Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ижевский государственный технический университет

имени М.Т.Калашникова»

(ФГБОУ ВПО «ИжГТУ им.М.Т. Калашникова»)

Факультет «Информатики и вычислительной техники»

Кафедра «Вычислительная техника»

Отчет по лабораторной работе

«Моделирование работе сканирующего туннельного микроскопа»

по дисциплине: «Информационно-измерительные системы»

Вариант №13

Выполнил: ст. гр. Б06-781-4т

Шаклеин И.В.

Проверил: д.т.н., профессор

Шелковников Е. Ю.

Ижевск 2016

Оглавление

[Задание 3](#_Toc448847021)

[Теоретическая часть 3](#_Toc448847022)

[Принцип работы туннельного микроскопа 4](#_Toc448847023)

[Метод статистических испытаний Монте-Карло для расчета интегралов 6](#_Toc448847024)

[Формирование туннельного тока между иглой и поверхностью 10](#_Toc448847025)

[Построение СТМ-профилограмм 14](#_Toc448847026)

[Листинг программы 17](#_Toc448847027)

[Результаты работы 21](#_Toc448847028)

[Заключение 22](#_Toc448847029)

[Список литературы 23](#_Toc448847030)

# Задание

Провести моделирование работы туннельного микроскопа и построить СТМ-профилограммы для заданной поверхности (рис. 1).

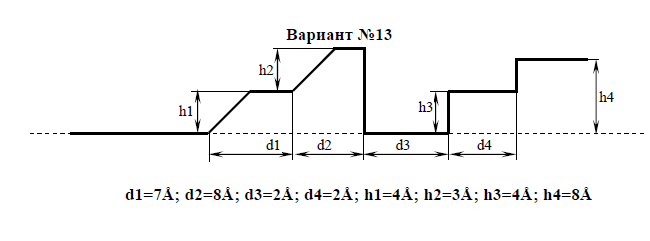


Рисунок 1. Поверхность для построения СТМ-профилогорамм

# Теоретическая часть

Физика поверхностных явлений в настоящее время является одним из наиболее интенсивно развивающихся разделов науки. Именно на фундаментальных исследованиях в области физики поверхности твердого тела основаны успехи современных микро и наноэлектроники, исследование разнообразных электронных, атомных и молекулярных процессов, происходящих на поверхности твердых тел. Остается актуальной задачей и заветное желание ученых на протяжении многих лет – непосредственное наблюдение за поведением отдельных атомов на поверхности твердого тела и изучение процессов с участием одиночных или небольших групп атомов.

Первостепенное значение для понимания свойств любого объекта имеет знание его атомной структуры, поэтому определение поверхностных структур - один из наиболее важных разделов физики поверхности. Последние 30 лет микроструктура поверхностей твердых тел интенсивно изучалась методами дифракции и рассеяния электронных и ионных пучков. Однако большинство этих методов первоначально разрабатывалось для исследования объемной структуры твердых тел, поэтому они не всегда годятся для получения информации о структуре поверхности, тем более на атомном уровне. Изобретение в 1982 году Г. Биннигом и Г. Рорером сканирующего туннельного микроскопа, который не накладывает ограничений на размеры образцов, реально открыло двери в новый микроскопический мир.

# Принцип работы туннельного микроскопа

В основе работы микроскопа (рис. 2) лежит явление квантовомеханического туннелирования электрона сквозь барьер, высота которого превышает энергию электрона. Если приблизить два проводника (электрода), разделённые диэлектриком (вакуум, газ или жидкость) на достаточно малое расстояние (∼ 4-20Å) и создать между ними смещающее напряжение *UT*, то в цепи потечёт туннельный ток *IT*, пропорциональный приложенному напряжению и экспоненциально зависящий от расстояния между электродами. [1]

Для реализации туннельного микроскопа проводящий образец используется в качестве основного электрода по отношению к металлическому острию, служащему сканирующим электродом. Сканирующий электрод установлен над основным на расстоянии нескольких ангстрем, представляющим собой туннельный барьер. Металлическое остриё служит в качестве источника электронов или, так называемого, холодного катода. Туннельный эффект применяется только для высвобождения электронов из металла контакта в диэлектрик. Под влиянием сильного электрического поля электроны высвобождаются из эмиттирующего острия и ускоряются для формирования изображения. Каждый из электродов может двигаться по трём осям X, Y, Z с помощью трёх пьезопреобразователей (ПП), обеспечивающих грубое перемещение образца и точное перемещение острия. Электронный блок управления подсоединен к электродам (т.е. к образцу и к острию) и к ПП, обеспечивающим малые перемещения электродов, которые также должны быть определёнными и воспроизводимыми. Точное положение острия в трёх координатах становится известным на основании значений напряжений, обеспечивающих соответствующий шаг ПП. Различные конструкции ПП позволяют перемещать остриё зонда на расстояния от долей ангстрема до нескольких десятков микрометров.

