Министерство образования и науки Российской Федерации
Федеральное агентство по образованию

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

А.П. Смирнов

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ИЗМЕРИТЕЛЬНЫХ ПРОЦЕССОВ

Практикум в среде MathCAD
на примерах из механики и оптики

Учебное пособие

Санкт-Петербург
2006

УДК 681.7:53.081.5
Смирнов А.П. Компьютерное моделирование измерительных процессов. Практикум в среде MathCAD на примерах из механики и оптики. Учебное пособие. - СПб: СПб ГУ ИТМО, 2006. 101с.

Изложены принципы построения программ в среде MathCAD. Приведены модели механического устройства и оптической системы в параксиальном приближении, на основе которых рассмотрены некоторые свойства измерительных процессов применительно к конструированию оптико-механических систем.

Пособие предназначено для студентов, обучающихся по направлению «Оптотехника» и специальности «Оптико-электронные приборы и системы».

“Рекомендовано УМО по образованию в области приборостроения и оптотехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200200 – Оптотехника и специальности 200203 – Оптико-электронные приборы и систем

Оглавление

Введение

Измерительный процесс (ИП) является предметом изучения метрологии. Кроме науки об измерениях метрология исследует методы и средства обеспечения единства измерений и требуемой точности.

ИП как предмет метрологии есть способ извлечения количественной информации о свойствах объектов и процессов с заданной точностью и достоверностью.

Структура ИП включает такие аспекты как: измерительная задача, объект измерения, принцип, метод и средство измерения, условия измерения, субъект измерения, результат и погрешность измерения.

ИП протекает в двух тесно связанных между собой плоскостях – реальной и модельной. Для каждого реального аспекта ИП строится его отражение, или модель. Как реальная, так и модельная плоскость ИП имеет свои собственные погрешности. Для реального ИП – это погрешности изготовления и функционирования элементов прибора, внешние влияющие факторы и погрешности субъекта измерения (оператора). Для модельной плоскости – это погрешности вычислений, погрешности же реальной модели моделируются методом Монте-Карло.

Кроме того, построение модели ИП сопряжено с методической погрешностью. Действительно, реальный объект измерения по отношению к другим объектам характеризуется совокупностью свойств, часть из которых требуется изучить путём измерения. Построение модели объекта происходит с помощью формирования физической величины, описывающей изучаемое свойство. При этом неизбежно происходит потеря точности соответствия физической величины как абстракции и описываемого с помощью этой величины реального свойства.

Принцип измерения базируется на совокупности физических законов, имеющих свои ограничения при описании реальности, и поэтому содержит методическую погрешность.

Метод измерения как совокупность приёмов сравнения измеряемой физической величины с её единицей измерения вносит погрешность дискретизации.

Построение модели ИП предшествует реальному измерению и при этом реальный ИП использует свою модель при оценке результатов измерений.

Для конструктора точных (оптических) приборов, очевидно, что построение модели ИП эквивалентно этапу конструирования измерительного прибора, а реальный ИП связан с его эксплуатацией.

Проектирование, конструирование и эксплуатация технических устройств и оптических приборов в частности предполагает возможность получения информации с высоким уровнем доверительной вероятности. Это возможно, только при условии тщательного изучения и выявления свойств ИП.

Наиболее эффективным способом изучения свойств проектируемого прибора является его математическое, или компъютерное, моделирование ИП, оперирующее не только с моделью ИП, но и стремящееся описать, промоделировать, и всевозможные внешние влияющие факторы: внешние условия эксплуатации и первичные погрешности элементов прибора.

Критерием оценки служит качество изображения оптических приборов. Оно определяется остаточными оптическими аберрациями и первичными погрешностями функционирования прибора. Изучение свойств первичных погрешностей позволяет оптимизировать процессы проектирования и конструирования оптических приборов.

Оптический прибор включает в себя оптические элементы: линзы, зеркала, плоскопараллельные пластины, оптические клинья, спектральные и специальные призмы. Первичные погрешности оптических элементов в составе оптического прибора включают в себя погрешности изготовления и погрешности положения элемента в пространстве, а также погрешности, связанные с влиянием внешней среды, изменяющей физические свойства материалов.

С классификацией погрешностей по типам (систематические, случайные или методические и инструментальные), а также по квалитетам точности и уровням производства можно познакомиться в литературе [1,2].

Погрешности могут быть скалярными: погрешности показателя преломления, толщины линз и воздушных промежутков, температуры, массы и т.д., и векторными: погрешность положения элемента при децентрировки оптической поверхности в процессе изготовления и при его фиксации в зазоре оправы, в результате торцевого биения прижимной детали и т. п.

Основной задачей при разработке конструкции прибора является распределение конструктивного допуска на изготовление между первичными погрешностями. С аналитическими методами расчёта оптического прибора на точность можно познакомиться в литературе [2-3]. В данном пособии представлена методика моделирования первичных погрешностей заданного типа и свойства, исследования их влияния на тот или иной отклик прибора и по полученной реакции определения первичных погрешностей функционирования оптического прибора с использованием метода Монте-Карло в рамках программирования в среде MathCAD.

Пособие предваряет краткое описание средств программирования в среде MathCAD. Для закрепления материала предложено ряд заданий по составлению программ.

Ключевые программы модели снабжены подробными комментариями. Предполагается, что освоение всем комплексом программ даст в руки конструктора полезный инструмент для расчёта оптического прибора на точность, габаритного расчёта, энергетического расчёта, включающего учёт виньетирования.

1. Алгоритмы и языки программирования

Алгоритм – это описание процесса решения задачи.

В повседневной жизни всевозможные *правила*, *наставления*, *уставы*, *законы* суть алгоритмы. Они отличаются друг от друга формой описания и степенью детализации.

Уточним определение алгоритма.

Алгоритм – конечный **набор правил**, расположенных в определённом логическом порядке, позволяющий исполнителю пошагово решать любую конкретную задачу из некоторого класса *однотипных* задач.

Составление алгоритма называют **алгоритмизацией.**

Слово "алгоритм" возникло из названия латинского перевода книги арабского математика IX века Аль-Хорезми "Арифметическое искусство индусов" – "Algoritmi de numero Indorum".

Алгоритм должен отвечать определённым требованиям. Выделяют следующие 7 условий:

1) *наличие ввода* исходных данных,
2) *наличие вывода* результата выполнения,
3) *однозначность* – компьютер понимает только однозначные инструкции,
4) *общность* – алгоритм предназначен для решения не одной задачи, а целого класса задач,
5) *корректность* – алгоритм должен давать правильное решение задачи,
6) *конечность* – решение задачи должно быть получено за конечное число шагов,
7) *эффективность* – для решения задачи должны использоваться ограниченные ресурсы компьютера (процессорное время, объём оперативной памяти и т.д.).

Алгоритм имеет иерархическую структуру.

Степень *детализации* (подробности описания) зависит от уровня языка программирования. Чем *больше* деталей устройства реального компьютера содержится в описании воображаемой машины - исполнителя, для которой составляется алгоритм, тем *ниже* уровень абстракции.

Самый высокий уровень абстракции содержится в представлении о компьютере как фон-неймановской машине, состоящей из процессора, памяти и устройств ввода-вывода.

Самый низкий уровень абстракции используется в языках программирования низкого уровня. Языки низкого уровня, называемые *машинными кодами и ассемблерами*, используются для очень подробного и детального описания операций, при котором учитывается архитектура компьютера, устройство его центрального процессора.

Программа на языке ассемблера получается длинной, *трудоёмкость* программирования возрастает, растёт *вероятность ошибок*. Она привязана к *конкретной архитектуре*. Это можно считать недостатками языков низкого уровня. Преимуществом является возможность получить максимум *быстродействия*.

Первые языки программирования были языками низкого уровня.

Языки программирования *высокого уровня* не заботятся о деталях их реализации на конкретной архитектуре. Программы оказываются более *короткими, надёжными и универсальными*, а процесс их составления сокращается. Программы на языках высокого уровня легче читать, в них проще разбираться.

Языки высокого уровня подразделяются на 4 вида:

1) императивные (процедурные);
2) функциональные;
3) логические;
4) объектно-ориентированные.

Основными объектами в *императивных* языках являются переменные, операторы присваивания, стандартные алгоритмические конструкции. Императивные языки программирования привязаны к традиционной фон-неймановской архитектуре. Примерами императивных языков являются Fortran, Algol, ПЛ-1, Basic, Pascal и др.

Первым языком высокого уровня был FORTRAN, родителями которого является корпорация IBM и группа разработчиков во главе с Джоном Бэкусом.

В *функциональных языках* программирования используются функции, значения которых определяются по заданным параметрам. Традиционные переменные и операторы присваивания при этом уже не нужны, или, по крайней мере, не обязательны. Примером функционального языка является LISP.

В программах, написанных на *логических языках*, нет определённого, фиксированного порядка выполнения правил и шагов алгоритма. Выбор подходящей последовательности возлагается на систему. Примером такого языка является Progol.

Объектно-ориентированные языки упрощают программирование с использованием технологии объектно-ориентированного программирования (Java, C^{++}). Объектами служат классы изображений, классы окон, кнопки.

2. Программирование в среде MathCAD

Программирование в среде MathCAD не имеет своего названия, но оно построено на базе объектно-ориентированного языка C^{++} с помощью дальнейшего абстрагирования.

К обычным арифметическим операторам: сложение, умножение *добавляются операторы* интегрирования, дифференцирования, предельного перехода, суммирования и произведения рядов.

MathCAD – *визуально-ориентированная* система программирования. Это означает, что в подавляющем большинстве расчётных задач входной язык общения с MathCAD позволяет задавать их решение в виде вводимых с помощью операторов и функций математических формул в их естественном виде и указывать тип желаемых результатов (таблицы или графики). Язык диалога с MathCAD отличается от обычного языка высокого уровня, языка реализации системы C^{++}, сверхвысоким уровнем абстрагирования.

Входной язык MathCAD относится к *интерпретирующему* типу. Это означает, что когда система опознаёт какой-либо объект, она немедленно исполняет указанные в блоке операции, если блок не имеет особого признака пассивности.

Объектами системы могут быть *формульные, текстовые и графические* блоки. При этом формульные блоки могут иметь особые признаки – атрибуты: *активности, пассивности и оптимизации*.

Блоки анализируются в строго определённом порядке – *слева направо и сверху вниз.* Это означает, что блоки нельзя располагать в документе произвольно. Это правило нарушается при использовании *глобального* определения.

Основные *обычные средства* программирования включены в язык MathCAD.

Специальные приёмы предусмотрены для циклического изменения переменных и создания так называемых *ранжированных* переменных, имеющих набор значений.

По мере того как пользователь создаёт (средствами текстового, формульного и графического редакторов) в окне редактирования объекты (тексты, формулы, таблицы и графики), система сама составляет программу на некотором промежуточном *языке связи*, которая хранится в оперативной памяти до тех пор, пока не будет сохранена на диске с расширением *.mcd*.

2.1. Константы

Константы – объекты языка программирования, имеющие уникальные имена, хранящие некоторые значения числового или строкового характера.

Числовые константы могут быть десятичные, двоичные, восьмеричные, шестнадцатеричные, целые, дробные, вещественные и комплексные. Диапазон изменения констант в десятичном представлении от 10^{-307} до 10^{307}.

Признаком двоичных констант является буква *b* в конце числа, для восьмеричных – *o*, и для шестнадцатеричных – *h*.

Для записи комплексной константы требуется предварительно организовать с помощью операторов присваивания и извлечения корня мнимую единицу $i := \sqrt{-1}$.

Строковые константы – любые цепочки символов, исключая кириллицу, заключённые в кавычки: “<*string*>”.

Особый тип констант – единицы физических величин и некоторые физические константы, например,

$acre = 4046.856\ m^2$, $atm = 101325\ Pa$, $g = 9.807 \cdot \frac{m}{s^2}$.

2.2. Переменные

Переменные – имеют уникальные имена (идентификаторы), и принимают значение, тип которого определяется типом арифметического выражения в операторе присваивания. Предварительного описания переменные не требуют.

Оператор присваивания имеет два представления:

$$a := A \qquad a \leftarrow A\ ,$$

здесь a – переменная, A – арифметическое выражение.

Представление ":=" используется при составлении простых формульных блоков, а представление "←" для образования составных и вложенных блоков.

Идентификаторы имеют практически *любую длину*, должны начинаться с буквы, латинской или греческой (кириллица исключается), могут включаться цифры и некоторые спецсимволы (знак подчёркивания).

Выделяют следующие типы арифметического выражения: целый, вещественный, комплексный, скалярный, векторный, матричный, структурный, строковый, логический. Например:

$$a := 5,\ \ a := 5.,\ \ a := 5+2i,\ \ a := \begin{pmatrix} 2 \\ 5 \end{pmatrix},\ \ a := \begin{pmatrix} 2 & 1 \\ 5 & 2 \end{pmatrix},\ \ a := \begin{pmatrix} "d" \\ 5 \end{pmatrix},\ \ a := 2 \neq 1.$$

Представление переменой может быть выбрано с помощью пункта меню "Формат" и подменю "Формат результата" или путём обращения к контекстному меню при двукратном щелчке левой кнопкой на результате.

При этом может быть задана форма представления, в том числе и в виде неправильной дроби (в поздних версиях), и число знаков в представлении числа.

2.3. Программирование с помощью простых блоков

Как известно язык MathCAD– интерпретирующий. Поэтому он обрабатывает вводимые блоки сразу после их создания, если нет блокировки вычислений. Блоки обрабатываются слева направо и сверху вниз.

Последовательное применение при вычислениях простых формульных блоков с использованием встроенных функций среды и специальных директив условно можно назвать программированием с помощью линейных программ. Оно дополняется функциональным программированием с помощью функций пользователя.

Признаками программирования являются операции ветвления и цикла. Во-первых, циклическая операция неявно присутствует при образовании вектора случайной величины с заданным законом распределения. Это, например, встроенные функции:

rnorm(N,M,σ) и runif(N,a,b).

При обращении к ним генерируется N-вектор случайных величин, распределённых нормально или равномерно.

Во-вторых, операция цикла вводится с помощью ранжированной переменной, которая задаётся тремя своими значениями:

$$a:=b,c..d.,$$

начальным, вторым и границей последнего числа. Числа следуют с шагом $c - b$, если второе значение не указано, то шаг единичный. Последнее значение ранжированной переменной $a < d$. Целочисленная ранжированная переменная используется для образования индексированной переменной, например,

$$i := 0..6 \qquad a_i := i^2 \qquad a^T = (0\ \ 1\ \ 4\ \ 9\ \ 16\ \ 25\ \ 36)$$

Ранжированная переменная не вектор – никакой элемент ранжированной переменной не может быть выделен.

С помощью ранжированной переменной ведутся параллельные вычисления.

Ветвление при использовании простых блоков используется в следующих случаях.

1) Функция *if*

a:=if(логическая переменная или выражение,выражение1, выражение2).

Если значение логической переменной = 1, то возвращается значение выражения 1, иначе – выражения 2.

Пример. Задание прямоугольного импульса:

$$f(x) := \text{if}\left(|x| < \frac{1}{2}, 1, 0\right)$$

2) Функция обработки ошибочных действий, например, деления на ноль

a:= выражение 1 on error выражение 2.

В случае возникновения ошибки при вычислении выражения 1, вычисляется выражение 2.

Пример:

$$f(x) := \frac{\sin(x)}{x} \text{ on error } 1 \qquad f(0) = 1$$

2.4. Программирование с помощью составных блоков

Собственно программирование начинается с возможности образования составных блоков, возможностей полноценного ветвления алгоритма и образования циклов. Набор средств программирования сосредоточен в палитре программных элементов математической панели. Вызов палитры программирования осуществляется кнопкой вызова палитры программирования:

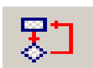


Рис.1. Кнопка на панели “Math”(Математика)

для вызова палитры программирования

Палитра программирования, вынесенная на поле документа, имеет вид:

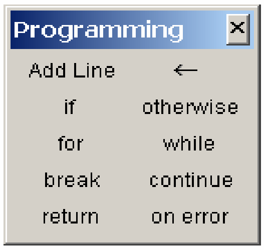


Рис.2. Палитра программирования

Элементом построения программного составного блока является кнопка для формирования блоков: “Add Line” (*Добавить линию*), поставляющая шаблон с колонкой свободных мест для введения простых блоков. С этой кнопки палитры начинается построение программы.

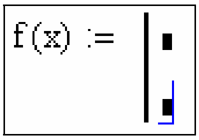


Рис.3. Фрагмент начала программы с именем f
и формальным параметром x.

Каждый из прямоугольников является местом заполнения, стартовой точкой, для введения программных операторов: простых блоков присваивания, циклов, операторов ветвления и др., а также для дальнейшего дробления на составные блоки с помощью кнопки “Add Line”.

Внутреннее присваивание в составном блоке осуществляется с помощью оператора “←”. Горячая клавиша – открывающая фигурная скобка - “{“.

Кнопки-инструкции *“if”* и *“otherwise”* используют для полноценного ветвления алгоритма, причём *“if”* может применяться самостоятельно, то есть без предписанного инструкцией *“otherwise”* ветвления. В этом случае невыполнение условий инструкции *“if”* равносильно пустому оператору “continue”.

Две кнопки-инструкции *“for”* и *“while”* применяют для формирования циклов. *“for”* – с фиксированным числом шагов, *“while”* - с выходом из цикла по условию.

Две кнопки-инструкции *“break”* и *“return”* организуют прерывание вычислений. *“break”* – внутри блока с переходом в блок на ступень выше, *“return”* – с выходом из программы.

В палитру программных элементов включена уже описанная ранее функция обработки ошибок *“on error*”.

При построении программ может быть полезна функция сообщения об ошибке *error(“текст”)*. Она набирается с клавиатуры. При обращении к этой функции происходит покраснение выражения, как при возникновении ошибки, и при наведении указателя мыши на покрасневшее выражение выдаётся текст, помещённый в аргумент этой функции.

Левым операндом оператора присваивания составного блока, или программы, может быть простая переменная или идентификатор функции – оригинальное имя программы - с перечнем в круглых скобках произвольного количества формальных локальных аргументов (рис.3).

Формальные они потому, что им не присвоено никакого значения. Только при обращении к функции, когда она появляется в составе выражения в правом операнде оператора присваивания, формальные аргументы должны быть определены обычным путём или непосредственно в списке *фактических* аргументов. Число и порядок фактических и формальных параметров не должны различаться.

