
EMinI := I; {фиксируем все параметры в точке с новым найденным
EMin}

```



```

    EminPoz := BasePoint[3];
    Emin := E;
    EMinIk := 1;
    Links_min := NSectLinksDS;
    LongLinks_min := NLongSectLinksDS;
    ShortLinks_min := NShortSectLinksDS;
    Al_min := NAlDS;
    Ox_min := NOxDS;
    AlLong_min := NAlLongDS;
    AlShort_min := NAlShortDS;
    OxLong_min := NOxLongDS;
    OxShortr_min := NOxShortDS;
  end {if E < Emin then ...}
  else if E = Emin then EMinIk := EMinIk+1; {при повторе увеличиваем
кратность}
  end; {I := 0 to BPN do ...}
end; {procedure YBPProgonGraf;}
{=====}
procedure YBPProgon;
  var I: Integer;
  begin
    EminI := 0; {обнуляем переменные, связанные с Emin}
    EMinIk := 0; EminPoz := 0;
    Emin := MaxDouble; {максимально возможное значение; любое E будет меньше}
    for I := 0 to BPN do {цикл прогонки}
      begin
        BasePoint[2] := VPBase+VPStep*I; {новое положение базовой точки}
        CalcPlaneS;
        CalcPlaneP;
        SectCalc; {основные вычисления вызовом процедуры из модуля CalcU}
        if E < Emin then {проверяем, не найдено ли новое меньшее значение E}
          begin
            EminI := I; {фиксируем все параметры в точке с новым найденным
Emin}

            EminPoz := BasePoint[3];
            Emin := E;
            EMinIk := 1;
            Links_min := NSectLinksDS;
            LongLinks_min := NLongSectLinksDS;
            ShortLinks_min := NShortSectLinksDS;
            Al_min := NAlDS;
            Ox_min := NOxDS;
            AlLong_min := NAlLongDS;
            AlShort_min := NAlShortDS;
            OxLong_min := NOxLongDS;
            OxShortr_min := NOxShortDS;
          end {if E < Emin then ...}
          else if E = Emin then EMinIk := EMinIk+1; {при повторе увеличиваем
кратность}
          end; {I := 0 to BPN do ...}
        end; {procedure YBPProgon;}
      {*****}
    initialization
    finalization
      SetLength(BPLinksAr, 0);
      SetLength(BPLongLinksAr, 0);
      SetLength(BPShortLinksAr, 0);
      SetLength(BPAlAr, 0);
      SetLength(BPOxAr, 0);
      SetLength(BPAlLongAr, 0);

```

```

SetLength(BPAlShortAr,0);
SetLength(BPOxLongAr,0);
SetLength(BPOxShortAr,0);
SetLength(BPEAr,0);

```

end.