Туннельный ток измеряется и поддерживается постоянным с помощью точной настройки ПП по оси Z посредством системы ООС. Строчная развёртка осуществляется пьезоприводом по оси Х, кадровая – по оси Y. Поскольку положение пьезопривода пропорционально пьезонапряжению, напряжения трёх пьезоэлектрических двигателей определяют положение острия на каждой оси. В целом координаты являются декартовыми с тремя ортогональными осями. Анализ информации осуществляется в виде анализа трёхмерного изображения поверхности образца. [3]

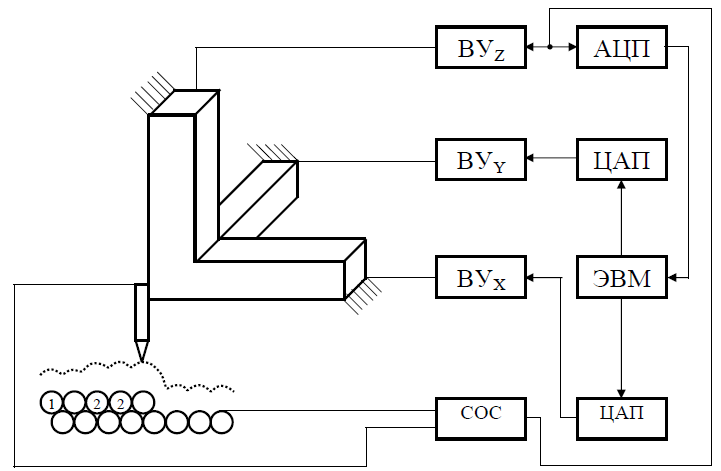


Рисунок 2. Функциональная схема сканирующего туннельного микроскопа: 1 – образец; 2 – примесные атомы; И – игла; ППX,Y,Z – пьезопреобразователь; СОС – система обратной связи; ЦАП и АЦП – цифро-аналоговые и аналого-цифровые преобразователи; ВУX,Y,Z – высоковольтные усилители.

СОС функционирует таким образом, чтобы при постоянном туннельном напряжении поддерживать действительный (измеренный) туннельный ток равным заданному. Исследованиями установлено, что при поддерживании системой СОС заданного значения туннельного тока с точностью не хуже 2% величина туннельного зазора остаётся также постоянной с ошибкой, не превышающей 0,1Å. Если производить сканирование остриём поверхности образца, подавая на пьезоприводы X,Y развёртывающие напряжения подобно телевизионной развёртке, то система СОС будет стремиться поддерживать заданным туннельный ток, а значит и постоянный зазор игла-остриё, изменяя напряжение на Z-пьезоприводе в соответствии с рельефом образца.

Таким образом, напряжение на Z-пьезоприводе острия будет линейно отражать текущую высоту рельефа поверхности образца. Для управления процессом сканирования, задания необходимых туннельных тока и напряжения, запоминания и хранения измерительной информации, а также её преобразования, обработки и отображения в требуемом виде в составе СТМ необходима ЭВМ с ЦАП, АЦП и устройствами отображения.

Первичное СТМ-изображение обычно формируется в виде карты распределения измеряемого параметра (высоты рельефа, работы выхода электрона и т.д.) по плоскости сканирования. Строго говоря, регистрируется не рельеф, а электронное состояние поверхности, но в большинстве случаев на однородных по составу образцах эти зависимости подобны. Для корректного использования таких изображений требуется тщательная калибровка пьезоприводов на объектах с известной топографией.

# Метод статистических испытаний Монте-Карло для расчета интегралов

В СТМ реализуется косвенный метод измерений. В результате косвенных измерений измеряются значения физических величин, функционально связанных с другими, являющимися конечной целью измерения. Для численных исследований параметров СТМ вместо физических экспериментальных данных можно воспользоваться экспериментальными данными, полученными в опытах над моделью объекта, т.е. с помощью вычислительного эксперимента. [6]

Задачи, связанные с вычислением многократных интегралов для исследования параметров СТМ, могут быть эффективно решены методом статистических испытаний, так как в этом случае к точности результатов расчёта не предъявляются очень жёсткие требования. Рассмотрим простейший пример для случая одного измерения.

Пусть ξ – непрерывная случайная величина, принимающая свои значения xiв некоторой области Ω на оси OX. Закон распределения задан плотностью вероятностей f(x) в Ω. Рассмотрим задачу об определении вероятности попадания случайной величины ξ в интервал ω с фиксированными границами a и b, содержащийся в Ω. Если обозначить искомую вероятность P(a ≤ ξ < b) = p, то она выражается в виде интеграла:

. (1)

Этот интеграл можно вычислить по методу статистических испытаний (эксперимент с получением случайных значений случайной величины ξ). Если появившееся при данном испытании значение xi находится внутри интервала ω, данное испытание будем считать удачным. После проведения N испытаний подсчитаем число m удачных испытаний и вычислим частоту p попадания случайной величины ξ в интервал ω:

. (2)

Располагая частотой p, мы можем приближённо оценить искомую вероятность p на основании закона больших чисел.