Локальными будут также и все переменные в теле программы, образованные с помощью внутреннего присвоения. Если имеется внешняя переменная с одинаковым именем, то приоритет имеет локальное присвоение, хотя этой ситуации следует избегать во избежание путаницы. Внешне определённые переменные и функции, если они не переопределены локально, действуют и внутри программного блока.

Рассмотрим фрагмент программы.

$$a := 2 \qquad a = 2$$

$$b_1 := 1 \qquad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad a := 8$$

$$c := a \cdot b \qquad c = \begin{pmatrix} 0 \\ 8 \end{pmatrix} \qquad d := \left| \begin{array}{l} a \leftarrow 4 \\ \dfrac{c}{a} \end{array} \right.$$

$$d = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \qquad a = 8$$

Рис.4. Фрагмент документа MathCAD,
иллюстрирующий оператор присвоения

Переменной *a* присваивается значение 2. Присвоение действительно вплоть до следующего присвоения переменной значения 8. Это присвоение не влияет на локальное присвоение переменной с таким же именем другого значения, но если бы не было локального присвоения переменная *a* имела бы внутри блока значение последнего внешнего присваивания – 8.

Как видим, переменные никак специально не определяются. Определением их типа служит непосредственно их задание с помощью оператора присваивания. Определённый таким образом тип существует до тех пор, пока данная переменная не будет переопределена.

2.5. Блок-схема

При разработке структуры программы удобнее пользоваться записью алгоритма в виде блок-схемы. Для изображения основных алгоритмических структур и блоков на блок-схемах используют специальные графические символы.

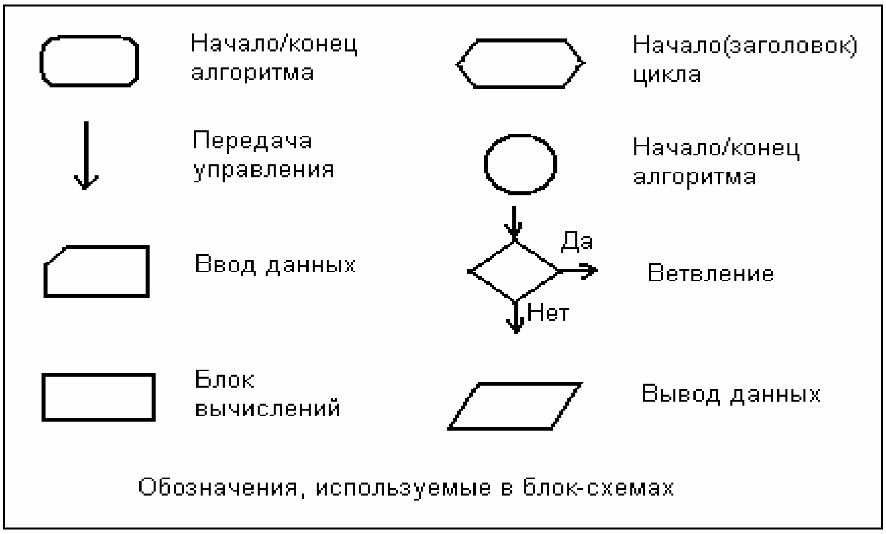


*Рис.5. Графические символы, используемые
при составлении блок-схем*

2.6. Составление и отладка программы в MathCAD

*Пример 1***.** Составим алгоритм вычисления квадратного корня из произвольного положительного вещественного числа *x* методом Герона. Метод основан на итерационной формуле:

$$y_{n+1} = \frac{1}{2}\; y_n + \frac{x}{y_n}\;, \quad \text{где} \quad y_0 = 1.$$

Окончанием вычислением будет ситуация, когда модуль изменения $\Delta y < eps$ – где *eps* – точность вычислений.

Блок-схема программы, например, может быть составлена одним из двух способов:

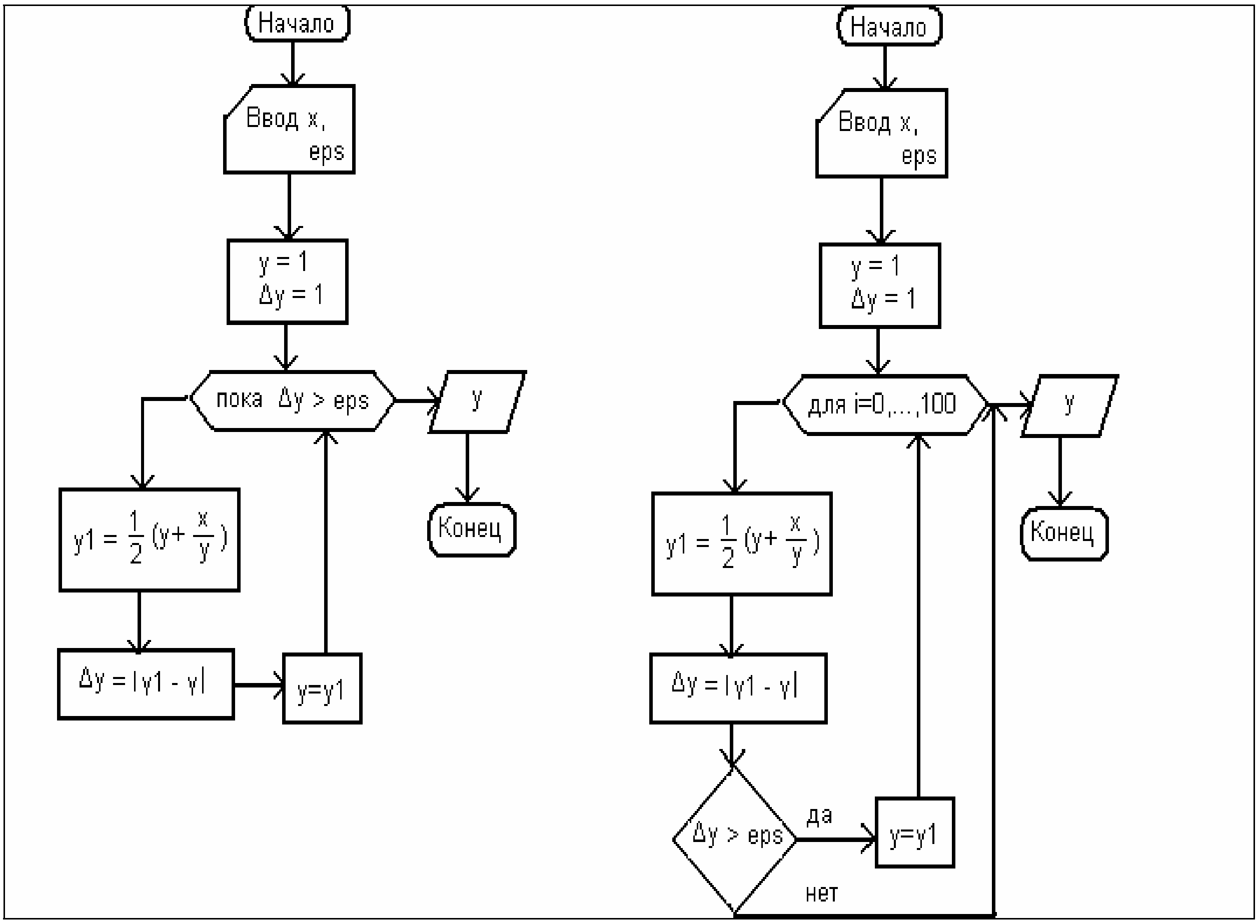


Рис.6. Блок-схемы для реализации формулы вычисления квадратного корня по методу Герона

В первом варианте используется цикл “пока”, во втором цикл с фиксированным числом шагов и выходом из цикла по дополнительному условию с ветвлением алгоритма.

Текст программы с применением цикла *while*:

Рис.7. Текст программы с применением цикла while:

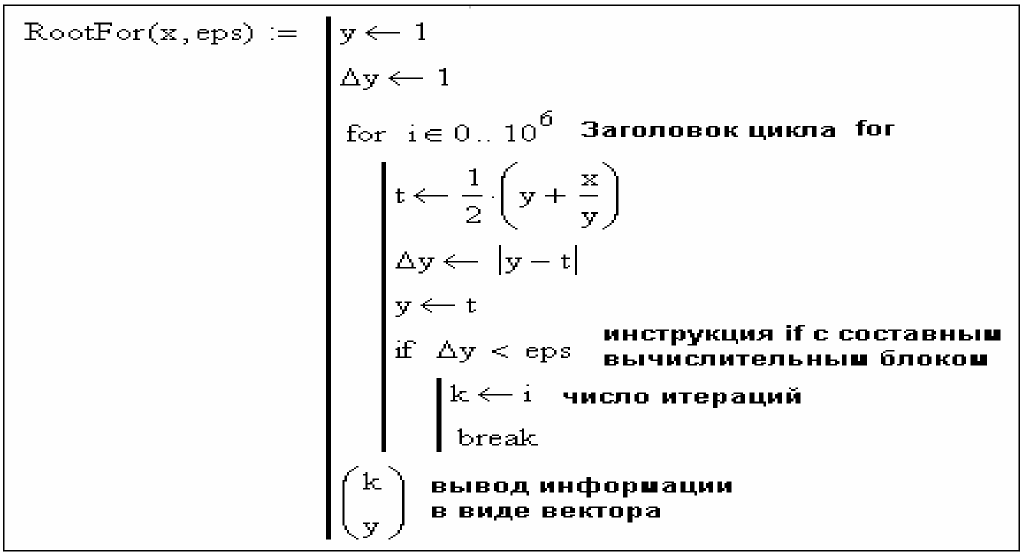


Рис.8. Текст программы с применением цикла for:

Блок-схема полезна при разработке программ независимо от того, на каком языке программирования она будет в дальнейшем составлена. Однако блок-схема не универсальна - характерные особенности языка программирования накладывают на неё свой отпечаток: блок-схема должна учитывать уровень детализации языка программирования и перечень обязательных и доступных операторов, а также ограничения используемого языка программирования.

Пример 2. Даны два отрезка: симметричный отрезок [-*r*, *r*] и произвольный отрезок [*a*, *b*] или [*b*, *a*], то есть порядок отношения величин *a* и *b* неизвестен. Требуется определить их общую часть, отнесённую к длине произвольного отрезка.

Все возможные случаи взаимного расположения отрезков приведены на рис.9.

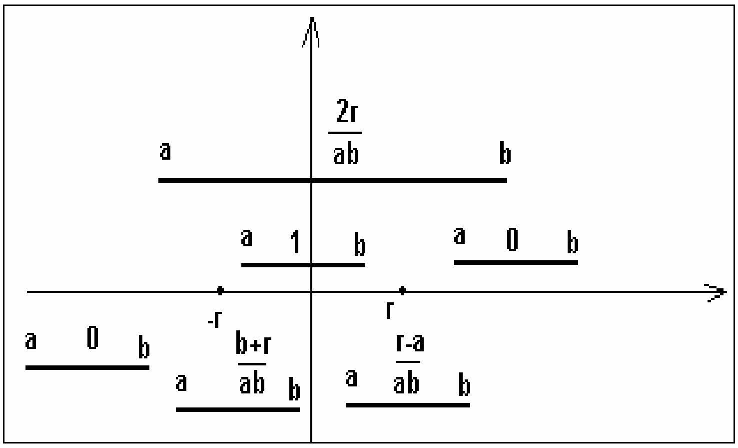


Рис.9. Относительный коэффициент перекрытия двух отрезков

Блок-схема варианта программы приведена ниже.

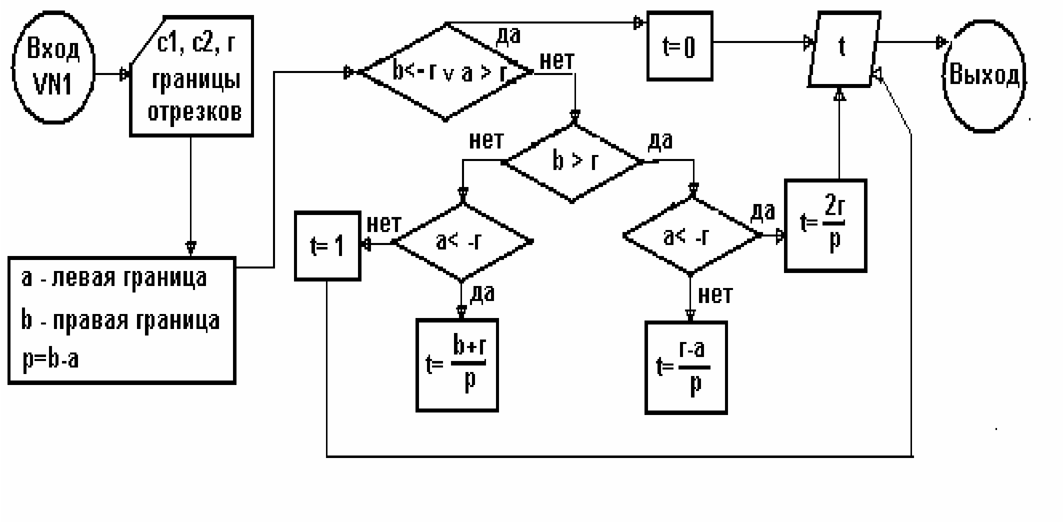


Рис.10. Блок-схема программы из примера 2, вариант1

Эта же программа может быть скомпонована по-другому за счёт объединения условий.

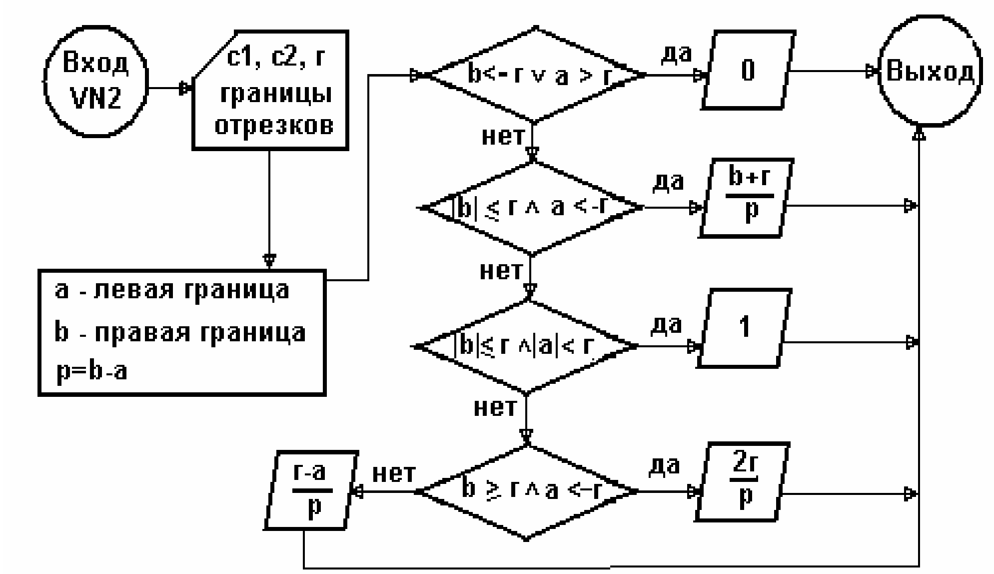


Рис.11. Блок-схема программы из примера 2, вариант 2

Тексты программ приведены ниже.

```
Vn1(c1,c2,r) := | a ← min(c1,c2)
                | b ← max(c1,c2)
                | p ← b − a
                | return 0  if  b < −r ∨ a > r
                | return (b + r)/p  if  |b| ≤ r ∧ a < −r
                | return 1  if  |b| ≤ r ∧ |a| ≤ r
                | return 2r/p  if  b > r ∧ a < −r
                | (r − a)/p

Vn2(a,b,r) := | c ← min(a,b)
              | d ← max(a,b)
              | p ← d − c
              | t ← 0  if  d < −r ∨ c > r
              | otherwise
              |   | if  d ≥ r
              |   |   | p ← 2r/p  if  c < −r
              |   |   | t ← (r − c)/p  otherwise
              |   | otherwise
              |   |   | t ← (d + r)/p  if  c < −r
              |   |   | t ← 1  otherwise
              | t
```

Рис.12 Тексты программ коэффициента виньетирования
в линейном приближении

С помощью одной из этих программ может быть составлена простая программа определения размера симметричного отрезка (диафрагмы), доставляющего заданный коэффициент относительного перекрытия (коэффициент виньетирования в линейном приближении):

```
Dv(c,a,b) := | δ ← |a − b| / 1000
             | x ← δ
             | while  Vn1(a,b,x) < c
             |   x ← x + δ
             | x
```

Рис.13. Текст программы вычисления радиуса диафрагмы
с заданным коэффициентом виньетирования

Входными параметрами программы служат: коэффициент относительного перекрытия (*c*), и границы “светового пучка” (*a*,*b*). Программа возвращает радиус диафрагмы.

Программирование в среде MathCAD не поддерживает:

А) нумерацию строк;
Б) операцию прерывания естественного хода программы с выходом по метке;
В) организацию промежуточного вывода результатов;
Г) организацию подпрограмм и функций пользователя в теле программы;
Д) встроенный в текст программы комментарий.

3. Реализация метода Монте-Карло

Система MathCAD обладает большими возможностями по моделированию погрешностей с заданными характеристиками. Обычно функция плотности распределения вероятности случайной составляющей погрешности неизвестна. В этом случае задаются либо нормальным, либо равномерным законом распределения.

Для моделирования нормальной случайной величины необходимо задать три параметра: объём выборки *N*, математическое ожидание *M* и стандартное отклонение, или СКО (средне-квадратическое отклонение) σ. Используется встроенная функция rnorm. Ниже представлен фрагмент листинга, иллюстрирующего моделирование трёх нормально распределённых значений:

$$N := 3 \quad M := 1 \quad \sigma := 0.5 \quad V := \mathrm{rnorm}(N, M, \sigma) \quad V = \begin{pmatrix} 0.781 \\ 0.66 \\ 0.763 \end{pmatrix}$$

Для моделирования равномерно распределённой случайной величины используется встроенная функция runif. Входными параметрами этой функции являются: объём выборки *N*, левая *A* и правая *B* границы изменения случайной величины, причём $A<B$. Результат моделирования представлен ниже:

$$N := 3 \quad A := 0 \quad B := 2 \quad V := \mathrm{runif}(N, A, B) \quad V = \begin{pmatrix} 0.183 \\ 0.295 \\ 1.977 \end{pmatrix}$$

Для доступа к значениям смоделированного случайного вектора необходимо задать ранжированную целочисленную переменную или воспользоваться инструкциями по формированию циклов.