```

{*****}
{Модуль содержит программы вычислений:
1. Точки пересечения прямой и плоскости. function SectPointCalc(X1,X2:
TRealVect3): TRealVect3; Прямая задаётся двумя точками X1,X2: TRealVect3
(вектора из начала координат). Плоскость - плоскость сечения из модуля
SetPlaneU, заданная вектором нормали PlaneNCos: TRealVect3 и параметр PlaneP
(уравнение PlaneNCos[1]*X+PlaneNCos[2]*Y+PlaneNCos[3]*Z-PlaneP=0 относительно
координат X,Y,Z). Результат - точка TRealVect3.
2. Проверка принадлежности точки X отрезку (X1,X2). function
SectPointCheck(X,X1,X2: TRealVect3): Boolean;
3. procedure GetLinkPoints - определение конечных точек и направляющих
косинусов линии связи Al-O по указателю на запись в списке связей.
4. function SectLinkBuild создаёт запись о разорванной связи, вычисляет все её
поля и возвращает указатель на эту запись
5. procedure SectCalc - основные вычисления для заданной плоскости сечения,
без вывода на формы. Используются 4 предыдущих подпрограммы.}
{*****}
unit CalcU;
interface
uses Math, Al2O3U, AtomsAndLinksU, PlaneU, SysUtils, Dialogs;
(*В данной версии эта функция перенесены внутрь procedure SectCalc
function SectPointCalc(X1,X2: TRealVect3; var ParallelFlag:
Boolean):TRealVect3;
{вычисление координат точки пересечения прямой, заданной точками X1,X2 и
плоскости сечения; если прямая || плоскости, ParallelFlag := True, иначе
False, а результат полагается равным (0,0,0}*)
(*В данной версии эта процедура перенесены внутрь procedure SectCalc
procedure GetLinkPoints(ALink: PLinkRec; var X1,X2,LinkCos: TRealVect3);
{определение конечных точек X1,X2 и направляющих косинусов LinkCos линии связи
Al-O по указателю на запись в списке связей ALink} *)
(*В данной версии эта процедура перенесены внутрь procedure SectCalc
function SectPointCheck(X,X1,X2: TRealVect3): Boolean; {проверяет
принадлежность точки X отрезку (X1,X2)}*)
var CalcStart,CalcStop: TDateTime;
procedure SectCalc; {основные вычисления для заданной плоскости сечения
!!!Измеренное время работы процедуры от 45 до 67 мс !!!}
{*****}
implementation
(*В данной версии эта функция перенесены внутрь procedure SectCalc
function SectPointCalc(X1,X2: TRealVect3; var ParallelFlag:
Boolean):TRealVect3;
{вычисление координат точки пересечения прямой, заданной точками X1,X2 и
плоскости сечения; если прямая || плоскости, ParallelFlag := True, иначе
False, а результат полагается равным (0,0,0)}
var A,B,C,D,Det: Double;
begin
A := (X2[1]-X1[1])/(X2[3]-X1[3]);
C := (X2[2]-X1[2])/(X2[3]-X1[3]);
B := X1[1]-A*X1[3];
D := X1[2]-C*X1[3];
Det := PlaneNCos[1]*A+PlaneNCos[2]*C+PlaneNCos[3];
if Det=0 then

```

```

begin
  ParallelFlag := True;
  Result[3] := 0; Result[1] := 0; Result[2] := 0;
end {if Det=0 then ...}
else
begin
  ParallelFlag := False;
  Result[3] := (PlaneP-PlaneNCos[1]*B-PlaneNCos[2]*D)/Det;
  Result[1] := A*Result[3]+B;
  Result[2] := C*Result[3]+D;
end; {if Det=0 then ... else ...}
end; {function SectPointCalc}
{-----}*)
(*В данной версии эта процедура перенесены внутрь procedure SectCalc
procedure GetLinkPoints(ALink: PLinkRec; var X1,X2,LinkCos: TRealVect3);
{определение конечных точек X1,X2 и направляющих косинусов LinkCos линии связи
Al-O по указателю на запись в списке связей ALink}
var Al,O: Integer;
begin
  Al := ALink.AlNum; {номер атома Al для связи ALink}
  PAl := AlDataList[Al]; {указатель на запись данных этого атома}
  X1[1] := PAl.X; {первая точка}
  X1[2] := PAl.Y;
  X1[3] := PAl.Z;
  O := ALink.ONum; {номер атома O для связи ALink}
  POxygen := OxigenDataList[O]; {указатель на запись данных этого
атома}
  X2[1] := POxygen.X; {вторая точка}
  X2[2] := POxygen.Y;
  X2[3] := POxygen.Z;
  LinkCos[1] := (X2[1]-X1[1])/ALink.Length;
  LinkCos[2] := (X2[2]-X1[2])/ALink.Length;
  LinkCos[3] := (X2[3]-X1[3])/ALink.Length;
end; {procedure GetLinkPoints}*)
{-----}
(*В данной версии эта процедура перенесены внутрь procedure SectCalc
function SectPointCheck(X,X1,X2: TRealVect3): Boolean; {проверяет
принадлежность
точки X отрезку (X1,X2)}
var Flag1,Flag2,Flag3: Boolean; {результаты проверки по координатам
X[1],X[2], X[3]}
begin
  if X1[1]<X2[1] then Flag1 := (X[1]>=X1[1]) and (X[1]<=X2[1])
  else if X1[1]>X2[1] then Flag1 := (X[1]>=X2[1]) and (X[1]<=X1[1])
  else Flag1 := X[1]=X1[1];
  if X1[2]<X2[2] then Flag2 := (X[2]>=X1[2]) and (X[2]<=X2[2])
  else if X1[2]>X2[2] then Flag2 := (X[2]>=X2[2]) and (X[2]<=X1[2])
  else Flag2 := X[2]=X1[2];
  if X1[3]<X2[3] then Flag3 := (X[3]>=X1[3]) and (X[3]<=X2[3])
  else if X1[3]>X2[3] then Flag3 := (X[3]>=X2[3]) and (X[3]<=X1[3])
  else Flag3 := X[3]=X1[3];
  Result := Flag1 and Flag2 and Flag3;
end; {function SectPointCheck}*)
{-----}
procedure SectCalc; {основные вычисления для заданной плоскости сечения}
var I: Integer;
  LinePoint1,LinePoint2,SectPoint,LinkCos: TRealVect3;
  Al,O: Integer; {указатели на атомы Al и O в списках; внутренние переменные
блока определения начальной и конечной точек связи}

```