Для этого воспользуемся теоремой Бернулли: если событие А имеет вероятность p и если m – число наступления событий А при N независимых испытаниях, то каково бы ни было постоянное ε>0.

. (3)

При достаточно большом числе испытаний в качестве оценки для интеграла можно взять частоту , т.е.

. (4)

Моделирование эксперимента включает:

1. из совокупности случайных чисел с законом распределения f(x) извлекается число xi;
2. случайное число xi сравнивается с границами a и b интервала ω. Результаты сравнения отмечаются специальным признаком β, равным единице, если выполнено неравенство:

a ≤ xi<b (5)

и равным нулю в противном случае;

1. полученная величина β прибавляется к содержимому «счётчику числа удачных испытаний»;
2. к содержимому «счётчика количества испытаний» прибавляется единица.

После проведения N испытаний определяется приближённое значение искомой вероятности:

. (6)

Описанная процедура не требует запоминания всех случайных чисел, извлекаемых в процессе счёта. По ходу вычисления запоминаются только число испытаний N и число удачных испытаний m.

Для того, чтобы метод статистических испытаний можно было считать практически приемлемым, необходимо оценить точность равенства (1) и на этом основании определить число испытаний N для вычисления интеграла с достаточной точностью. Представление о точности можно получить, рассматривая p, как случайную величину. Она имеет математическое ожидание:

. (7)

и дисперсию:

. (8)

Поэтому средняя квадратичная ошибка  равенства (4) будет равна:

. (9)

Видно, что максимум  достигается при р = 0,5.

Обсудим вопрос о точности метода более подробно. Равенство (4) имеет точность ε с надежностью α, если для неравенства  справедливо соотношение:

. (10)

Свяжем величины ε и α с числом испытаний N. Первую ориентировку в этом вопросе можно получить из неравенства Чебышева, справедливого для любой случайной величины. В наших обозначениях оно имеет вид:

. (11)

Сопоставляя выражения (10) и (11), можно принять:

. (12)

Если подставить вместо  выражение (9), получим:

. (13)

Отсюда:

. (14)

Формула (14), полученная на основании неравенства Чебышева, даёт сильно завышенное значение N. Более точную оценку для N можно получить в том случае, если использовать закон распределения случайной величины из равенства (10). Величинаимеет приасимптотически нормальное распределение. На этом основании при больших N равенство (10) можно записать в виде:

, (15)

где ta– величина критического интервала, которая выбирается из таблиц нормального распределения по заданной надежности α. Сравнивая это соотношение с (27) мы видим, что или

. (16)

Отсюда можно определить значение N:

. (17)

Для того, чтобы наглядно представить порядок величины N, обеспечивающей заданную точность, положим, как и прежде, α=0,95 и вычислим N для различных значений p и ε. Результаты расчёта даны в таблице 1. При увеличении требований к точности расчёта (уменьшение ε) существенно увеличивается необходимое число испытаний N. Это обстоятельство является одним из серьёзных ограничений применимости метода статистических испытаний. Метод целесообразно применять для решения тех задач, для которых требования к точности не являются слишком жёсткими.

Таблица 1

Значения N при α=0,95 и различных p и ε

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | | ε | | | |
| 0.05 | 0.01 | 0.005 | 0.001 |
| 0.1 | 0.9 | 140 | 3600 | 14000 | 360000 |
| 0.2 | 0.8 | 250 | 6200 | 25000 | 620000 |
| 0.3 | 0.7 | 330 | 8400 | 33000 | 840000 |
| 0.4 | 0.6 | 380 | 9400 | 38000 | 940000 |
| 0.5 | | 390 | 9800 | 39000 | 980000 |

Рассмотрим простой вариант метода Монте-Карло для вычисления одно-мерных интегралов. Выберем N случайных чисел 0 ≤ Ui ≤ 1 и осуществим пере-ход к отрезку [a,b] по формуле ui = a+Ui(b-a).

Рассмотрим величину:

 (18)

которая представляет собой среднее арифметическое значений функции f, умноженное на b-a. Такой способ вычисления интеграла можно интерпретировать как статистический вариант метода прямоугольников, когда в качестве узла берется случайное число, равномерно распределенное на интервале интегрирования [a,b].

# Формирование туннельного тока между иглой и поверхностью

Пространственное разрешени СТМ определяется размерами области пространства, в котором в основном протекают токи между остриём и подложкой.