Для обработки случайной величины *V* MathCAD предлагает ряд функций:

mean(*V*) – среднее значение,
stdev(*V*) – СКО,
skew(*V*) – коэффициент асимметрии,
min(*V*) – минимальное значение,
max(*V*) – максимальное значение,
sort(*V*) – возвращает отсортированный по убыванию вектор,
csort(*V*,0) – возвращает отсортированный по возрастанию вектор.

Если при моделировании нормальной случайной величины стандартное отклонение неизвестно, но известны границы изменения случайной величины, то можно воспользоваться критерием 3σ и в качестве СКО использовать одну шестую часть диапазона изменения случайной величины. В качестве математического ожидания указывается середина интервала.

Моделирование систематической составляющей, постоянной или периодической, моделируется как соответствующая сумма случайной и регулярной составляющей, например:

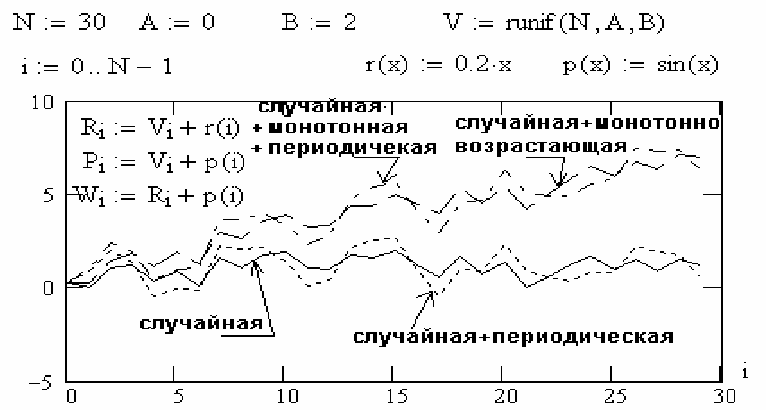


Рис.14. Графики моделированных погрешностей

4. Допуски на конструктивные параметры оптических элементов

Допуски на конструктивные параметры оптических систем изменяются в довольно узких пределах [2].

Допуск на радиус сферической поверхности суммируется из допуска на радиус пробного стекла, который в соответствии с ГОСТ 2786-76 для радиусов, меньших 1000 мм изменяется в диапазоне от 0,3 – 0,01%, и допуска на подгонку под пробное стекло, выраженного в числе колец Ньютона и изменяющегося, как правило, в пределах от 10 до 1 кольца.

В [2] рекомендуют задавать изменение радиуса кривизны в относительных единицах в пределах $\frac{\Delta r}{r} = 0{,}001 - 0{,}01$. Для плоской поверхности задаётся значение радиуса. В этом случае исходят из числа колец *N* и светового диаметра *D*. Радиус вычисляется по формуле:

$$R \approx 450 \frac{D^2}{N}.$$

Допуск на толщину линзы и воздушный промежуток заключён в интервале от 0,01 до 1,0 мм.

Допуск для показателей преломления определяется по ГОСТ 3514-76 и изменяется в пределах Δn=0,0002-0,002.

5. Модель центрированной оптической системы в параксиальном приближении

Центрированная оптическая система представляет собой последовательность оптических поверхностей и плоскости приёмника, имеющих общую ось симметрии, называемую оптической осью. Каждая поверхность ограничена круглой диафрагмой. Одна из диафрагм является апертурной диафрагмой, а другая полевой диафрагмой. Между поверхностями помещаются оптические среды с различными показателями преломления.

Реальная система в общем случае обладает аберрациями. Изображение строится негомоцентрическими пучками.

Можно ли построить идеальный прибор, дающий стигматическое изображение трёхмерного пространства и сохраняющий геометрическое подобие? Известны два примера абсолютного прибора: это плоское зеркало в однородной среде и “рыбий глаз” Максвелла для неоднородной среды с радиально изменяющимся показателем преломления.

“Рыбий глаз” Максвелла реализован с помощью преобразования инверсии (рис.15).

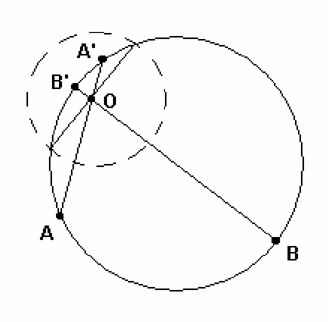


Рис.15. Ход лучей в среде “рыбий глаз” Максвелла

Показатель преломления изменяется радиально. В центре среды, в центре инверсии, в точке O он равен n_0 и с удалением от центра монотонно убывает по закону:

$$n(r) = \frac{n_0}{1 + \left(\frac{r}{a}\right)^2}. \qquad (1)$$

Если не требовать идеального преобразования всего трёхмерного пространства, а только какой-либо одной поверхности, то такую систему называют *абсолютной*. Абсолютное преобразование может быть осуществлено обычной сферической поверхностью, разделяющей две однородные среды. Такие поверхности называют апланатическими. Ход лучей в апланатической системе приведён на рис.16.

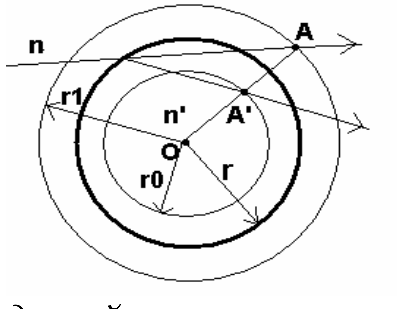


Рис.16. Ход лучей в апланатической системе

Апланатические системы идеально сопрягают две сферы с радиусами r_0 и r_1таких, что

$$r = r_0 r_1, \quad r_0 = \frac{n}{n'}, \quad r_1 = \frac{n'}{n}. \qquad (2)$$

Здесь также выполняется преобразование инверсии с центром в центре кривизны преломляющей сферической поверхности.

Согласно теореме Лиувилля-Каратеодори абсолютное преобразование есть преобразование либо инверсии, либо проективное преобразование, либо их комбинация.

Проективное преобразование двух пространств друг в друга осуществляется с помощью соотношений:

$$x' = \frac{F_1(x,y,z)}{F_0(x,y,z)}, \quad y' = \frac{F_2(x,y,z)}{F_0(x,y,z)}, \quad z' = \frac{F_3(x,y,z)}{F_0(x,y,z)},$$

где (3)

$$F_i = a_i x + b_i y + c_i z + d_i$$

Такие преобразования называют дробно-линейными. Обратное преобразование также оказывается дробно-линейным. Преобразование осуществляется между выделенными в пространствах плоскостями предмета и изображения, которые в результате параллельного сдвига заполняют всё трёхмерное пространство.

При переходе к осесимметричной системе, когда одна из осей координат является осью симметрии, остаётся 4 независимых переменных из первоначального числа 15. Выбор начала координат в точках фокусов отнимает ещё две степени свободы и упрощает преобразования, так, что требуется всего два параметра, называемые передним f и задним f' фокусным расстоянием. Преобразования приобретают предельно простой вид:

$$y_1 = \frac{f}{z_0} y_0, \quad x_1 = \frac{f}{z_0} x_0, \quad z_1 = \frac{f \cdot f'}{z_0}. \qquad (4)$$

В оптике раздел, называемый коллинеарная оптика, изучает свойства таких преобразований. Заметим, что в такой оптике нет оптических поверхностей, а есть математические объекты, которые по аналогии трактуют как точки предмета и изображения, есть прямые в обоих пространствах, которые можно трактовать как лучи, причём лучи образуют гомоцентрические пучки и изображения получаются стигматическими. Точка преобразуется в точку, прямая в прямую, плоскость в плоскость без нарушения углов между прямыми.

В таких пространствах определяют фокальные плоскости, главные плоскости, кардинальные точки.

Единственной идеальной оптической системой, описываемой с помощью проективного преобразования, реально является плоское зеркало.

5.1. Параксиальная оптика

Однако если задаться вопросом, а какими свойствами должна обладать реальная сферическая оптическая поверхность, чтобы изображение было стигматическим, т.е. образованным гомоцентрическими пучками.

Выберем систему координат, так чтобы пучки из пространства предмета распространялись, как этого требует ГОСТ 7427-76, слева направо в направлении оси *OZ*. Начало координат поместим в полюс сферической поверхности (рис.17):

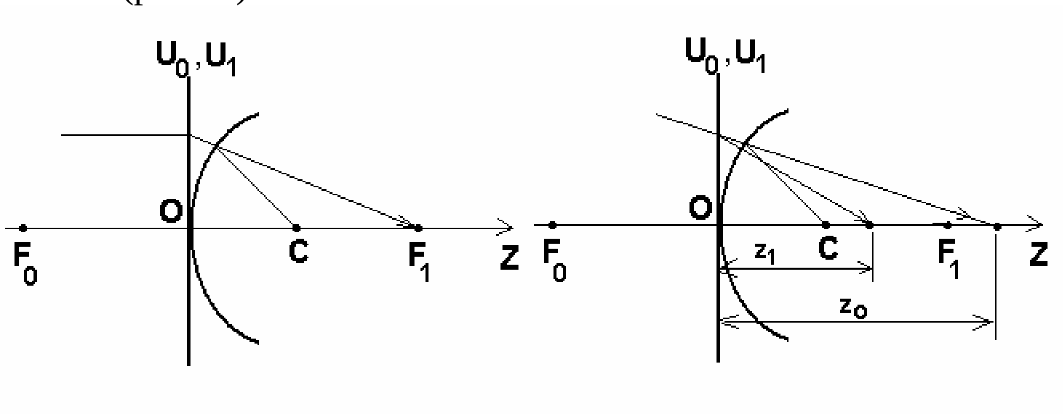


Рис.17. Положение кардинальных точек при преломлении
на сферической поверхности (слева)
и точек предмета и изображения (справа)

Оказалось, что в приближении лучей, распространяющихся близко к оптической оси, т.е. почти с нулевой апертурой, в параксиальном приближении, оптическая система, чтобы быть абсолютной, то есть давать стигматическое и подобное изображение предметной плоскости, должна удовлетворять инварианту Аббе:

$$n_0\left(\frac{1}{r}-\frac{1}{z_0}\right)=n_1\left(\frac{1}{r}-\frac{1}{z_1}\right), \qquad (5)$$

где n_0 и n_1 – показатели преломления в пространствах предмета и изображения r – радиус кривизны поверхности, z_0 - *Z*-координата осевой точки предмета, а z_1 – изображения.

Инвариант Аббе – есть мостик от коллинеарной оптики к параксиальной. При переходе от коллинеарной оптики к гауссовой (параксиальной) в соотношениях, описывающих преобразование оптической системой, появляется радиус кривизны в точке полюса поверхности. В параксиальном приближении все поверхности приближаются либо сферой, либо плоскостью.

Инвариант Аббе может быть преобразован к виду:

$$\frac{n_1}{z_1}-\frac{n_0}{z_0}=\frac{n_1-n_0}{r}=\Phi, \qquad (6)$$

здесь Ф – называется оптической силой поверхности.

Кроме того, главные плоскости проективного преобразования совпадают и проходят через вершину сферической поверхности. То есть, главные точки совпадают с полюсом сферической поверхности. Напомним, что главными плоскостями называются плоскости в пространствах предмета и изображения, в которых линейное увеличение равно единице.

Устремляя в инварианте Аббе поочерёдно z_0 и z_1 к минус и плюс бесконечности, находим аппликаты фокусов:

$$F_0 = -\frac{n_0 r}{n_1 - n_0}, \quad F_1 = \frac{n_1 r}{n_1 - n_0}. \tag{7}$$

В отличие от коллинеарной оптики, где начало координат помещают в точки фокусов, в параксиальной оптике, поддерживаемой ГОСТ 7427-76, фокусные расстояния отсчитывают от главных плоскостей, поэтому фокусные расстояния определяются соотношениями:

$$f_0 = -\frac{n_0 r}{n_1 - n_0}, \quad f_1 = \frac{n_1 r}{n_1 - n_0}. \tag{8}$$

Под фокусным расстоянием системы понимают заднее фокусное расстояние f_1, поэтому положительная линза имеет положительное фокусное расстояние.

Используя, соотношение для оптической силы поверхности, получаем соотношение:

$$-\frac{n_0}{f_0} = \frac{n_1}{f_1} = \Phi . \tag{9}$$

То есть оптическая сила может вычисляться по параметрам как среды в пространстве предметов (со знаком минус), так и в среде пространства изображения (со знаком плюс).

Оказывается, что соотношение для вычисления оптической силы справедливо не только для одной поверхности, но и для любой центрированной оптической системы.

5.2. Толстая линза

Для того чтобы построить параксиальную модель центрированной оптической системы кроме инварианта Аббе ничего не требуется. В начале рассмотрим систему из двух оптических поверхностей, представленную в виде толстой линзы (рис.21).

Оптическая сила линзы описывается формулой:

$$\Phi = \Phi_1 + \Phi_2 - \frac{t}{n_1}\Phi_1\Phi_2, \tag{10}$$

где n_1 – показатель преломления между поверхностями.

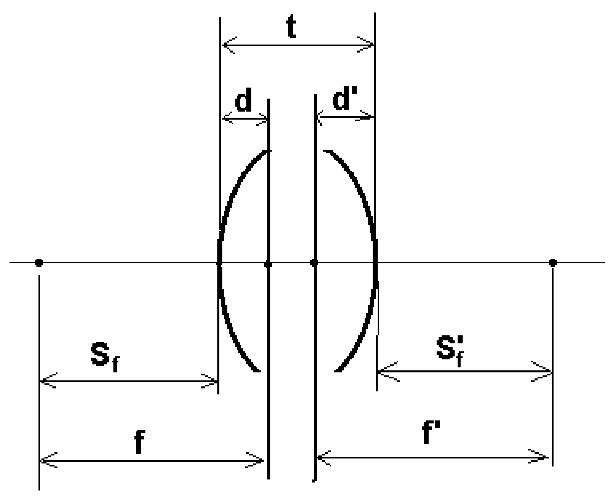


Рис.18. Толстая линза

Для проведения тестирования модели удобными являются формулы для определения фокусного расстояния и положений кардинальных точек толстой линзы в воздухе.

$$f' = -\frac{n \cdot r_1 \cdot r_2}{(n-1) \cdot [n \cdot (r_1 - r_2) - (n-1) \cdot t]}. \qquad (11)$$

$$d = \frac{n-1}{n \cdot r_2} t \cdot f', \quad d' = \frac{n-1}{n \cdot r_1} t \cdot f'$$

Кардинальные отрезки отсчитывают от главных плоскостей.

5.3. Центрированная оптическая система из произвольного числа поверхностей

В рамках коллинеарной оптики преломление происходит не на реальной поверхности, а на главной плоскости. Такие лучи называют нулевыми, имея в виду, что законы их распространения будут тем точнее, чем ближе апертурный угол луча к нулю.

С помощью нулевых лучей можно оценить габариты светового пучка и тем самым оценить размеры диафрагм. Если известна высота луча на предыдущей главной плоскости h_k, то его высота на последующей главной плоскости h_{k+1} определяется из рассмотрения подобных треугольников (рис.17).

Получаем:

$$h_{k+1} = h_k \; 1 - \frac{t_k}{z'_k} \;, \qquad (12)$$

здесь t_k – расстояние между текущей поверхностью и предыдущей, z'_k – задний отрезок после текущей k-той поверхности (рис.19).

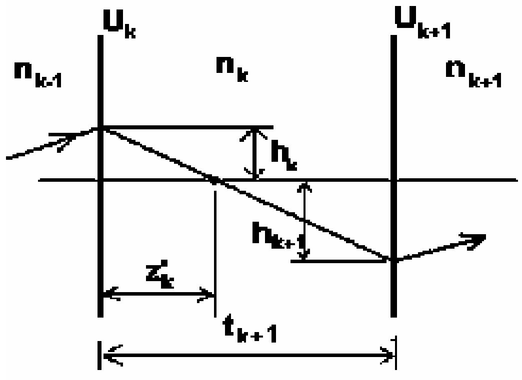


Рис.19. К определению световых высот и индексов параметров схемы

При расчёте оптической силы последовательности поверхностей по формуле (10) следует под параметром t понимать алгебраическое расстояние между задней главной плоскостью предыдущей части системы и передней главной плоскостью текущей поверхности, отсчитываемое по ходу луча. Для линзы это расстояние равно её толщине. Для системы поверхностей необходимо выполнять вычисления по определению положения задней главной плоскости рассчитанной части системы.

Определение положения задней главной плоскости

Оптическая сила линзы описывается формулой (10).Так как для сферической поверхности главные плоскости совпадают и проходят через полюс поверхности (рис.20), то t – есть и расстояние между полюсами поверхностей.

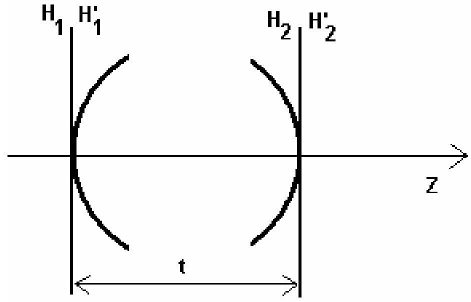


Рис.20. К определению оптической силы линзы

Формула (10) справедлива и для системы, содержащей более двух поверхностей. Однако в этом случае задняя главная плоскость предшествующей системы поверхностей не совпадает с полюсом последней поверхности. Положение задней главной точки задаётся отрезком d, отсчитываемым от полюса поверхности (рис.21) и d <0, если главная точка находится слева от полюса и d > 0 – в противном случае. На рисунке d <0.

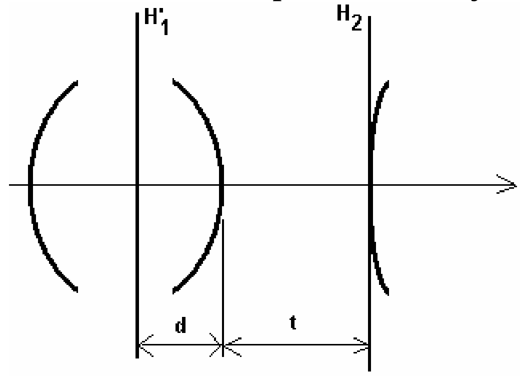


Рис.21. К определению оптической силы системы поверхностей

Отрезок d = 0 для первой поверхности системы. Далее d рассчитывается как разность заднего фокального отрезка z' и фокусного расстояния всей системы (рис.22).