```

A,B,C,D,Det: Double; {внутренние переменные блока вычисления точки
пересечения прямого и плоскости}
Flag1,Flag2,Flag3: Boolean; {результаты проверки принадлежности точки
отрезку по координатам X[1],X[2],X[3]; внутренние переменные блока проверки
принадлежности точки пересечения прямой и плоскости самой связи}
ParallelFlag: Boolean;
ACos: Double; {cos угла между нормалью к плоскости и связью; острый угол
(ACos>0) - на поверхности Al, тупой (ACos<0) - на поверхности O}
begin {Сбрасываем счётчики числа связей}
(*CalcStart := Time; {!!! использовалось при определении времени работы
процедуры !!!}*)
NAlLong := 0;
NAlShort := 0;
NOxLong := 0;
NOxShort := 0;
CalcPlaneS; {процедура из модуля PlaneU}
CalcPlaneP; {процедура из модуля PlaneU}
{Перебор связей, проверка на разрыв определение характеристик
разорванных связей и приращение счётчиков. Критическая секция. Её длительность
определяет общее время счёта.}
for I := 0 to LinksDataList.Count-1 do {перебор связей}
begin
PLink := LinksDataList[I]; {берём очередную связь}
{Определяем начальную и конечную точки и направляющие cos линии связи.
В предыдущей версии блок был оформлен как процедура GetLinkPoints.}
Al := PLink^.AlNum; {номер атома Al для связи PLink}
PA1 := AlDataList[Al]; {указатель на запись данных этого атома}
LinePoint1[1] := PA1.X; {первая точка}
LinePoint1[2] := PA1.Y;
LinePoint1[3] := PA1.Z;
O := PLink^.ONum; {номер атома O для связи PLink}
POxygen := OxigenDataList[O]; {указатель на запись данных этого атома}
LinePoint2[1] := POxygen.X; {вторая точка}
LinePoint2[2] := POxygen.Y;
LinePoint2[3] := POxygen.Z;
LinkCos[1] := (LinePoint2[1]-LinePoint1[1])/PLink.Length;
LinkCos[2] := (LinePoint2[2]-LinePoint1[2])/PLink.Length;
LinkCos[3] := (LinePoint2[3]-LinePoint1[3])/PLink.Length;
{Вычисление координат точки пересечения прямой, заданной точками
LinePoint1,LinePoint2 и плоскости сечения; если прямая || плоскости,
ParallelFlag := True, иначе False, а результат полагается равным (0,0,0). В
предыдущей версии блок был оформлен как процедура SectPointCalc.}
A := (LinePoint2[1]-LinePoint1[1])/(LinePoint2[3]-LinePoint1[3]);
C := (LinePoint2[2]-LinePoint1[2])/(LinePoint2[3]-LinePoint1[3]);
B := LinePoint1[1]-A*LinePoint1[3];
D := LinePoint1[2]-C*LinePoint1[3];
Det := PlaneNCos[1]*A+PlaneNCos[2]*C+PlaneNCos[3];
if Det=0 then
begin
ParallelFlag := True;
SectPoint[3] := 0; SectPoint[2] := 0; SectPoint[1] := 0;
end {if Det=0 then ...}
else
begin
ParallelFlag := False;
SectPoint[3] := (PlaneP-PlaneNCos[1]*B-PlaneNCos[2]*D)/Det;
SectPoint[1] := A*SectPoint[3]+B;
SectPoint[2] := C*SectPoint[3]+D;
end; {if Det=0 then ... else ...}

```

{Проверяем, принадлежит ли точка пересечения отрезку связи. т.е. является ли связь разорванной. В предыдущей версии блок был оформлен как функция SectPointCheck.}