С целью упрощения задачи определения туннельного тока в СТМ рассмотрим её в приближении условно изолированных атомов, когда на поверхности острия (или подложки) расположены эмитирующие точки – независимые точечные источники тока с плотностью J, экспоненциально зависящей от расстояния между рассматриваемыми точками острия и подложки и отражающей величины как нормальной к эмитирующей поверхности, так и тангенциальной компонент начальных скоростей туннелирующих электронов. При этом размер источника тока на поверхности сравним с диаметром атома, а общий ток IT, текущий через туннельный промежуток, является суммой всех элементарных токов J. Следует отметить, что ВКБ-приближение, справедливое для поля плоского конденсатора, т.е. для перпендикулярной к подложке линии плотности тока, дополняется в данной модели другими линиями плотности тока с целью учёта ухудшения ПР за счёт тангенциальной составляющей начальных скоростей туннелирующих электронов.[2]

Проведём анализ туннельного тока для острия иглы в форме кругового конуса, т.к. такая форма достаточно точно аппроксимирует рабочие микровыступы острия при ориентации (111) монокристалла, определяющие атомное разрешение СТМ.

Рассмотрим формирование туннельного тока с поверхности подложки ко всей поверхности острия иглы. Выберем на плоскости т.А с координатами (х; 0) и т.В на поверхности конуса с радиусом r и углом β между радиусом r и фронтальной плоскостью (рис. 2.a), где α – полуугол раствора конуса острия; Z0 – расстояние между остриём и подложкой; Xp – радиус окружности, полученной пересечением конической поверхности (порождаемой движением отрезка образующей конуса острия между его вершиной и подложкой) с поверхностью подложки; Hк – высота т.В, b – расстояние между т.А и проекцией т.В; Хm, Rm – соответственно, предельные значения х и r. Для определения тока , протекающего с подложки на всю поверхность конуса, выберем в окрестности т.А элементарную площадку dx×dk = xdγdxна поверхности подложки и в окрестности т.В элементарную площадку dl1×dl2 = (r/sinα)dβdr на боковой поверхности конуса (рис. 2.б). Для удобства расчётов представим ток  в виде двух составляющих: тока , протекающего с поверхности подложки, ограниченной окружностью с радиусом Хp, и тока , протекающего с остальной части подложки. Плотность тока  () на поверхности подложки для первой (второй) составляющей численно равна току, протекающему через единичную площадку в окрестностях т.А с координатой x<Xp(x>Xp) ко всей поверхности острия:

; , (19)

где βp = arcsin(Xp/x)+π/2 – предельное значение угла βдля x>Xp. Из рис.3.б определим расстояние Zпк между точками А и В:

. (20)

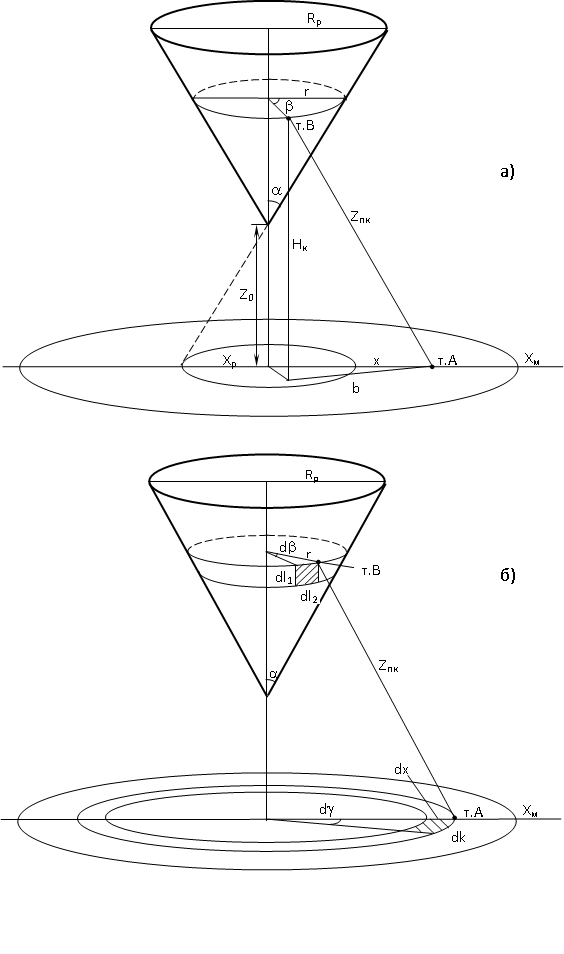


Рисунок 2. Формирование туннельного тока с плоской подложки на конусообразное острие