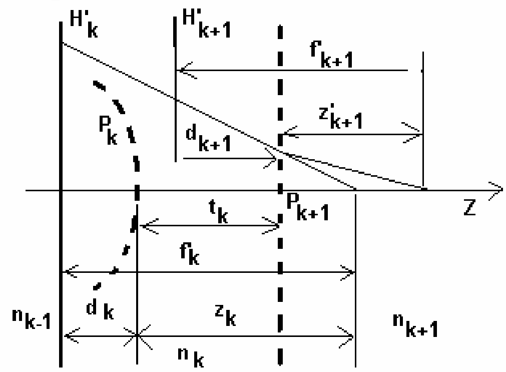


Рис.22. К определению положения главных поверхностей

Таким образом, для вычисления положения задней главной плоскости предыдущей части оптической системы требуется:

1) определить задний фокальный отрезок *k*-той поверхности по формуле:

$$z_k = \frac{h_k}{\alpha_{k+1}}, \qquad (13)$$

2) рассчитать текущее фокусное расстояние:

$$f_k = \frac{n_{k+1}}{\Phi}, \qquad (14)$$

где n_{k+1} - показатель преломления среды справа от текущей поверхности, Ф – оптическая сила части системы, включая k-тую поверхность.

3) определить задний кардинальный отрезок текущей поверхности:

$$d_{k+1} = z_k - f_k, \qquad (15)$$

4) пользуясь формулой (10) для вычисления оптической силы, на следующей поверхности применить подстановку:

$$t \equiv t_k - d_k, \qquad (16)$$

При вычислениях удобнее пользоваться не отрезками, а апертурными углами. Обозначим тангенс апертурного угла падения на *k*-тую главную плоскость как:

$$\alpha_k = \frac{h_k}{z_k}. \qquad (17)$$

Формулы (6), (10), (12), (13)-(16) основа для построения параксиальной модели оптической системы. Формулы (6) и (12) запишем с учётом (17). Выпишем эти формулы для составления модели центрированной оптической системы, состоящей из *N* поверхностей, *k*=1,2,…,*N*.

1) $\Phi_k = \frac{n_k - n_{k-1}}{r_k}$ 2) $\alpha_{k+1} = \frac{1}{n_{k+1}}(n_k\alpha_k + h_k\Phi_k)$ 3) $h_{k+1} = h_k - \alpha_{k+1}t_{k+1}$

4) $z_k = \frac{h_k}{\alpha_{k+1}}$ 5) $\Phi = \Phi + \Phi_k - \frac{t_k - d_k}{n_k}\Phi\Phi_k$

6) $f_k = \frac{n_{k+1}}{\Phi}$ 7) $d_{k+1} = f_k - z_k$ (18)

Заметим, что поскольку преломление никак не связано с физической реализуемостью акта преломления, то в коллинеарной оптике любой луч достигает плоскости изображения. Если сравнить рассчитанные по модели световые радиусы с радиусами соответствующих поверхностей, то можно обнаружить физическую нереализуемость системы. Даже если световой радиус немного больше радиуса кривизны, то система также может быть забракована по аберрационным соображениям.

Исходную информацию о системе удобно собрать в матрицу. Назовём её информационной матрицей системы Syst:

$$Syst := \begin{pmatrix} R_0 & R_1 & .... & R_{N-1} & D \\ t_0 & t_1 & .... & t_{N-1} & Z_0 \\ n_0 & n_1 & .... & n_{N-1} & n_N \end{pmatrix} \qquad (19)$$

Матрица состоит из N+1 столбца, где N – число поверхностей. В столбце, связанном с поверхностью, помещается радиус кривизны в полюсе поверхности, расстояние до предыдущей поверхности (t_0=0) и показатель преломления среды слева от поверхности. В последний столбец помещаются: диаметр входной диафрагмы D, передний отрезок Z_0 и показатель преломления среды в пространстве изображения.

Составим программу расчёта фокусного расстояния системы, определим также задний кардинальный отрезок и световые диаметры. В этом случае положим: передний апертурный угол α_0=0, $Z_0=\infty$. Блок-схема и текст программы Gauss(Syst) приведены ниже. Номера формул в блок-схеме соответствуют их порядку в (18).

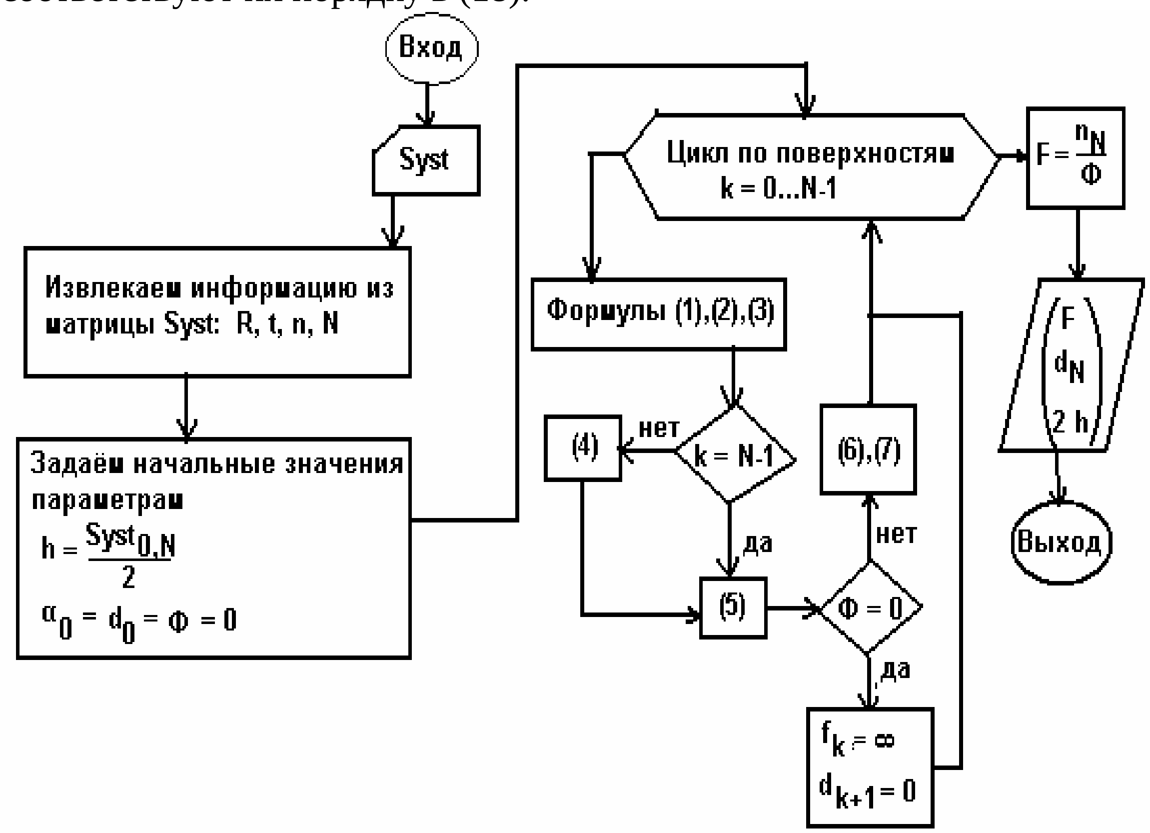


Рис.23. Блок-схема коллинеарной модели центрированной оптической системы.

Тестирование программы Cardinal

Тестирование выполним с помощью формул фокусного расстояния и кардинальных отрезков толстой линзы (11). Для случая тестирования двояковыпуклой линзы с радиусами кривизны 100 мм и толщиной 10 мм фрагмент документа MathCAD приведён ниже.

Проверка

$$f(r1,r2,n,d) := \frac{-n\cdot r1\cdot r2}{(n-1)\cdot[n\cdot(r1-r2)-(n-1)\cdot d]} \qquad f(100,-100,1.5,10) = 101.695$$

$$d'(r1,r2,n,d) := \frac{n-1}{n\cdot r1}\cdot d\cdot f(r1,r2,n,d) \qquad d'(100,-100,1.5,10) = 3.39$$

$$\mathrm{Cardinal}(S) = \begin{pmatrix} 101.695 \\ 3.39 \\ \{2,1\} \end{pmatrix}$$

$$
\mathrm{Gauss}(Syst) :=
\left|
\begin{array}{l}
r \leftarrow \left((Syst)^{T}\right)^{\langle 0\rangle} \\
t \leftarrow \left((Syst)^{T}\right)^{\langle 1\rangle} \\
n \leftarrow \left((Syst)^{T}\right)^{\langle 2\rangle} \\
N \leftarrow \mathrm{cols}(Syst) - 1 \\
h_0 \leftarrow \dfrac{Syst_{0,N}}{2} \\
\alpha_0 \leftarrow 0 \\
\Phi \leftarrow 0 \\
d_0 \leftarrow 0 \\
\text{for } k \in 0..N-1 \\
\quad \left|
\begin{array}{l}
\phi_k \leftarrow \mathrm{if}\left(r_k = \infty, 0, \dfrac{n_{k+1} - n_k}{r_k}\right) \\
\alpha_{k+1} \leftarrow \dfrac{1}{n_{k+1}} \cdot \left(n_k \cdot \alpha_k + h_k \cdot \phi_k\right) \\
z_k \leftarrow \mathrm{if}\left(\alpha_{k+1} \neq 0, \dfrac{h_k}{\alpha_{k+1}}, \infty\right) \\
h_{k+1} \leftarrow h_k - \alpha_{k+1} \cdot t_{k+1} \quad \text{if } k \neq N-1 \\
\Phi \leftarrow \Phi + \phi_k - \dfrac{t_k - d_k}{n_k} \cdot \Phi \cdot \phi_k \\
f_k \leftarrow \mathrm{if}\left(\Phi \neq 0, \dfrac{n_{k+1}}{\Phi}, \infty\right) \\
d_{k+1} \leftarrow \mathrm{if}\left(\Phi \neq 0, z_k - f_k, 0\right)
\end{array}
\right. \\
F \leftarrow \dfrac{n_N}{\Phi} \\
\begin{pmatrix} F \\ z_{N-1} \\ 2 \cdot h \end{pmatrix}
\end{array}
\right.
$$

Рис.24. Текст программы Gauss расчёта фокусного расстояния, последнего отрезка и световых диаметров центрированной линзовой системы

6. Моделирование погрешностей на примере линзовой системы

6.1. Определение передаточных функций погрешности фокусного расстояния

Влияние первичных погрешностей параметров линзы и внешних факторов, например, температуры, может быть получено методом дифференцирования соответствующего выражения [1, стр.77]. Для многолинзовой оптической системы рациональнее использовать метод моделирования. Рассмотрим пример моделирования передаточной функции влияния погрешности радиусов кривизны плоско-выпуклой линзы.

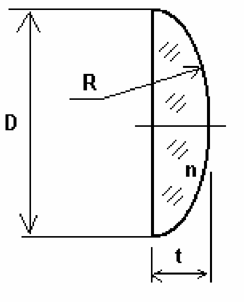


Рис.25. Исследуемая в примере линза.
R = -500 мм, t = 10 мм, D= 30 мм, n=1,5

Фокусное расстояние плоско-выпуклой линзы (11) выражается формулой:

$$f' = \frac{R}{n-1}, \qquad (20)$$

поэтому передаточная функция погрешности криволинейной поверхности есть величина постоянная:

$$A_R = \frac{-1}{n-1}, \qquad (21)$$

(минус перед формулой (21) учитывает правило знаков для радиусов).

Для модельного определения передаточной функции составим две системные матрицы: номинальную матрицу:

$$Syst := \begin{pmatrix} \infty & R & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix}$$

и матицу, содержащую первичную погрешностью исследуемого фактора:

$$Syst1 := \begin{pmatrix} \infty & R+\Delta & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix}$$

Передаточная функция определится с помощью программы Gauss следующим образом:

$$A(\Delta) := \frac{Gauss(Syst1)_0 - Gauss(Syst)_0}{\Delta}$$

Нулевой индекс после обращения к программе Gauss возвращает первое значение выводимого программой вектора, то есть фокусное расстояние системы.

В данном примере из (21) и из расчётов получаем $A(\Delta) = -2$.

Методом дифференцирования не может быть определена передаточная функция погрешности плоской поверхности. Величина первичной погрешности определяется стрелкой прогиба:

$$\Delta = \frac{D^2}{8R_0}, \qquad (22)$$

где ρ – радиус кривизны поверхности, приближающей плоскость. Погрешность плоской поверхности задают числом полос Ньютона N по формуле $R_0 \approx 450\frac{D^2}{N}$. При изменении числа полос от 1 до 10, нижняя граница радиуса первой поверхности линзы при D = 30 мм изменяется в пределах: $|R_0| = 4\ 10^4 - 4\ 10^5$ мм.

Системная матрица линзы с погрешностью запишется в виде:

$$\text{Syst1} := \begin{pmatrix} R_0 & r & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix}.$$

Моделирование передаточной функции осуществляется также с помощью программы Cardinal по формуле:

$$A(R_{02}) := \frac{Gauss(Syst0)_0 - Gauss(Syst)_0}{D^2} 8R_{02} \qquad A(R_{02}) = 8.647\ x\,10^3$$

$$R_{02} = 4\ x\,10^5 \qquad D = 30$$

Как видим, передаточная функция первичной погрешности кривизны плоской поверхности линзы на несколько порядков больше, чем передаточная функция погрешности криволинейной поверхности.

6.2. Суммирование частичных погрешностей

Практические правила расчётного суммирования результирующей погрешности состоят в следующем [2, стр. 205]:

1) Погрешность по абсолютному значению всегда меньше самой измеряемой величины.

2) Необходимо учитывать все корреляционные связи составляющих погрешности. Если точное значение коэффициента корреляции найти не возможно, то погрешности подразделяют на сильно коррелированные ($|\rho| > 0{,}7$), тогда полагают $\rho = \pm 1$. В противном случае полагают, что $\rho = 0$.

3) Для каждой частичной погрешности должно быть найдено СКО.

4) Частичные погрешности суммируются по классам: класс аддитивных и класс мультипликативных погрешностей.

5) Суммирование систематических ошибок при равномерном законе распределения составляющих ведётся по формуле:

$$\theta = \min\left(k\sqrt{\sum_{i=1}^{m}\theta_i^{\,2}},\ \sqrt{\sum_{i=1}^{m}\theta_i}\right), \qquad (23)$$

где θ_i – граница погрешности (полуинтервал), коэффициент *k* зависит от числа слагаемых m и доверительной вероятности *P*. При усреднении по числу слагаемых коэффициенты приведены в таблице:

Таблица 1. Усреднённые эмпирические коэффициенты при суммировании систематических погрешностей

P	*k*
0,90	0,95
0,95	1,1
0,99	1,4

6) Суммирование случайных погрешностей.

Оценка дисперсии погрешности χ измеряемой величины *y*(*x*) определяется по формуле:

$$s^2 = \frac{1}{N-1}\sum_{k=1}^{N}\chi_k^{\,2} = \frac{1}{N-1}\sum_{k=1}^{N}(y_k - \bar{y})^2, \quad \bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k., \qquad (24)$$

где *N* – число измерений, $\bar{y}$ - среднее значение.

Оценка дисперсии суммарной погрешности случайных частичных погрешностей определяется выражением:

$$s_\Sigma^2 = \sum_{i=1}^{m} s_i^2 + 2\sum_{i<j}^{m}\rho_{i,j} s_i s_j\ . \qquad (25)$$

Если суммируется две случайные частичные погрешности, то:

$$s^2 = s_1^2 + s_2^2 + 2\rho\cdot s_1 s_2\,. \qquad (25а)$$

Откуда получаем выражение для коэффициента корреляции:

$$\rho = \frac{s^2 - s_1^2 - s_2^2}{2 s_1 s_2}\,. \qquad (26)$$

Коэффициент корреляции может быть найден по формуле (26). Для определения дисперсий частичных погрешностей может быть использован метод моделирования.

7) Суммирование систематических и случайных погрешностей.

Систематическая погрешность, присутствующая во всех отсчётах, не усредняется при статистической обработке, тогда как случайная составляющая за счёт многократных измерений может быть уменьшена в много раз.

Согласно ГОСТ 8.207-76 погрешность результата измерений определяется по правилам:

а) если границы не исключённой систематической погрешности θ и СКО случайной погрешности *s* удовлетворяют неравенству:

$$\theta < 0{,}8\,s, \qquad (27)$$

то следует пренебречь систематической ошибкой, б) если же имеет место равенство

$\theta > 8\,s,$ (28)

то, наоборот, следует пренебречь случайной составляющей.

в) В общем случае доверительный интервал суммарной погрешности определяется по формуле:

$\Delta = 2(|\,\theta\,| + t\cdot s),$ (29)

где t – коэффициент Стъюдента.

На рис.26 отображены все три ситуации.

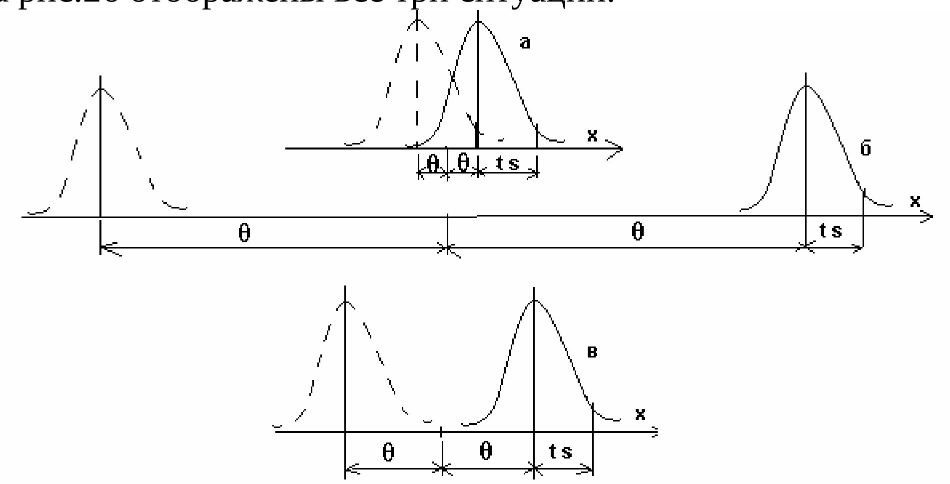


Рис.26. Суммирование систематической и случайной погрешностей

В качестве примера определим коэффициент корреляции частичных случайных погрешностей, связанных с погрешностями радиусов кривизны плосковыпуклой линзы. Промоделируем частичные погрешности фокусного расстояния в зависимости от первичных погрешностей радиусов кривизны двух поверхностей и их суммарную погрешность. Для этого определим три вектора, элементами которых являются системные матрицы со случайными параметрами.