```

    if not ParallelFlag then {если связь не параллельна плоскости}
    if LinePoint1[1]<LinePoint2[1] then
        Flag1 :=
(SectPoint[1]>=LinePoint1[1])and(SectPoint[1]<=LinePoint2[1])
    else if LinePoint1[1]>LinePoint2[1] then
        Flag1 :=
(SectPoint[1]>=LinePoint2[1])and(SectPoint[1]<=LinePoint1[1])
    else Flag1 := SectPoint[1]=LinePoint1[1];
    if LinePoint1[2]<LinePoint2[2] then
        Flag2 :=
(SectPoint[2]>=LinePoint1[2])and(SectPoint[2]<=LinePoint2[2])
    else if LinePoint1[2]>LinePoint2[2] then
        Flag2 :=
(SectPoint[2]>=LinePoint2[2])and(SectPoint[2]<=LinePoint1[2])
    else Flag2 := SectPoint[2]=LinePoint1[2];
    if LinePoint1[3]<LinePoint2[3] then
        Flag3 :=
(SectPoint[3]>=LinePoint1[3])and(SectPoint[3]<=LinePoint2[3])
    else if LinePoint1[3]>LinePoint2[3] then
        Flag3 :=
(SectPoint[3]>=LinePoint2[3])and(SectPoint[3]<=LinePoint1[3])
    else Flag3 := SectPoint[3]=LinePoint1[3];
    if Flag1 and Flag2 and Flag3 then
        begin {если да, то фиксируем разорванную связь}
            ACos := LinkCos[1]*PlaneNCos[1]+LinkCos[2]*PlaneNCos[2]
                +LinkCos[3]*PlaneNCos[3];
            if PLink.LongLinc then {если разорвана длинная связь}
                if ACos>0 then {длинная связь от A1} NAllong := NAllong+1
                else {длинная связь от O} NOxLong := NOxLong+1
            else {если разорвана короткая связь}
                if ACos>0 then {короткая связь от A1} NAlShort := NAlShort+1
                else {короткая связь от O} NOxShort := NOxShort+1;
            end; {if Flag1 and Flag2 and Flag3 ...then... }
        end; {for I := 0 to LinksDataList.Count-1 do ...}
        {Завершающие вычисления}
        NLongSectLinks := NAllong+NOxLong; {всего разорвано длинных связей}
        NShortSectLinks := NAlShort+NOxShort; {всего разорвано коротких связей}
        NSectLinks := NLongSectLinks+NShortSectLinks; {всего разорвано связей}
        NAl := NAllong+NAlShort; {всего разорвано связей от A1}
        NOx := NOxLong+NOxShort; {всего разорвано связей от O}
        E0 := NAllong*eAllong+NAlShort*eAlShort+NOxLong*eOxLong+NOxShort*eOxShort;
            {поверхностная энергия, не нормированная на площадь}
            {Нормируем на площадь}
        NSectLinksDS := NSectLinks/PlaneS;
        NLongSectLinksDS := NLongSectLinks/PlaneS;
        NShortSectLinksDS := NShortSectLinks/PlaneS;
        NAlDS := NAl/PlaneS;
        NOxDS := NOx/PlaneS;
        E := E0/PlaneS;
        (*CalcStop := Time; {!!! использовалось при определении времени работы
процедуры !!!}*)
    end; {procedure SectCalc}

```

end.