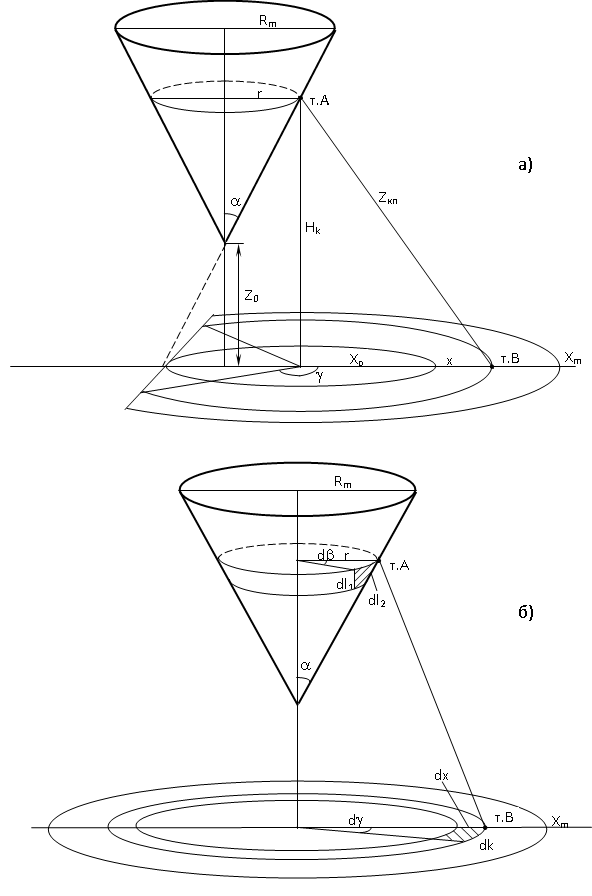


Рисунок 3. Формирование туннельного тока с поверхности конусообразного острия к плоской подложке

С учётом выражения (20) запишем выражение для общего тока :

. (21)

Рассмотрим формирование туннельного тока с боковой поверхности конуса ко всей поверхности подложки. Выберем на поверхности конуса т.А с радиусом r и на поверхности подложки т.В с координатами (x; 0) согласно рис. 3.а, где α – полуугол раствора конуса острия; Z0 – расстояние между остриём и подложкой; Xp – радиус окружности, полученной пересечением конической поверхности (порождаемой движением отрезка образующей конуса острия между его т.А и подложкой) с поверхностью подложки; Hк – высота т.А; Хm, Rm – соответственно, предельные значения х и r. Для определения тока , протекающего из т.А на всю поверхность подложки выберем в окрестности т.В элементарную площадку dx×dk = xdγdx (рис. 3.б). Плотность тока на поверхности острия численно равна току, протекающему через единичную площадку в окрестностях т.А ко всей поверхности подложки. Представим суммарный ток из т.А в виде двух составляющих, физической границей которых является коническая поверхность, порождаемая вращением отрезка образующей конуса острия между т. А и подложкой вокруг нормали из т.А к плоскости подложки. Плотности тока  для обеих составляющих запишутся следующим образом:

;. (22)

Из рис. 4 определим расстояние Zкп между точками А и В:

. (23)

Плотность тока Jкп между точками А и В определяется из (7.4). Для определения всего тока  с конуса на подложку выберем в окрестности т.А элементарную площадку на боковой поверхности конуса dl1×dl2 = (r/sinα)dβdr (рис. 3.а). С учетом (22) запишем выражение для общего тока :

. (24)

# Построение СТМ-профилограмм

Недостатком, ограничивающим потенциальные возможности СТМ, является то, что электроны туннелируют к поверхности расходящимся пучком, а это приводит к значительным искажениям СТМ-изображения и ухудшению пространственного разрешения СТМ. Для исследования возможностей улучшения ПР, как одной из важнейших метрологических характеристик нанопрофилометра, необходима информация о механизме формирования СТМ-изображения.[5]

На основании проведённых численных исследований электронного пучка острия иглы создана методика построения СТМ-изображения при движении острия иглы известных формы и размеров над исследуемой поверхностью, суть которой заключается в следующем. Выбираются геометрические параметры и материал иглы, геометрические параметры и материал образца, рабочие параметры сканирования. Программно моделируется процесс работы СТМ, при котором туннельные токи рассчитываются с учётом геометрии этих поверхности, а также с использованием обобщенной формулы Симмонса для потенциального барьера произвольной формы и метода статистических испытаний.

Алгоритм формирования СТМ-изображения с учетом полученных выражений для туннельного тока представлен на рис. 4.