Пусть объём выборки M=200. Первичную погрешность радиуса кривизны первой плоской поверхности линзы будем рассматривать как случайную величину, распределённую равномерно на интервале:

$$(-R_{01}, -R_{02}) \cup (R_{01}, R_{02}),$$

где R_{01}, R_{02} – границы радиусов кривизны, соответствующие интервалу полос Ньютона от N=1 до 10. Напомним, что $R_0 \approx 450\dfrac{D^2}{N}$.

Фрагмент документа MathCAD, на котором осуществляется моделирование данной первичной погрешности, приведён ниже.

$M := 200 \qquad R_{01} := 4\cdot 10^4 \qquad R_{02} := 4\cdot 10^5$

$r1 := \mathrm{runif}\left(\frac{M}{2}, -R_{02}, -R_{01}\right) \qquad r2 := \mathrm{runif}\left(\frac{M}{2}, R_{01}, R_{02}\right) \qquad r0 := \mathrm{stack}(r1, r2)$

$r0^T =$

	0	1	2	3	4	5
0	$-1.067\cdot 10^5$	$-3.931\cdot 10^5$	$-2.854\cdot 10^5$	$-2.053\cdot 10^5$	$-8.658\cdot 10^4$	$-1.319\cdot 10^5$

В приведённом фрагменте используется встроенная функция runif, возвращающая вектор случайной величины, распределённой по равномерному закону. Входными параметрами функции служат: объём выборки *M* и границы, причём, левая граница строго меньше правой. Две случайные величины объединяются в один столбец с помощью встроенной функции stack.

Первичную погрешность радиуса кривизны криволинейной поверхности промоделируем как случайную величину с нормальным распределением. Среднее значение случайной величины равно номинальному значению радиуса *R*. Границы погрешности выбираются из рекомендованного интервала $\Delta \in (0{,}001;\, 0{,}01)R$. Обычно, основываясь на 90% уровне доверительной вероятности, полагают, что СКО погрешности может быть оценена как:

$$s = \frac{\Delta}{1{,}6}. \qquad (30)$$

Соответствующий фрагмент документа MathCAD, моделирующего нормально распределённую первичную погрешность приведён ниже.

$R := -100 \qquad \Delta := 0.01 \cdot |R| \qquad \sigma := \frac{\Delta}{1.6} \qquad r1 := \text{rnorm}(M, R, \sigma)$

$r1^T =$

	0	1	2	3	4	5	6
0	-100.091	-100.586	-100.08	-99.635	-99.165	-99.954	-101.264

Здесь используется встроенная функция rnorm, которая возвращает нормально распределённый вектор случайной величины. Входными параметрами служат: объём выборки, среднее значение (математическое ожидание) и стандартное отклонение (СКО).

Имея в распоряжении данные случайные векторы, можно создать соответствующие векторы из системных матриц:

$i := 0..M-1 \qquad n := 1.5 \qquad t := 10 \qquad D := 20$

$$Syst := \begin{pmatrix} \infty & R & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix} \quad Syst0_i := \begin{pmatrix} r0_i & R & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix} \quad Syst1_i := \begin{pmatrix} \infty & r1_i & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix} \quad Syst2_i := \begin{pmatrix} r0_i & r1_i & D \\ 0 & t & \infty \\ 1 & n & 1 \end{pmatrix}$$

В этом фрагменте для оперирования с вектором вводится ранжированная переменная *i*. Далее с помощью программы Cardinal моделируем частичные погрешности, рассчитываем СКО и определяем коэффициент корреляции.

При определении СКО мы пользуемся встроенной функцией stdev.

В результате расчёта получили, что данные первичные погрешности практически не коррелированны.

$F := \mathrm{Cardinal}(Syst)_0 \qquad F = 1\times10^3$

$\delta0_i := \mathrm{Cardinal}(Syst0_i)_0 - F \qquad \sigma0 := \mathrm{stdev}(\delta0) \qquad \sigma0 = 3.965$

$\delta1_i := \mathrm{Cardinal}(Syst1_i)_0 - F \qquad \sigma1 := \mathrm{stdev}(\delta1) \qquad \sigma1 = 5.965$

$\delta2_i := \mathrm{Cardinal}(Syst2_i)_0 - F \qquad \sigma2 := \mathrm{stdev}(\delta2) \qquad \sigma2 = 7.329$

$$\rho := \frac{\sigma2^2 - \sigma1^2 - \sigma0^2}{2\sigma0\cdot\sigma1} \qquad \rho = 0.051$$

Для исследования погрешностей используют построение гистограммы. Гистограмма является оценкой функции плотности распределения вероятности.

6. 3. Построение гистограммы в среде MathCAD

Пусть имеется выборка результатов из n параллельных опытов: $Y=\{Y_1,Y_2,\ldots,Y_n\}$.

1. Строится вариационный ряд, т.е. все данные располагаются в порядке возрастания. Это осуществляется с помощью функции:

$$Ys=\mathrm{scort}(Y,0).$$

Данная функция может применяться к матрице, второй параметр указывает на номер столбца.

2. Находят минимальное и максимальное и интервал изменения случайной величины *Y:*

$$Y\mathrm{min}=\min(Y), \quad Y\mathrm{max}=\max(Y), \quad RY=Y\mathrm{max}-Y\mathrm{min}.$$

3. Выбирают число интервалов разбиения m. Для этого используют эмпирические соотношения:

$$m_{\min}=0{,}55n^{0{,}4}, \quad m_{\max}=25/11m_{\min},$$

при этом предпочтение отдают нечётному значению числа интервалов, в этом случае середина размаха выборки будет приходиться на срединный интервал разбиения.

Если в качестве числа интервалов выбрать среднее:

$$m = floor\left(\frac{m_{\min}+m_{\max}}{2}\right),$$

где функция floor возвращает целую часть числа, то автоматический выбор нечётного значения числа интервалов можно осуществить, например, с помощью функции if, имеющей три входных параметра: условие, возвращаемое значение при выполнении этого условия и возвращаемое значение, если данное условие не выполняется:

$$m = if\left(m - 2\,floor\left(\frac{m}{2}\right) = 0, m+1, m\right).$$

4. Определяется ширина интервала разбиения: $I = \dfrac{RY + 10^{-9}}{m}$. Размах выборки чуть увеличивают, чтобы крайние значения попали в интервалы разбиения

5. Определяют границы интервалов. Для этого вводят ранжированную переменную j=0..m и тогда:

$$\mathrm{int}_j = Y_{\min} + j*I$$

6. Распределение выборки по интервалам осуществляется с помощью встроенной функции hist, в которой первым входным параметром является вектор границ интервалов, а вторым – выстроенные по возрастанию значения случайной величины.

$$H = \mathrm{hist}(\mathrm{int}, Ys).$$

Размерность вектора H m-1 на единицу меньше размерности вектора int границ выборки.

7. Для построения графика (гистограммы или полигона) можно воспользоваться дискретной функцией $H_j(j)$ – функцией номера интервала или функцией $H_j(\mathrm{int}_j + I/2)$, – функцией середины интервала.

В качестве примера рассмотрим построение гистограммы суммарной погрешности радиусов кривизны линзы. Построим график погрешности.

В MathCAD построение двумерных графиков осуществляется с помощью палитры графиков (нажатая кнопка на рисунке) на панели “Math”

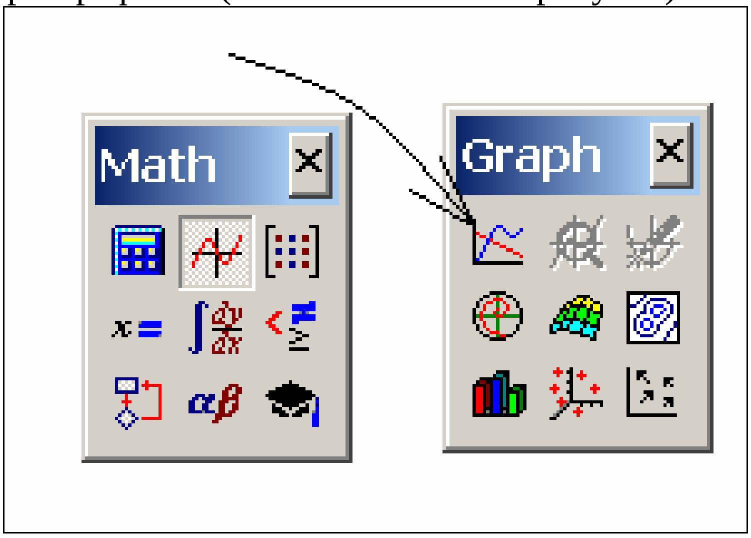


Рис.27. Панель “Math” (Математика) с нажатой кнопкой вызова палитры построения графиков “Graph”. На палитре отмечена кнопка вызова шаблона двумерных графиков. Горячая клавиша: @ (Shift+2)

Место для введения значений функции отмечено чёрным прямоугольником в левой части шаблона, а для введения аргумента – в нижней его части. Щёлкнув мышью вне поля графика, мы получаем следующий график погрешности.

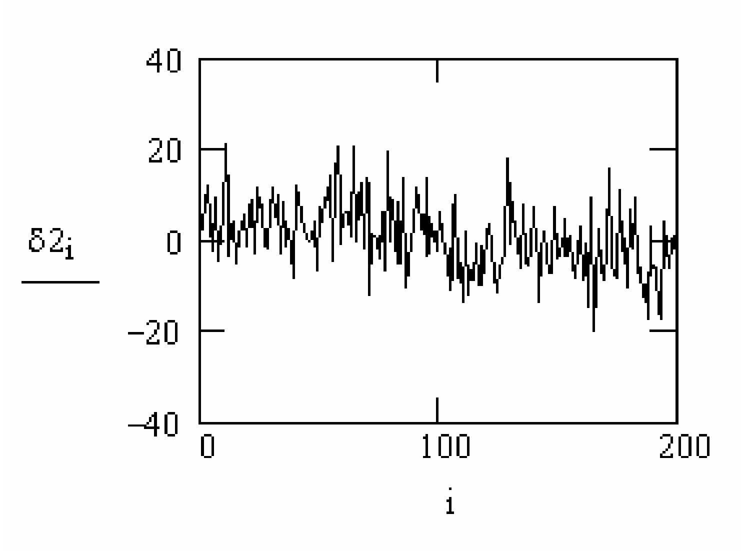


Рис.28. График суммарной погрешности фокусного расстояния в зависимости от первичных погрешностей радиусов кривизны линзы

Как видим, методика, применённая при моделировании первичной погрешности плоской поверхности: первая половина значений получена, когда плоская поверхность вогнута, а вторая – при выпуклой форме, проявилась в виде постоянной систематической ошибки. Первая половина графика сдвинута вверх, вторая – вниз. Очевидно, что реально данная систематическая ошибка отсутствует.

Строим вариационный ряд.

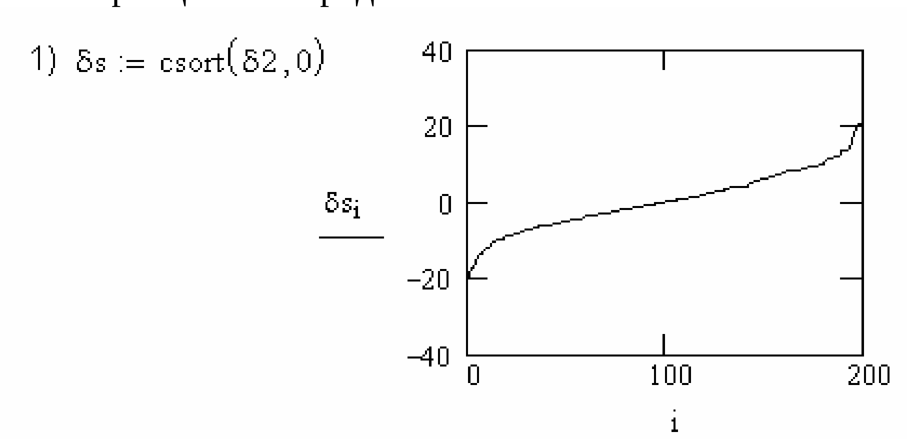


Рис.29. График вариационного ряда суммарной погрешности

Проводим вычисления:

2) $\delta min := min(\delta 2)$ $\delta max := max(\delta 2)$ $R\delta := \delta max - \delta min$

$\delta min = -20.01$ $\delta max = 21.101$ $R\delta = 41.111$

3) $m_{min} := 0.55 \cdot M^{0.4}$ $m_{max} := \frac{25}{11} \cdot m_{min}$ $m := \frac{floor(m_{min} + m_{max})}{2}$

$m_{min} = 4.579$ $m_{max} = 10.407$ $m = 7$

4) $I := \frac{R\delta + 10^{-9}}{m}$ $I = 5.873$ (I - интервал или квант)

5) $j := 0..m$ $int_j := \delta min + j \cdot I$

6) $H\delta := hist(int, \delta s)$ Hδ - вектор частотностей попадания случайной величины в построенные интервалы

$H\delta^T = (6\ \ 18\ \ 52\ \ 57\ \ 45\ \ 16\ \ 6)$

7) $\delta_j := int_j + \frac{I}{2}$ δ_j - середины интервалов

Чтобы построить гистограмму, вначале строим обычный двухмерный график, а далее осуществляем его форматирование.

Вызов вкладок меню форматирования осуществляется или с помощью строки Главного Меню документа “Format” (Форматирование) и опции в нём “Graph” (Графика), или, что удобнее, двойным щелчком левой кнопки мыши на графике. Выбрав вторую вкладку “Traces” (Траектории), а в ней опцию “Type”(Тип линии), с помощью которой выбираем тип “Bar” (Полоса). Нажимаем кнопку “OK” и получаем график гистограммы.

Чтобы на графике гистограммы построить полигон погрешности, в левом поле графика через запятую вводим ещё один экземпляр вектора частотностей *Hδ*. В результате проведённых операций получаем следующие графики

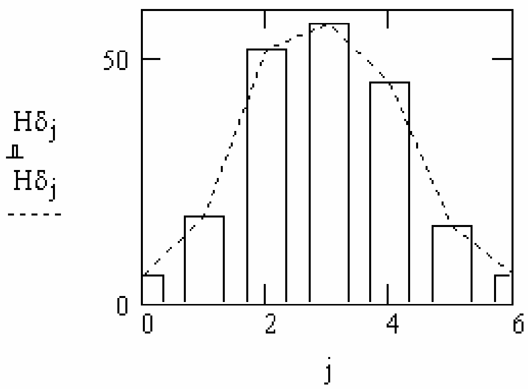


Рис.30. Гистограмма и полигон суммарной погрешности

фокусного расстояния линзы

Заметим, что хотя значение вектора частотностей $H\delta_m$ не определено (j=0,1,..,7), это не является преградой при построении графиков на рис.25. MahtCAD это значение просто игнорирует.

7. Вычисление положения и размеров входного зрачка

Апертурная диафрагма ограничивает пучок от осевой точки предмета. Её изображение через предшествующую часть системы называется входным зрачком (рис.31). Угол, составляемый лучом, идущим из осевой точки плоскости предмета на край входного зрачка, называется передним апертурным углом. Напомним свойства зрачка.

1) С положением входного зрачка связывается отображение перспективы: если он находится слева от предмета, то перспектива нормальная: с удалением от системы угловой размер предмета уменьшается, если справа, то перспектива неестественная: предмет удаляется, а его угловой размер увеличивается, она называется гиперцентрической. А если входной зрачок находится в бесконечности, то такая перспектива называется телецентрической: угловой размер предмета не меняется.

2) Апертурный угол определяет светосилу системы. Светосила – есть отношение освещённости изображения к яркости предмета, и определяется формулами:

$$H = \frac{\pi \cdot \tau \cdot \sin^2(\alpha_0)}{K}, \quad \begin{array}{l} K = V^2 - \textit{площадной предмет} \\ K = V - \textit{линейный предмет} \\ K = 1 - \textit{точечный предмет} \end{array}, \qquad (30)$$

где τ- коэффициент пропускной способности оптической системы.

Если предмет и входной зрачок находятся на конечном расстоянии, то для расчёта апертурной диафрагмы необходимо добавить апертурный луч 2. Луч 1 используется для вычисления фокусного расстояния.

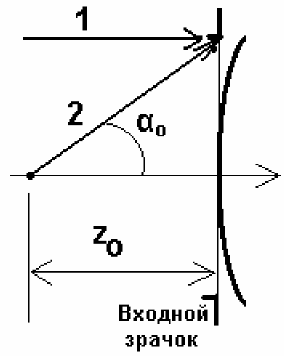


Рис.31. К определению апертурного угла

Пусть задано положение и размер входного зрачка. Первоначально мы полагали, что апертурной диафрагмой является диафрагма первой поверхности – она же являлась входным зрачком.

В общем случае апертурная диафрагма может совпадать с любой из диафрагм, ограничивающих оптические поверхности, или не совпадать и находиться или в пространстве предмета, или в пространстве изображения, или между оптическими поверхностями.

Исходя из размера апертурной диафрагмы, мы определим минимальные размеры всех материальных диафрагм. Для этого необходимо в системную матрицу внести информацию об апертурной диафрагме и, если она не совпадает ни с какой из оправ поверхностей, то описать её как некую оптическую поверхность в воздухе. Системная матрица может иметь такой вид:

$$Syst = \begin{matrix} R_0 & R_1 \ldots \infty & R_{k+1} \ldots R_{N-1} & A & \beta \\ t_0 & t_1 \quad t_{k-1} \quad t_k & \ldots t_{N-1} & z_0 & z_{BX} \\ n_0 & n_1 \quad 1 \quad 1 & \ldots \; n_{N-1} & n_N & k \end{matrix} . \qquad (31)$$

Матрица имеет N столбцов, описывающих параметры оптических поверхностей. Если апертурная диафрагма самостоятельная единица, то ей отводится k-столбец. Радиус поверхности $R_k=\infty$, поверхность слева и справа окружена воздушной средой n=1. Диаметр апертурной диафрагмы A помещаем в нулевую строку N-го столбца

Добавляется ещё один столбец справа: в первую строку помещаем величину полевого угла, вторая строка – резервная, куда в последствии можно поместить расстояние от первой поверхности до входного зрачка, а в третью – номер поверхности апертурной диафрагмы.

Сначала находим изображение апертурной диафрагмы через предшествующую часть оптической системы, то есть, рассчитываем часть системы в обратном ходе. Это изображение является входным зрачком системы.

Для вычисления положения входного зрачка необходимо рассчитывать прохождение через систему не одного, осевого как в программе Cardinal, а двух лучей: осевого и апертурного. С помощью осевого луча определятся положение текущей задней главной плоскости, а с помощью апертурного - положение точки изображения. В исходной программе передний апертурный угол α_A=0. Извлечём из системной матрицы вида (19) расстояние до плоскости предмета и зададим величину переднего апертурного угла следующим образом:

$$z_A = Syst_{1,N}, \quad \alpha_A = \frac{D}{2z_0}, \qquad (32)$$

N – число поверхностей оптической системы.

Для определения изображения апертурной диафрагмы требуется рассчитать прохождение двух лучей: осевого и апертурного (рис.32).

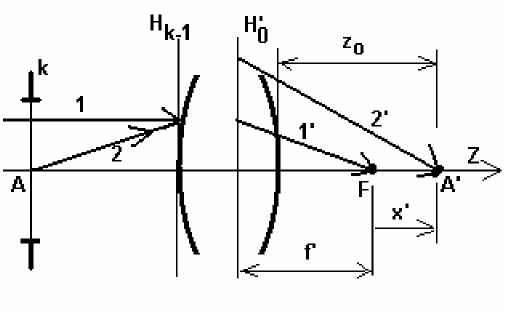


Рис.32. К расчёту изображения апертурной диафрагмы

Положение входного зрачка определяется величиной заднего отрезка z_k, а диаметр рассчитывается по формуле:

$$A_{ВЗ} = A \cdot |V| = A \cdot \left|\frac{x'}{f'}\right| = A\left|\frac{z_k - z_f}{f'}\right|, \qquad (33)$$

здесь V – линейное увеличение системы.

Текст программы приведён на рис.33. Выходными параметрами программы являются: фокусное расстояние системы, задние отрезки оптических поверхностей (расстояние от последней поверхности до изображения). Последний отрезок задаёт положение входного зрачка и с помощью него рассчитывается линейный диаметр входного зрачка.

Особенностями данной программы являются оперирование векторными данными: оба луча трассируются программой параллельно. Углы, высоты и отрезки представляют собой векторы.

Кроме того, с помощью данной программы можно промоделировать влияние наклона оптических поверхностей в результате торцевого биения оправ, промежуточных колец и опорных поверхностей. Углы наклона поверхности поместим в дополнительную нижнюю строку системной матрицы Syst.

$$\mathrm{Pupil}(Sy) :=$$

$$r \leftarrow \left((Sy)^{T}\right)^{\langle 0 \rangle}$$

$$t \leftarrow \left((Sy)^{T}\right)^{\langle 1 \rangle}$$

$$n \leftarrow \left((Sy)^{T}\right)^{\langle 2 \rangle}$$

$$\delta\alpha \leftarrow \left(Sy^{T}\right)^{\langle 3 \rangle}$$

$$N \leftarrow \mathrm{cols}(Sy) - 1$$

$$\Phi \leftarrow 0$$

$$d_0 \leftarrow 0$$

$$\text{for } i \in 0..N-1$$

$$\quad h_i \leftarrow \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\quad \alpha_i \leftarrow \begin{pmatrix} 0 \\ 0 \ \text{ on error } \ \frac{1}{t_N} \end{pmatrix}$$

$$\text{for } k \in 0..N-1$$

$$\quad z_k \leftarrow h_0$$

$$\quad (\alpha_k)_1 \leftarrow (\alpha_k)_1 - \delta\alpha_k$$

$$\quad \phi_k \leftarrow \mathrm{if}\left(r_k = \infty, 0, \frac{n_{k+1} - n_k}{r_k}\right)$$

$$\quad \alpha_{k+1} \leftarrow \frac{1}{n_{k+1}} \cdot (n_k \cdot \alpha_k + h_k \cdot \phi_k)$$

$$\quad \text{for } i \in 0..1$$

$$\quad\quad (z_k)_i \leftarrow \mathrm{if}\left[(\alpha_{k+1})_i \ne 0, \frac{(h_k)_i}{(\alpha_{k+1})_i}, \infty\right]$$

$$\quad s'_k \leftarrow (z_k)_1$$

$$\quad h_{k+1} \leftarrow h_k - \alpha_{k+1} \cdot t_{k+1} \ \text{ if } \ k \ne N-1$$

$$\quad \Phi \leftarrow \Phi + \phi_k - \frac{t_k - d_k}{n_k} \cdot \Phi \cdot \phi_k$$

$$\quad f_k \leftarrow \mathrm{if}\left(\Phi \ne 0, \frac{n_{k+1}}{\Phi}, \infty\right)$$

$$\quad d_{k+1} \leftarrow \mathrm{if}\left[\Phi \ne 0, f_k - (z_k)_0, 0\right]$$

$$F \leftarrow \frac{n_N}{\Phi}$$

$$B \leftarrow Sy_{0,N} \cdot \left| \frac{(z_{N-1})_0 - (z_{N-1})_1}{F} \right|$$

$$\begin{pmatrix} F \\ s' \\ B \end{pmatrix}$$

Рис.33. Текст программы Pupil

$$Syst := \begin{pmatrix} R_0 & R_1 & .... & R_{N-1} & D \\ t_0 & t_1 & .... & t_{N-1} & z_0 \\ n_0 & n_1 & .... & n_{N-1} & n_N \\ \delta\alpha_0 & \delta\alpha_1 & .... & \delta\alpha_{N-1} & 0 \end{pmatrix} \quad (34)$$

При подготовке данных для цикла по поверхностям в программе Pupil (рис. 33) происходит извлечение данной информации в вектор δα. В цикле происходит корректировка угла в соответствии с наклоном текущей поверхности.

Угол наклона определятся величиной торцевого биения поверхности δτ и диаметром оправы *D*.

$$\delta\alpha = \frac{\delta\tau}{D}. \quad (35)$$

Тестирование программы

Тестирование программы Pupil проведём с помощью формулы отрезков для линзы в воздухе.

$$S' = \frac{S \cdot f'}{S + f'}, \quad (36)$$

где *S* – передний отрезок, расстояние от осевой точки предмета до осевой точки передней главной плоскости, а *S’* – задний отрезок, расстояние от осевых точек задней главной плоскости и изображения. Определим системную матрицу Syst для плосковыпуклой линзы с радиусом задней поверхности 100, толщиной 10, диаметром 20 мм. Показатель преломления – 1.5.

Поскольку входным параметром для программы является расстояние от осевой точки предмета до осевой точки первой поверхности, то необходимо сначала вычислить передний отрезок. Задний отрезок и расстояние от осевой точки изображения до последней поверхности для линзы выбранной формы совпадают, так как задняя главная плоскость плосковыпуклой линзы проходит через полюс поверхности.

Здесь следует сделать замечание. В вычислительной оптике под передним и задним отрезком понимают именно расстояния от осевых точек передней и задней поверхности и формула отрезков (36) в этом случае справедлива только для линзы с нулевой толщиной. Для толстой же линзы необходимы поправки, сделанные в приведённом ниже тестировании.

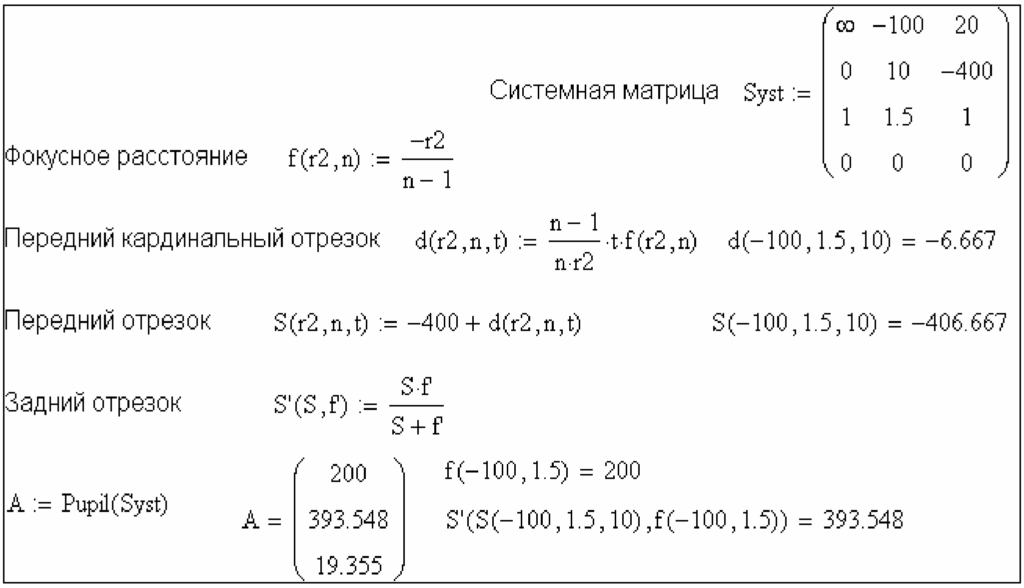


Рис.34. Фрагмент документа MahtCad по тестированию программы Pupil. (В отличие от текста программы здесь происходит вывод только последнего заднего отрезка)

В качестве примера рассмотрим окуляр Эрфле [3, стр.394] (рис.35)

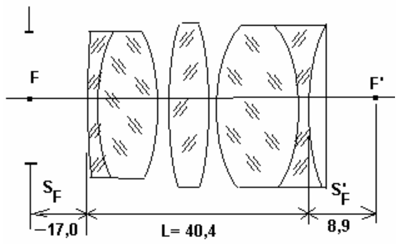


Рис.35. Окуляр Эрфле

Таблица 2. Конструктивные параметры окуляра Эрфле

r	*d*	*N*	Марка стекла	*D*св
170,23	1,8	1,6199	(Ф13)	32,5
34,42				
	13,8	1,5163	(К8)	
-29,41				
	0,25	1		
70,78				
	7,6	1,5163	(К8)	36,5
-70,78				
	0,25	1		
31,89				
	15,0	1,5163	(К8)	
-31,89				36,5
	1,7	1,6199	(Ф13)	
56,01				

Определим передаточную функцию торцевого биения оправы средней одиночной линзы.

Пределы изменения первичной погрешности |δτ|<0,01. Угол наклона линзы:

$$|\delta\alpha| < \frac{0,01}{36,5} 2 = 5,5 \cdot 10^{-4}.$$

Системная матрица как функция от величины торцевого биения средней линзы имеет вид:

$$\mathrm{Syst1}(\tau) := \begin{pmatrix} 170.23 & 34.42 & -29.41 & 70.78 & -70.78 & 31.89 & -31.89 & 56.01 & 32.5 \\ 0 & 1.8 & 13.8 & 0.25 & 7.6 & 0.25 & 15.0 & 1.7 & -17.0 \\ 1 & 1.6199 & 1.5163 & 1 & 1.5163 & 1 & 1.5163 & 1.6199 & 1 \\ 0 & 0 & 0 & \frac{\tau}{18.25} & \frac{\tau}{18.25} & 0 & 0 & 0 & 0 \end{pmatrix}$$

Предметную точку помещаем в передний фокус окуляра. Откликом на первичную погрешность торцевого биения будет изменение угла отклонения луча в пространстве изображения. Для этого изменим часть выходных параметров программы Pupil следующим образом.

$$\begin{bmatrix} F \\ (z_{N-1})_1 \\ B \end{bmatrix} \longrightarrow \begin{bmatrix} F \\ (\alpha_N)_1 \\ B \end{bmatrix}$$

Рис.36. Изменение выходного вектора программы Pupil. Линейную величину меняем на угловую

Результаты моделирования представлены на фрагменте (рис.37)

$$A(\tau) := \frac{\mathrm{Pupil}(\mathrm{Syst1}(\tau))_1 - \mathrm{Pupil}(\mathrm{Syst1}(0))_1}{\tau} \cdot \frac{1}{\mathrm{deg}} \qquad i := 0..10 \qquad t_i := (i+1) \cdot 0.001$$

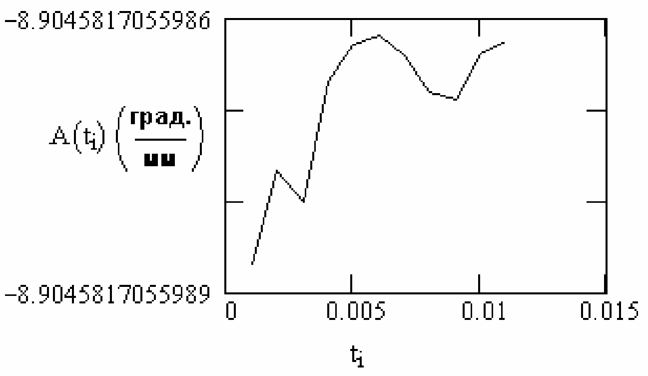


Рис.37. Фрагмент моделирования передаточной функции торцевого биения

Передаточная функция определена с помощью функции Pupil. С помощью ранжированной переменной задано 11 значений первичной погрешности и проведены вычисления непосредственно при построении графика функции. В пределах от 1 до 11 мкм передаточная функция постоянна и равна *A*=- 8,9 град/ мм.

8. Расчёт световых диаметров (габаритный расчёт)

Располагая информацией о положении и размерах входного зрачка можно оценить размеры всех диафрагм, обеспечивающих при заданном полевом угле три состояния системы:

1) отсутствие виньетирования;
2) виньетирование примерно 50%;
3) минимальный размер диафрагмы.

Для оценки можно полагать, что сечение светового пучка имеет круглую форму. На рис.38 отображены эти три ситуации:

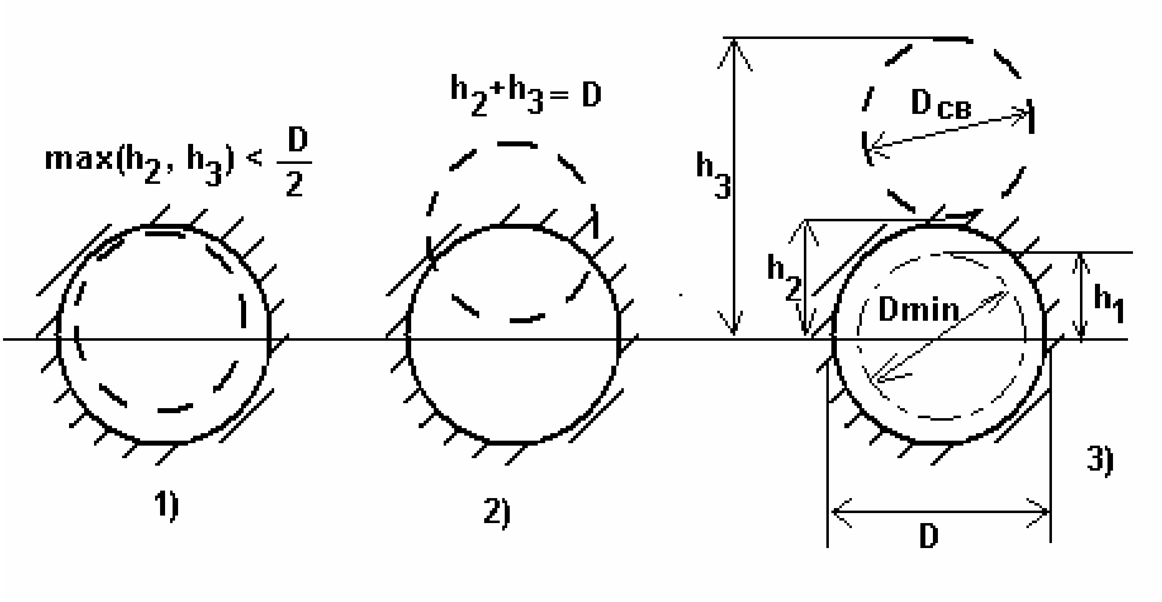


Рис.38. Взаимное расположение диафрагмы D и светового пучка Dсв

В этом случае кроме двух лучей, осевого и апертурного, необходимо прогонять ещё два луча, идущие на противоположные края входного зрачка. При расчётах световых высот необходимо оценивать положение двух полевых лучей и одного апертурного и определять размеры диафрагм виньетирования. Минимальный размер диафрагмы определяется по апертурному лучу. На рис.39 изображены три луча, по которым рассчитывается размер диафрагмы 0 – апертурный, 1, 2 – габаритные.

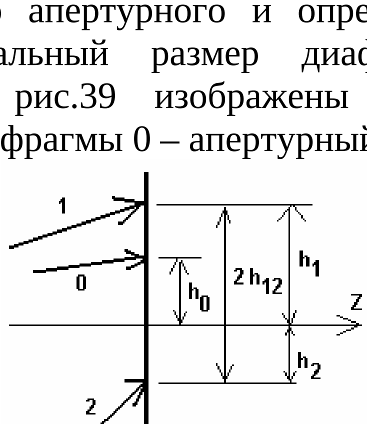


Рис.39. К определению размеров диафрагм

Отсутствие виньетирования происходит, если

$$D_k = 2\max(|h_1|,|h_2|). \qquad (38)$$

Виньетирование 50%, воспользуемся программой *Dv* (рис.13)

$$D_k = 2 \cdot Dv(0.5, h_1, h_2)\,. \qquad (39)$$

Минимальный размер диафрагмы:

$$D_k = 2 \cdot \left|h_0\right|, \qquad (40)$$

где k – номер поверхности.

Какой знак присваивать полевому углу? Согласно правилу знаков: луч, пересекающий оптическую ось сверху вниз, составляет с осью положительный угол, а пересекающий её снизу вверх – отрицательный. При расчётах в программе знак угла определяется из углового аналога инварианта Аббе. Проверим, отвечает ли используемая формула данному правилу. Формула имеет вид:

$$\alpha_{k+1} = \frac{1}{n_k}(n_{k-1}\alpha_k + h_k\Phi_k)\,. \qquad (41)$$

Если на поверхность падает луч, параллельный оптической оси, тогда α_k=0 и знак угла преломлённого луча определяется произведением $h_k\Phi_k/n_k$. Рассмотрим рис.43, иллюстрирующий преломление на собирающей ($\Phi_к$>0) и рассеивающих поверхностях ($\Phi_к$<0) при h_k>0 и n_k>0 и n_k<0. Как видно из рисунка 40, правило знаков в угловом аналоге инварианта Аббе выполняется.