Рисунок 4. Алгоритм сканирования поверхности с вычислением туннельного тока

Перед началом сканирования устанавливаются необходимые параметры: величина работы выхода электронов материалов иглы и образца, туннельный зазор Z0, угол конуса α. Затем вычисляется эталонный туннельный токIТэт, который виртуальная СОС СТМ поддерживает постоянным.. Вычисление IТэт проводится для заданного туннельного зазора Z0 и плоской поверхности. Компьютерное сканирование начинается с установки острия в начальную точку (x0,y0) растра исходной поверхности, СТМ-изображение которой необходимо получить. Далее находится плотность тока в каждой точке растра (xi,yi), определяется «видимость» острия в данной точке и вычисляется туннельный ток IТ(xi,yi) = JТ(xi,yi) ·S(xi,yi) ·sinγ, если точка «видима» (где S(xi,yi) = (xi+1 – xi-1)(yi+1 – yi-1)/4 – площадь поверхности в окрестностях данной точки (xi,yi);γ – угол между нормалью в точке(xi,yi) и отрезком прямой, являющимся минимальным расстоянием между точкой(xi,yi) и острием). Затем вычисляется суммарный ток со сканируемой поверхности на ЗО путем суммирования всех найденных IT(xi,yi) и сравнивается с эталонным током IТэт. Если токи IT(xi,yi) и IТэт не равны, то выполняется коррекция туннельного зазора следующим образом. Если IT(xi,yi)>IТэт, то туннельный зазор Z0 увеличивается (при IT(xi,yi)<IТэт, Z0 уменьшается) на определенную величину и повторно вычисляется IT(xi,yi). Далее с использованием линейной интерполяции уточняется значение туннельного зазора Z0 и снова вычисляется IT(xi,yi). Уточнение выполняется до тех пор, пока IT(xi,yi) не станет равен IТэт с заданной погрешностью δT. После этого в формируемый массив высот СТМ-изображения заносится сумма текущей высоты исходной поверхности и текущего значения туннельного зазора Z0. Затем центр острия перемещается в следующую точку сканирования (пока не будет отсканирована вся поверхность) и процедура «виртуальной» стабилизации туннельного тока повторяется. Таким образом, в результате компьютерного сканирования исходной поверхности формируется новый массив высот исследуемой поверхности.

Иглу нужно поддерживать на такой высоте, чтобы туннельный ток оставался постоянным. Для этого нужно находить такую новую высоту, чтобы выполнялось равенство:

. (25)

Это уравнение проще всего решить методом секущих или модифицированным методом Ньютона последовательными итерациями по формуле:

. (26)

Как видно, что перед тем, как приступить к последовательным итерациям, нам нужно знать два предыдущих приближения решения уравнения.

Пусть,  и  – первое и второе приближение соответственно. Принимая во внимание итерационную формулу метода секущих можно записать:

. (27)

Для метода секущих требуется знать два приближения z1 и z2. При этом очень важно, чтобы эти числа были близко расположены к решению уравнения (25), в противном случае алгоритм будет работать неустойчиво, возможно, бесконечное зацикливание, или будет найдена совсем другая высота z, которая тоже является решением уравнения (25). Чтобы избежать этих неприятностей, нужно выбирать малый шаг движения иглы в направлении оси OX, чтобы z1 близко находилось к решению уравнения (25). z2 следует выбирать, как

z2 = z1 + δ, если I1 > Iэт,.

z2 = z1 – δ, если I1 < Iэт. .

У этого метода имеются следующие недостатки:

1. малый шаг вдоль оси OX означает большое количество вычислений, а следовательно проигрыш в скорости вычислений;
2. требование малого шага – является необходимым, но недостаточным условием сходимости метода секущих т.к. по мере продвижения иглы вдоль оси OX всеже возможны большие скачки значений туннельного тока из-за неравномерности поверхности. Это может привести к затягиванию итерационного процесса, что неизбежно влечет потерю производительности.

# Листинг программы

#include <iostream>

#include <string>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <Windows.h>

#include <cstdlib>

const double Lferm = 5.71;

const double K = 1.0;

const double Fi = 4.5;

const int N = 10000;

double Random[N][2];

using namespace std;

class KoordPoint

{ public:

double x;

double y;

};

KoordPoint koordpointxy[12] =

{ {0, 0},{ 10, 0},{ 14, 4},{ 17, 4},

{ 20, 7},{ 25, 7},{ 25, 0},{ 27, 0},

{ 27, 4},{ 29, 4},{ 29, 8},{ 39, 8}};

double J(double x, double y, double z, double U)

{

double S1 = 3.0 / (K \* Fi);

double Zxy = sqrt(pow(x, 2) + pow(y, 2) + pow(z, 2));

double S2 = Zxy\*(1.0 - 23.0 / (3 \* Fi \* K \* Zxy + 10 - 2 \* U \* K \* Zxy)) + S1;

double FiOfZ = Fi - (U\*(S1 + S2) / (2 \* Zxy)) - (2.86 / (K\*(S2 - S1))\*log((S2\*(Zxy - S1)) / (S1\*(Zxy - S2))));

double res = 1620.0\*U\*Lferm\*exp(-1.025\*Zxy\*sqrt(FiOfZ));

return res;

}

double MC\_Int(double xL, double xH, double yL, double yH, double Z, double U)

{

double b[2] = { xH, yH }; //Верхние пределы интегрирования

double a[2] = { xL, yL }; //Нижние пределы интегрирования

double result = 0.0; //Результат интегрирования

double V = (b[0] - a[0])\*(b[1] - a[1]); //Объем

double X[2] = { 0.0, 0.0 }; //Буферный массив

bool Flag = true; //Флаг, определяющий попали ли мы в интервал

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

{

for (int j = 0; j < 2; j++) X[j] = a[j] + (b[j] - a[j])\*Random[i][j];

Flag = true;

for (int j = 0; j < 2; j++)

if ((X[j] < a[j]) && (X[j]>b[j])) Flag = false;

if (Flag) result += J(X[0], X[1], Z, U);