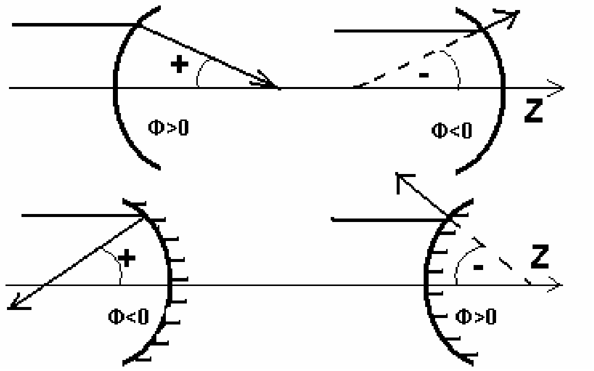


Рис.40. К определению знака угла луча с оптической осью

Таким образом, чтобы вычислять световые размеры необходимо обеспечить параллельный прогон четырёх лучей: один луч – для вычисления фокусного расстояния, один - апертурный и два луча – полевых.

Оптическая система с осевой симметрией может быть так ориентирована относительно внеосевой точки предмета, что меридиональная плоскость, в которой находится точка предмета, совпадёт с одной из координатных плоскостей, например, *OYZ* и габаритный расчёт в этих условиях провидится в двух измерениях, в меридиональной плоскости *OYZ*.

Для габаритного расчёта необходимо проследить ход через систему четырёх лучей (рис.38). Оптическая система на рисунке представлена передней главной плоскостью первой поверхности и входным зрачком. Главная плоскость перпендикулярна оптической оси и проходит через полюс поверхности. Входной зрачок задан двумя параметрами: алгебраическим отрезком $z_{\text{ВХ}}$ относительно полюса первой поверхности и диаметром $D_{\text{ВХ}}$.

Точка предмета определена двумя параметрами: алгебраической величиной переднего отрезка z_0, определяющего положение плоскости предмета, и полевым углом β. Знаки частных значений данных величин указаны на рис. 41.

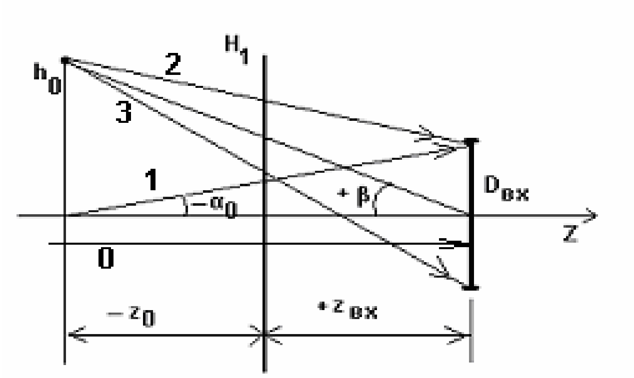


Рис.41. Входные лучи для габаритного расчёта оптической системы

Луч 0 параллелен оптической оси и имеет произвольную высоту падения кроме нулевой. Он используется для вычисления фокусного расстояния системы.

Луч 1 – апертурный луч стартует в осевой точке плоскости предмета и следует на край входного зрачка. В системе он определяет минимально возможные размеры диафрагм. Если какая-либо диафрагма окажется меньше, то она будет играть роль апертурной диафрагмы. Апертурный угол системы уменьшится и изменится положение входного зрачка. Чтобы это не произошло, все диафрагмы должны быть больше или равными двойной высоте апертурного луча.

Лучи 2 и 3 полевые углы, идущие на противоположные края входного зрачка, определяют размеры диафрагм, обеспечивающих заданный режим виньетирования: 0% или 50%. Если система предназначена для зрительного восприятия изображения, то, как показывает практика, виньетирование в 50% для глаза незаметно.

Если же заданы размеры диафрагм, то программа должна определить фактическое безаберационное виньетирование.

Системная матрица с добавлением параметров входного зрачка, полевого угла и диаметров реальных диафрагм может иметь, например, вид

$$Syst = \begin{pmatrix} R_0 & R_1 \dots \infty & R_{k+1} \dots R_{N-1} & D_{BX} & \beta \\ t_0 & t_1 \dots t_{k-1} & t_k \dots t_{N-1} & z_0 & z_{BX} \\ n_0 & n_1 \dots 1 & 1 \dots n_{N-1} & n_N & k \\ D_0 & D_1 \dots D_{k-1} & D_k \dots D_{N-1} & V & 0 \end{pmatrix}. \qquad (42)$$

Диаметры реальных диафрагм помещены в четвёртую строку.

Предпоследний столбец: $D_{\text{ВХ}}$ – диаметр входного зрачка, z_0 – передний отрезок (алгебраическое расстояние от осевой точки первой поверхности до плоскости предмета), n_N – показатель преломления пространства изображения, признак V, например, если $V = 0$, то строка перечисленных диаметров не учитывается и происходит вычисление размеров диафрагм, обеспечивающих три стандартные режима виньетирования, если же V = 1, то происходит вычисление коэффициента виньетирования на заданной диафрагме.

Последний столбец: β – полевой угол (рад.), $z_{\text{ВХ}}$- алгебраическое расстояние до входного зрачка от осевой точки первой поверхности, k – номер столбца, начиная с нулевого, в котором размещается информация об апертурной диафрагме (если известно положение входного зрачка, то этот элемент не используется), правое нижнее место матрицы – резервное.

Блок-схема программы изображена на рис. 42, а текст программы помещён на рис. 43.

Комментарии к программе.

1)Извлечение информации о радиусах, толщинах, показателях преломления, числе поверхностей и положении плоскости предмета производится по уже опробованной схеме с использованием функций и операторов матричных преобразований.

2)Радиус входного зрачка рассчитывается по величине диаметра, извлекаемого из заданного места системной матрицы, так же как и положение входного зрачка.

3)Полевой угол задаётся в радианах или градусах. В последнем случае нужно величину в градусах помножить на размерную величину deg, например, задать полевой угол в виде $5 \cdot \text{deg}$.

Извлечение информации о реальных размерах диафрагм и оправ линз происходит независимо от признака V, например, $D \leftarrow \left((Syst)^T\right)^{\langle 3 \rangle}$.

1. Вычисление тангенса апертурного угла происходит по формуле

$$\gamma = \text{tg}(\alpha_0) = \begin{cases} \dfrac{D_{\text{ВХ}}}{2(z_0 - z_{\text{ВХ}})}, & z_0 \neq \infty,\ z_{\text{ВХ}} \neq \infty \\ 0, & \text{в противном случае} \end{cases} . \qquad (43)$$

2. Задание 4-х мерного вектора тангенсов входных углов происходит следующим образом:

$$\alpha_0 = \begin{pmatrix} 0 \\ \gamma \\ \text{tg}(\beta) + \gamma \\ \text{tg}(\beta) - \gamma \end{pmatrix}, \qquad (44)$$

вычисляемый индекс 0 указывает на номер поверхности.

3. Вычисление 4-х мерного вектора высот на входной поверхности по формуле: $$h_0 = \frac{D_{\text{ВХ}}}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix} - \begin{cases} z_{\text{ВХ}} \alpha_0, & \text{если } z_{\text{ВХ}} \neq \infty \\ 0, & \text{в противном случае} \end{cases}, \qquad (45)$$

здесь осуществляется вычитание двух векторов, заданных соответственно явно и неявно.

4. Программа-функция Vign предназначена для вычисления по ключу – признак V, коэффициента виньетирования или характерных размеров диафрагмы или оправы.

Входными параметрами являются: вектор высот h, признак V и диаметр диафрагмы.

Обращение к программе может быть следующее: $H_0 \leftarrow Vign(h_0, V, D)$. В зависимости от признака определяемая присвоением величина H_0 может быть или скаляром (V=1), или 3-х мерным вектором характерных диаметров.

5. Задание начального значения счётчика оптической силы и начального кардинального отрезка происходит аналогично заданию в программе вычисления положения входного зрачка с условным названием Pupil.

6. В цикле по поверхностям необходимо создать шаблон 4-х-мерного вектора задних отрезков, соответствующих вектору углов трассируемых через систему лучей в силу специфики вычисления задних отрезков. Такими векторами являются h_0 и α_0, поэтому текущему вектору задних отрезков приемлемо выполнить следующее присвоение $z_k \leftarrow h_0$, где k – индекс номера поверхности.

7. Вычисление оптической силы выполняется аналогично вычислению в программе Pupil.

8. Внешне вычисление углов преломления не отличается от ранее применявшегося в программе Pupil. Но в данном случае осуществляется векторное вычисление всех четырёх значений параллельно.

9. Вычисление задних отрезков происходит в результате поэлементного деления вектора высот на вектор тангенсов углов. Такая операция в MathCAD выполняется компактно с помощью введения оператора поэлементных вычислений векторов. Эта операция обозначается в виде стрелки над операцией. Например, даны два вектора $a = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$. Их произведение $a \cdot b = 2 \cdot 1 + 8 \cdot 4 = 34$ понимается как скалярное произведение, если над ними не стоит стрелка. Если же оператор поэлементного вычисления в виде стрелки над операндами указан, то имеем

$$\overrightarrow{(a \cdot b)} = \begin{pmatrix} 2 \\ 32 \end{pmatrix} \quad \overrightarrow{\frac{a}{b}} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

В данном случае необходимо проверять наличие нулей в элементах вектора тангенсов углов, поэтому приходится поэлементное деление организовывать в цикле. Здесь как раз и используется шаблон вектора задних отрезков. Вычисляемым оператором в теле цикла i = 0…3 может быть единственный логический оператор, построенный с помощью логической функции *if*, например,

$$\left(z_k\right)_i \leftarrow \text{if}\left[\left(\alpha_{k+1}\right)_i \neq 0, \frac{\left(h_k\right)_i}{\left(\alpha_{k+1}\right)_i}, \infty\right]$$
.

10.Вычисление высот лучей на следующей поверхности происходит внешне по прежним формулам (программа Pupil), но в векторном исполнении. Так как для последней поверхности последующей не существует, то необходимо ввести соответствующее ограничивающее условие: $if \quad k \neq N-1$, где k – параметр цикла, а N – число поверхностей. Здесь используется значение системной переменной по умолчанию ORIGIN=0, задающее номер первого элемента вектора. Для изменения данной переменной нужно обратиться к пункту меню “Математика” и выбрать в ниспадающем меню последнюю кнопку “Опции”. Далее провести соответствующую замену и нажать клавишу “OK”.

11.Обращение к программе Vign с текущими значениями высот.

12.Вычисление оптической силы не отличается от ранее применявшегося в программе Pupil.

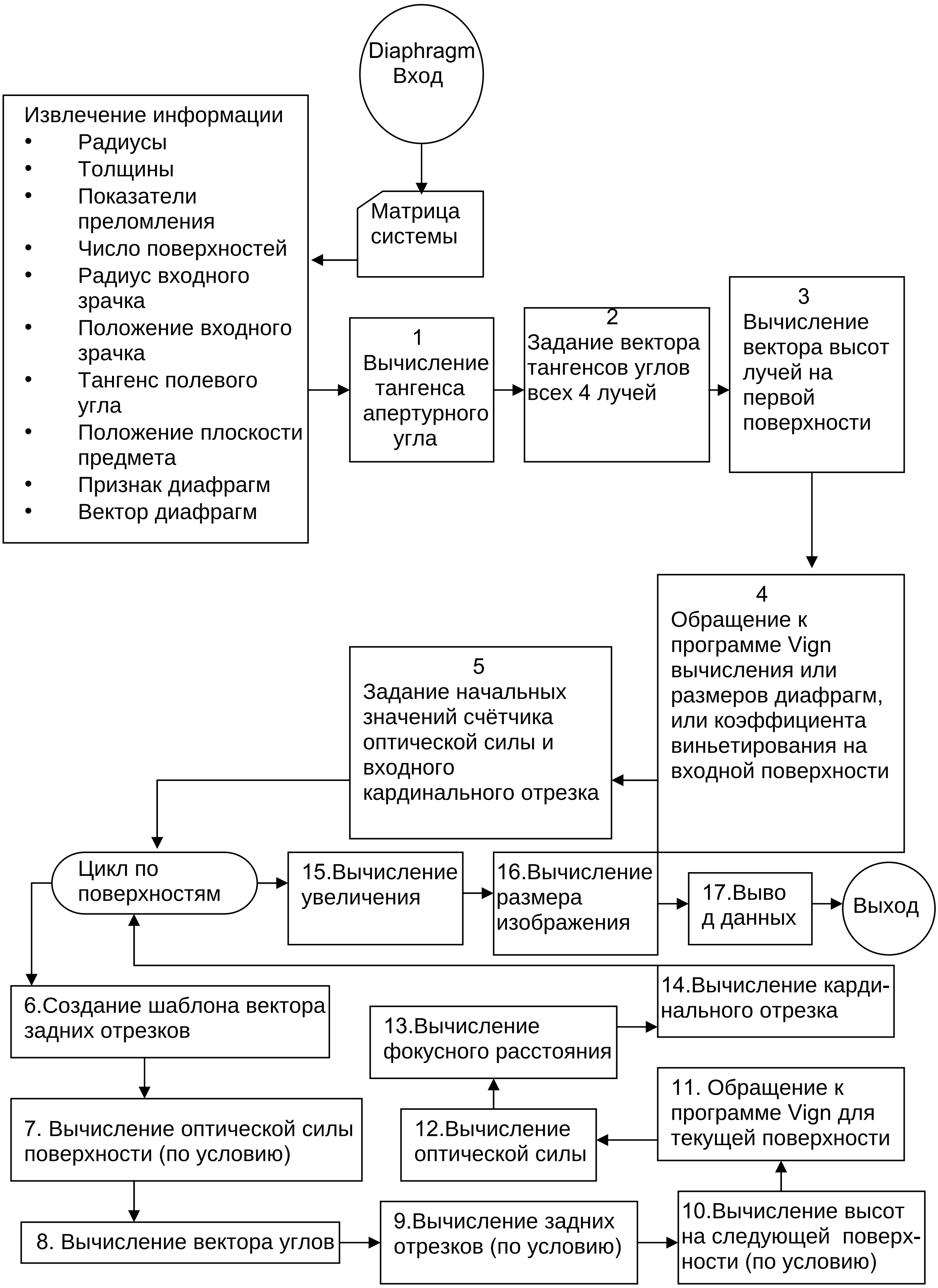


Рис. 42.Блок-схема программы габаритного расчёта осесимметричной оптической системы

$Diaphragm(Syst) :=$

$r \leftarrow \left((Syst)^T\right)^{\langle 0 \rangle}$ **Извлечение информации из матрицы Syst**
радиусы

$t \leftarrow \left((Syst)^T\right)^{\langle 1 \rangle}$ толщины

$n \leftarrow \left((Syst)^T\right)^{\langle 2 \rangle}$ показатели преломления

$D \leftarrow \left((Syst)^T\right)^{\langle 3 \rangle}$ диаметры диафрагм

$N \leftarrow cols(Syst) - 2$ число поверхностей

$v \leftarrow Syst_{3,N}$ признак (v=0 вычисление D_0, $D_{1/2}$, D_{min} v=1 вычисление коэффициентов виньетирования)

$R_v \leftarrow \frac{r_N}{2}$ радиус входного зрачка

$\beta \leftarrow \tan(r_{N+1})$ полевой угол (рад)

$z_v \leftarrow t_{N+1}$ расстояние от осевой точки первой поверхности до входного зрачка

$z0 \leftarrow t_N$ расстояние до плоскости предмета

$\gamma \leftarrow if\left(z0 \neq \infty \wedge z_v \neq \infty, \frac{R_v}{z0 - z_v}, 0\right)$ вычисление тангенса переднего апертурного угла

$\alpha_0 \leftarrow (0 \;\; \gamma \;\; \beta+\gamma \;\; \beta-\gamma)^T$ вектор тангенсов углов входных лучей

$h_0 \leftarrow R_v \cdot (1 \;\; 1 \;\; 1 \;\; -1)^T + if\left(z_v \neq 0, z_v \cdot \alpha_0, 0\right)$ вектор высот лучей

$di_0 \leftarrow Vign(h_0, v, D_0)$ Обращение к програме Vign

$\Phi \leftarrow 0$ Счётчик оптической силы

$d_0 \leftarrow 0$ Задний кардинальный отрезок первой поверхности

$for \;\; k \in 0..N-1$ заголовок цикла по числу поверхностей

- $z_k \leftarrow h_0$ устанавливаем шаблон вектора отрезков по образцу вектора высот
- $\phi_k \leftarrow if\left(r_k = \infty, 0, \frac{n_{k+1} - n_k}{r_k}\right)$ оптическая сила к-поверхности
- $\alpha_{k+1} \leftarrow \frac{1}{n_{k+1}} \cdot \left(n_k \cdot \alpha_k + h_k \cdot \phi_k\right)$ вычисление вектора преломленных лучей
- $for \;\; i \in 0..3$ цикл по элементам вектора задних отрезков
  - $\left(z_k\right)_i \leftarrow if\left[\left(\alpha_{k+1}\right)_i \neq 0, \frac{\left(h_k\right)_i}{\left(\alpha_{k+1}\right)_i}, \infty\right]$ вычисление к-того заднего отрезка

Рис.43. Начало программы Diaphragm

$\text{if } k \neq N-1$ так как для последней поверхности нет последующей

$h_{k+1} \leftarrow h_k - \alpha_{k+1} \cdot t_{k+1}$ вычисление высот на последующей поверхности

$di_{k+1} \leftarrow Vign\left(h_{k+1}, v, D_{k+1}\right)$ обращение к программе Vign

$\Phi \leftarrow \Phi + \phi_k - \dfrac{t_k - d_k}{n_k} \cdot \Phi \cdot \phi_k$ вычисление оптической силы текущей части системы

$f_k \leftarrow \text{if}\left(\Phi \neq 0, \dfrac{n_{k+1}}{\Phi}, \infty\right)$ текщее фокусное расстояние

$d_{k+1} \leftarrow \text{if}\left[\Phi \neq 0, f_k - \left(z_k\right)_0, 0\right]$ задний кардинальный отрезок текущей части оптической системы

цикл завершён

$\upsilon \leftarrow \dfrac{\left(z_{N-1}\right)_0 - \left(z_{N-1}\right)_1}{f_{N-1}}$ линейное увеличение системы

$Y' \leftarrow \text{if}\left(z0 \neq \infty, z0 \cdot \gamma \cdot \upsilon, 0\right)$ линейный размер предмета

$\left[f_{N-1} \;\; \left(z_{N-1}\right)_1 \;\; Y' \;\; di\right]^T$ Вектор выводимых параметров

Рис.43а. Окончание программы Diaphragm

13.Вычисление текущего значения фокусного расстояния происходит по условию отличия от нуля знаменателя. При этом используется показатель преломления, расположенный справа от поверхности, поэтому оператор присваивания имеет вид $f_k \leftarrow \text{if}\left(\Phi \neq 0, \dfrac{n_{k+1}}{\Phi}, \infty\right)$.