}

return result \* V / N;

}

/\* расчет нового значения туннельного тока \*/

double NewJ(double xp, double zp, double U)

{

const int MAX = 13;

double xH = 0.0, xL = 0.0, yH = 10.0, yL = -10.0;

double Z = 0.0;

double j = 0.0;

bool Vid[13]; //Массив видимости

for (int i = 0; i < 11; i++)

{

Vid[i] = false;

if ((abs(koordpointxy[i].x - xp) <= MAX) && ((koordpointxy[i].y - zp) <= MAX))

Vid[i] = true;

}

if (Vid[0] && (zp > 0))

{

xL = koordpointxy[0].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[1].x - xp;

if (xH > MAX) xH = MAX;

Z = zp;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[1] && (zp > (xp - koordpointxy[1].x)))

{

Z = (zp - (xp - koordpointxy[1].x))\*cos(M\_PI / 4);

xL = (koordpointxy[1].x - (xp + Z\*cos(M\_PI / 4))) / cos(M\_PI / 4);

if (xL > MAX) xL = MAX;

xH = (koordpointxy[2].x - (xp + Z\*cos(M\_PI / 4))) / cos(M\_PI / 4);

if (xH > MAX) xH = MAX;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[2] && (zp > koordpointxy[2].y))

{

xL = koordpointxy[2].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[3].x - xp;

if (xH > MAX) xH = MAX;

Z = zp - koordpointxy[2].y;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[3] && (zp > (xp - koordpointxy[3].x)))

{

Z = (zp - (xp - koordpointxy[3].x))\*cos(M\_PI / 4);

xL = (koordpointxy[3].x - (xp + Z\*cos(M\_PI / 4))) / cos(M\_PI / 4);

if (xL > MAX) xL = MAX;

xH = (koordpointxy[4].x - (xp + Z\*cos(M\_PI / 4))) / cos(M\_PI / 4);

if (xH > MAX) xH = MAX;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[4] && (zp > koordpointxy[4].y))

{

xL = koordpointxy[4].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[5].x - xp;

if (xH > MAX) xH = MAX;

Z = zp - koordpointxy[4].y;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[5] && (xp < koordpointxy[5].x))

{

xL = koordpointxy[5].y - zp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[6].y - zp;

if (xH > MAX) xH = MAX;

Z = koordpointxy[5].x - xp;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[6] && (zp > koordpointxy[6].y))

{

xL = koordpointxy[6].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[7].x - xp;

if (xH > MAX) xH = MAX;

Z = zp - koordpointxy[6].y;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[7] && (xp < koordpointxy[7].x))

{

xL = koordpointxy[7].y - zp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[8].y - zp;

if (xH > MAX) xH = MAX;

Z = koordpointxy[7].x - xp;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[8] && (zp > koordpointxy[8].y))

{

xL = koordpointxy[8].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[9].x - xp;

if (xH > MAX) xH = MAX;

Z = zp - koordpointxy[8].y;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[9] && (xp < koordpointxy[9].x))

{

xL = koordpointxy[9].y - zp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[10].y - zp;

if (xH > MAX) xH = MAX;

Z = koordpointxy[9].x - xp;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

if (Vid[10] && (zp > koordpointxy[10].y))

{

xL = koordpointxy[10].x - xp;

if (xL > MAX) xL = MAX;

xH = koordpointxy[11].x - xp;

if (xH > MAX) xH = MAX;

Z = zp - koordpointxy[11].y;

j += MC\_Int(xL, xH, yL, yH, Z, U);

}

return j;

}

void BuildProf(double Z0, double U0, HDC hDC)

{

const double e = 0.05;

const double step = 0.1;

double j0 = 0.0;

double z[3] = { 0.0, 0.0, 0.0 };

double j[3] = { 0.0, 0.0, 0.0 };

KoordPoint Diagramm[350];

POINT dot;

Diagramm[0].x = 0.0;

Diagramm[0].y = Z0;

MoveToEx(hDC, 5, 280 - Diagramm[0].y \* 11, &dot);

for (int i = 0; i < 10; i++) j0 += NewJ(0, Z0, U0) / 10.0;

for (int i = 1; i < 350; i++)

{

Diagramm[i].x = Diagramm[i - 1].x + step;

z[2] = Diagramm[i - 1].y;

j[1] = NewJ(Diagramm[i].x, z[2], U0);

if ((j[1] < (j0 \* (1 - e))) || (j[1] > (j0 \* (1 + e))))

{

z[1] = z[2] + step \* ((j[1] - j0) / abs(j[1] - j0));

j[0] = NewJ(Diagramm[i].x, z[1], U0);

if ((j[0] < (j0 \* (1 - e))) || (j[0] > (j0 \* (1 + e))))

{

int k = 0;

do

{

z[0] = z[1] - (j[0] - j0) \* (z[1] - z[2]) / (j[0] - j[1]);

j[1] = j[0];

j[0] = NewJ(Diagramm[i].x, z[0], U0);

z[2] = z[1];

z[1] = z[0];

k++;

if (k > 1000) break;

} while (!(j[0] >= j0 \* (1 - e) && (j[0] <= j0 \* (1 + e))));

Diagramm[i].y = z[0];

if (k > 1000) Diagramm[i].y = Diagramm[i - 1].y;

}

else Diagramm[i].y = z[1];

}

else Diagramm[i].y = z[2];

LineTo(hDC, Diagramm[i].x \* 11 + 5, 280 - Diagramm[i].y \* 11);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int k = 0;

string s = "";

cout << "Введите:" << endl;

cout << "1 - Для вывода профилограмм (U = 0.01 В)" << endl;

cout << "2 - Для вывода профилограмм (U = 0.1 В)" << endl;

cout << "3 - Для выхода" << endl;

cin >> k;

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

for (int j = 0; j < 2; j++) Random[i][j] = (double)rand() / RAND\_MAX;

/\* Инициализация рисования \*/

HWND hWnd = GetConsoleWindow();

HDC hDC = GetDC(hWnd);

POINT dot;

RECT consoleRect;

SelectObject(hDC, GetStockObject(WHITE\_PEN));

GetClientRect(hWnd, &consoleRect);

double Z0[4] = { 5, 7, 10, 15 };

//for (int j = 0; j < 2; j++)

system("cls");

FillRect(hDC, &consoleRect, (HBRUSH)GetStockObject(BLACK\_BRUSH));

MoveToEx(hDC, 5, 280, &dot);

for (int i = 1; i < 12; i++) LineTo(hDC, koordpointxy[i].x \* 11 + 5, 280 - koordpointxy[i].y \* 11);

switch (k)

{

case 1:

cout << "U0 = 0.01 В" << endl;

for (int i = 0; i < 4; i++)

{

BuildProf(Z0[i], 0.01, hDC);

}

system("pause");

return 0;

case 2:

cout << "U0 = 0.1 В" << endl;

for (int i = 0; i < 4; i++)

{

BuildProf(Z0[i], 0.1, hDC);

}

system("pause");

return 0;

case 3:

system("pause");

return 0;

}}}

# Результаты работы

С помощью разработанной программы построим две профилограммы поверхности для высот 5 Å, 7 Å, 10 Å и 15 Å при различных туннельных напряжениях (рис. 5 и 6).

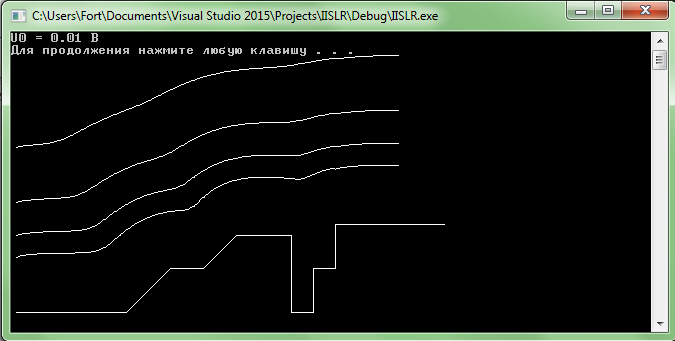


Рисунок 5. Профилограмма при U = 0.01 В

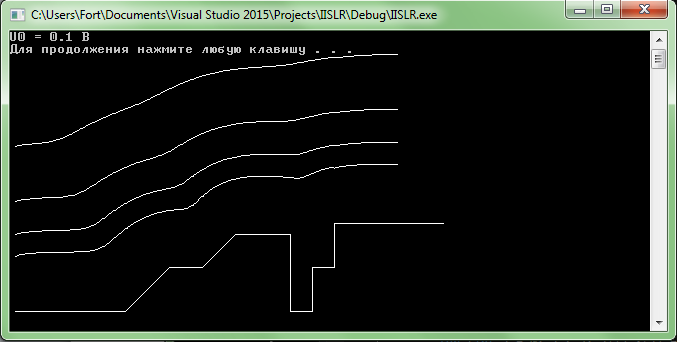


Рисунок 6. Профилограмма при U = 0.1 В

# Заключение

В данной работе были изучены принципы построения и функционирования сканирующего туннельного микроскопа (с помощью разработки программы на ЯВУ). Помимо этого в работе были отображены результаты выполнения программы.

# Список литературы

1. Эдельман В.С. Сканирующая туннельная микроскопия // ПТЭ, 1989.– №5.– С.25-49.
2. Бусленко Н.П., Шрейдер Ю.А. Метод статистических испытаний.– М.: ГИФМЛ, 1961.– 226с.
3. Шелковников Е.Ю. Исследование параметров туннельного микроскопа с применением метода Монте-Карло // Информационно-измерительные системы на базе наукоёмких технологий. - Ижевск, 1998. - С.95-102.