14.Вычисление кардинального отрезка происходит по той же формуле, что и раннее, но из вектора задних отрезков используется значение, соответствующее лучу, входящему в систему параллельно оптической оси $d_{k+1} \leftarrow \text{if}\left[\Phi \neq 0, f_k - \left(z_k\right)_0, 0\right]$, при этом по условию отделяется случай бесконечного фокусного расстояния. Заметим, что d_N последний вычисленный кардинальный задний отрезок является таковым для всей системы.

15.Цикл закончился. По рассчитанным данным определяем линейное увеличение системы. По формуле Ньютона оно равно $v = \dfrac{-x'}{f'}$, где x' – отрезок, отсчитываемый от фокуса до изображения. В данном случае вычисление организуется как:

$$v \leftarrow -\frac{\left(z_{N-1}\right)_0 - \left(z_{N-1}\right)_1}{f_{N-1}}$$

.

16. Вычисление линейного размера изображения происходит через вычисление линейного размера предмета по расстоянию до плоскости предмета z_0 и тангенсу полевого угла и линейного увеличения системы. Ограничивающим условием является условие бесконечности переднего отрезка $Y' \leftarrow \mathrm{if}(z0 \neq \infty, z0 \cdot \gamma \cdot v, 0)$.

17. В выводимые данные включаем:

фокусное расстояние системы $f_{N\text{-}1}$;

положение изображения относительно последней поверхности $(z_{N\text{-}1})_1$;

линейный размер изображения $Y`$;

вектор виньетирования на поверхностях, или матрицу характерных диаметров.

8.1. Описание программы Vign

А) Если признак V=0, то определяются характерные диаметры оправ и диафрагм по вектору высот на поверхности. В первую строку вектора диафрагм помещаем размер диафрагмы, свободной от виньетирования:

$$D_0 = 2\max(|h_2|,|h_3|) \qquad (46)$$

во вторую размер диафрагмы с виньетированием 50%:

$$D_1 = Dv(0.5, h_1, h_2) \qquad (47)$$

и в третью строку минимальный размер диафрагмы, рассчитываемый по апертурному лучу:

$$D_2 = 2\cdot|h_1|. \qquad (48)$$

Б) Если признак V = 1, то заданы размеры оправ и диафрагм. Виньетирование вычисляется как отношение площадей:

$$\eta = \frac{S_O}{\pi\rho^2}. \qquad (49)$$

Площади перекрытия диафрагмы и светового пучка S_0 площади сечения светового пучка.

Для вывода формул перекрытия двух кругов расположим больший круг с центром в начале координат, а малый над ним. Возможны три случая пересечения (рис.44) и один случай: малый круг полностью заключён внутри большого.

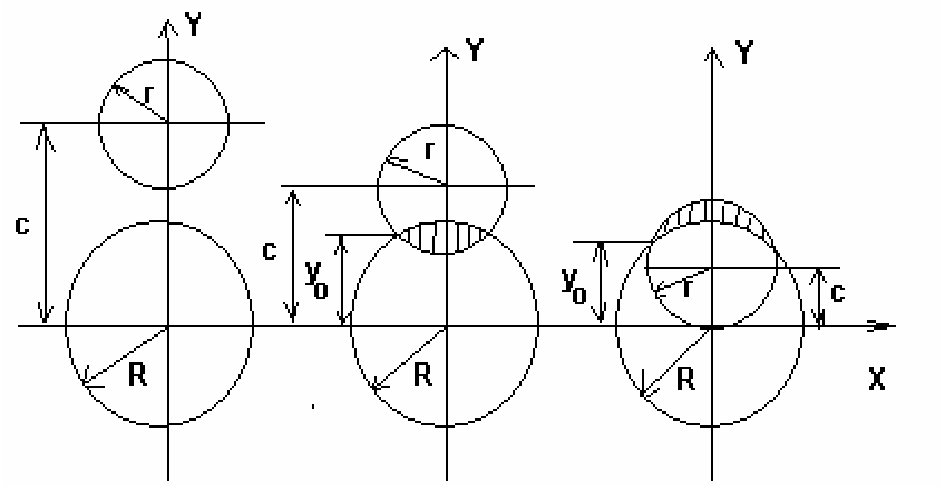


Рис.44. Возможные случаи взаимного расположения диафрагм

1) $c>r+R$ – S_o=0, круги не пересекаются, общая площадь – нулевая.

2) $c> y_0$ – расстояние между центрами кругов больше ординаты точек пересечения окружностей, S_o>0,

3) $c< y_0$ – в этом случае удобнее вычислять внешнюю часть S_v и $S_o = \pi r^2 - S_v$.

Если обозначить через x_0 абсциссу точек пересечения окружностей, то:

$$S_0(c > y_0) = 2\int_0^{x0}\left[\sqrt{R^2 - x^2} - \left(c - \sqrt{r^2 - x^2}\right)\right]dx = I(R, x_0) + I(r, x_0) - 2cx_0$$

$$S_0(c \le y_0) = \pi r^2 - 2\int_0^{x0}\left[\left(c + \sqrt{r^2 - x^2}\right) - \sqrt{R^2 - x^2}\right]dx = \pi r^2 + I(R, x_0) - I(r, x_0) - 2cx_0, \quad (50)$$

где $I(A, x) = 2\int_0^{x}\sqrt{A^2 - t^2}\,dt = x\sqrt{A^2 - x^2} + A^2 \arcsin(x / A)$

Координаты точек пересечения окружностей легко находятся из системы двух уравнений:

$$\begin{matrix} x^2 + y^2 = R^2 \\ x^2 + (y - c)^2 = r^2 \end{matrix} \Rightarrow \quad y_0 = \frac{R^2 + c^2 - r}{2c}, \quad x_0 = \sqrt{R^2 - y_0^2} \qquad (51)$$

Далее радиус светового пучка определяется высотами второго и третьего лучей: $r1 = 0{,}5\left|h_3 - h_2\right|$.

Высота центра светового пучка $c = \left|h_2 + h_3\right|$.

Находим больший и меньший радиусы: R=max(r1,D/2), r = min (r1, D/2), где D – диаметр диафрагмы.

Итак, получены все формулы для составления программы вычисления коэффициента виньетирования.

Ниже приведена блок-схема программы.

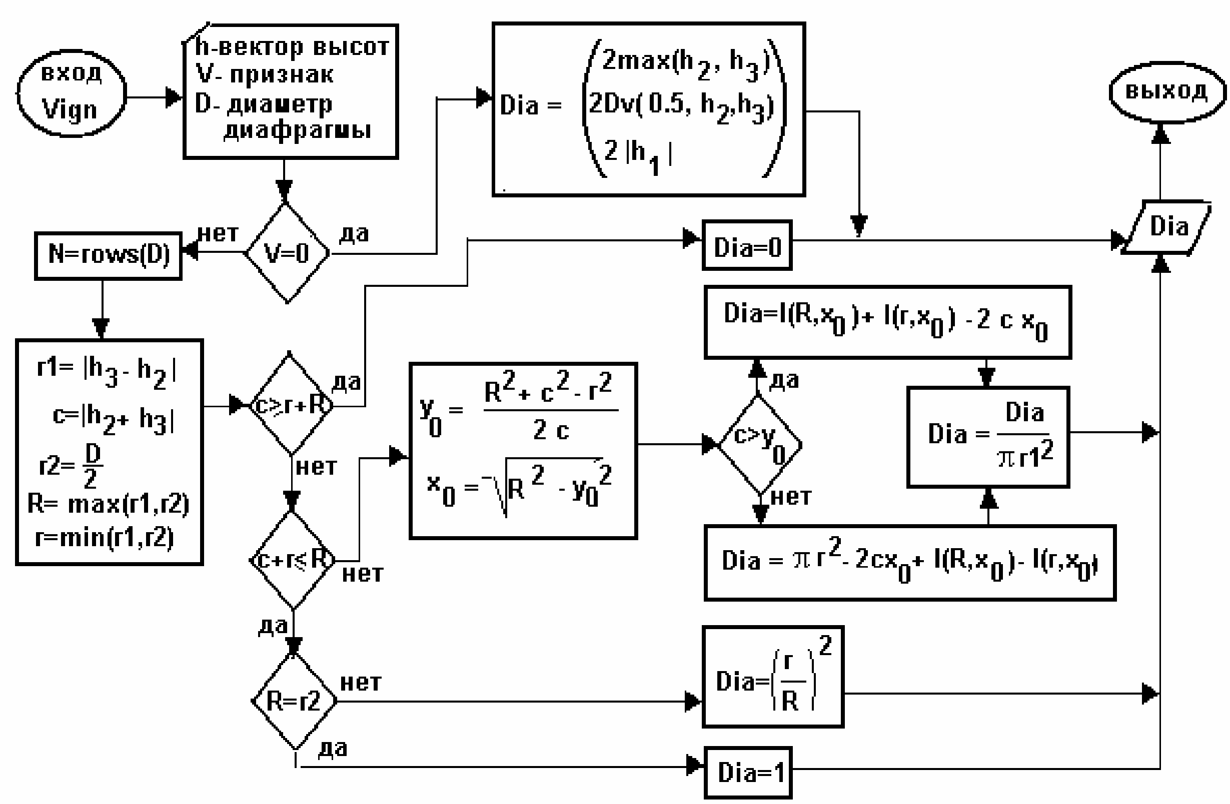


Рис.45. Блок-схема программы виньетирования

Вариант текста программы приведён ниже (рис.47).

$$f(t) := Vign\left[\begin{pmatrix} 5 \\ 0 \\ t \\ t+4 \end{pmatrix}, 1, 10\right] \qquad i := 0..20 \qquad y_i := 5 - \frac{i}{5} \qquad K_{vin_i} := f(y_i)$$

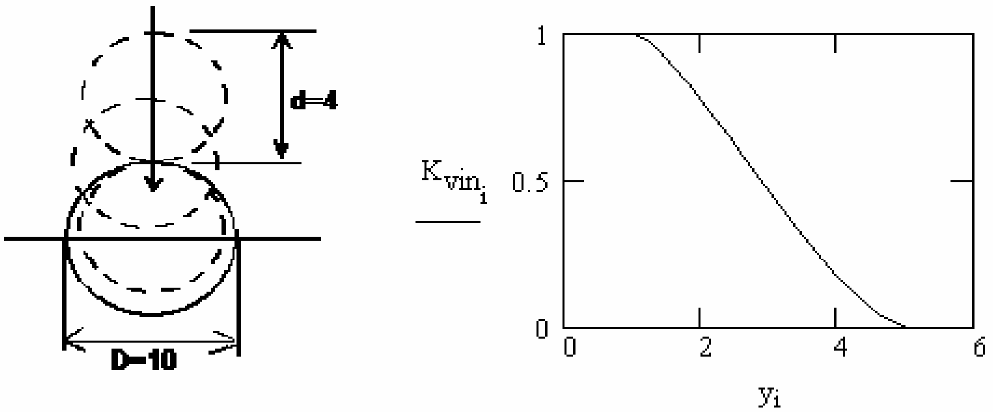


Рис.46. Фрагмент документа MathCAD,
иллюстрирующий тестирование программы виньетирования Vign

```
Vign(h,V,D) := | Dia ← ( 2·max(h2,h3)      ) if V = 0
               |        ( 2·Dv(0.5,h2,h3)   )
               |        ( 2·|h1|            )
               | otherwise
               |   | r1 ← 0.5·|h3 − h2|
               |   | c ← 0.5·|h2 + h3|
               |   | r2 ← 0.5·D
               |   | R ← max(r1,r2)
               |   | r ← min(r1,r2)
               |   | return 0 if c ≥ r + R
               |   | if c + r ≤ R
               |   |   | return 1 if R = r2
               |   |   | (r/R)^2 otherwise
               |   | otherwise
               |   |   | y0 ← (R^2 + c^2 − r^2)/(2·c)
               |   |   | x0 ← √(R^2 − y0^2)
               |   |   | return (I(R,x0) + I(r,x0) − 2·c·x0)/(π·r1^2) if c > y0
               |   |   | return (π·r^2 − 2·c·x0 + I(R,x0) − I(r,x0))/(π·r1^2) otherwise
```

Рис.47. Текст программы виньетирования Vign

8.2. Плосковыпуклая линза (пример)

В качестве примера рассмотрим простую систему, состоящую из плосковыпуклой линзы и апертурной диафрагмы, установленной за линзой (рис.48).

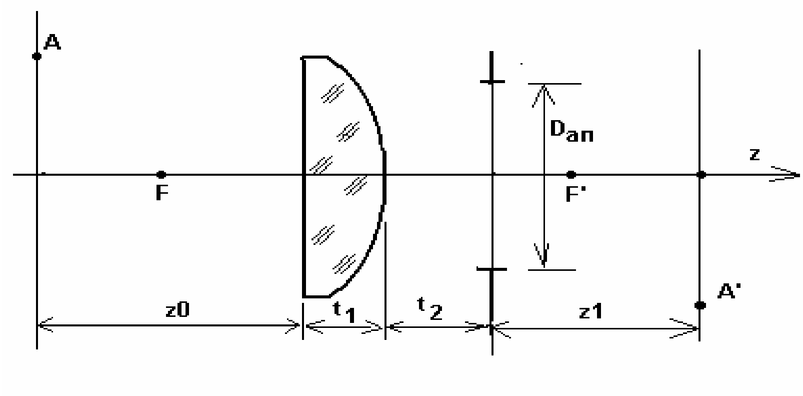


Рис.48.Пример оптической системы.

$R_1=\infty$, $R_2=-100$, $t_1=10$, $t_2=15$, $D_{ап}=20$ мм, $n=1{,}5$

Для расчёта положения и размеров входного зрачка используем следующую системную матрицу:

$$Syst := \begin{pmatrix} 100 & \infty & 20 \\ 0 & 10 & -15 \\ 1 & 1.5 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Обращаясь к программе Pupil, находим искомые параметры:

$$Pupil(Syst) = \begin{pmatrix} 200 \\ -22.883 \\ 21.622 \end{pmatrix} \begin{matrix} \text{фокусное расстояние} \\ \text{задний отрезок (расстояние до входного зрачка)} \\ \text{диаметр входного зрачка} \end{matrix}$$

Теперь оптическая система имеет вид (рис.49).

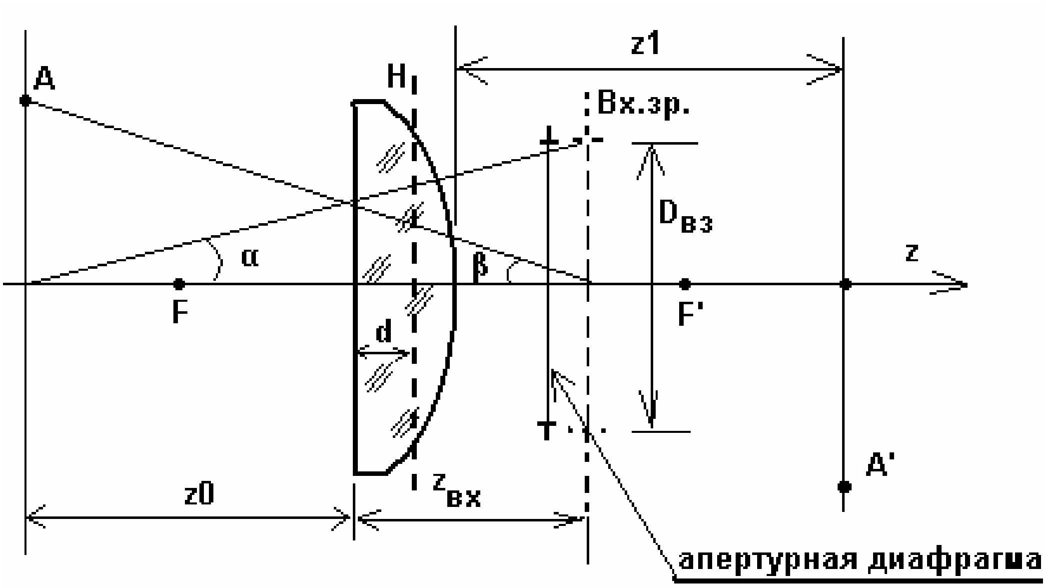


Рис.49. Оптическая система с входным зрачком

Определим размер диафрагм, обеспечивающих режимы виньетирования 0 и 50% для точки объекта А, расположенном на двойном фокусном расстоянии. Полевой угол $\beta = 45^0$.

Системная матрица имеет вид:

$$\mathrm{Syst} := \begin{pmatrix} \infty & -100 & \infty & 21.622 & 45\cdot \mathrm{deg} \\ 0 & 10 & 15 & -393.333 & 22.883 \\ 1 & 1.5 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Передний отрезок $z_0 = -(2f'-d)$. Для определения переднего кардинального отрезка обратимся к программе Pupil, составив системную матрицу Syst0. Фрагмент документа MathCAD:

$$\mathrm{Syst0} := \begin{pmatrix} 100 & \infty & 20 \\ 0 & 10 & \infty \\ 1 & 1.5 & 1 \\ 0 & 0 & 0 \end{pmatrix} \qquad B := \mathrm{Pupil}(\mathrm{Syst0}) \qquad B = \begin{pmatrix} 200 \\ 193.333 \\ 0 \end{pmatrix}$$

$$d := B_0 - B_1 \qquad d = 6.667$$

Вычисление размеров диафрагм с использованием программы Diaphragm имеет вид

$$s := \mathrm{Diaphragm}(Syst) \qquad s = \begin{pmatrix} 200 \\ 385 \\ -1 \\ \{3,1\} \end{pmatrix} \begin{matrix} \text{фокусное расстояние} \\ \text{Последний отрезок (до апертурной диафрагмы)} \\ \text{линейное увеличение} \\ \text{вектор диафрагм} \end{matrix}$$

поверхность 1	поверхность 2	апертурная диафрагма	Виньетирование
$(s3)_0 = \begin{pmatrix} 66.199 \\ 45.766 \\ 20.433 \end{pmatrix}$	$(s3)_1 = \begin{pmatrix} 53.212 \\ 32.433 \\ 20.78 \end{pmatrix}$	$(s3)_2 = \begin{pmatrix} 20.001 \\ 20 \\ 20 \end{pmatrix}$	0% 50% Dmin

Выберем размер диафрагм, обеспечивающий 50%-виньетирование и составим системную матрицу для программы Diaphragm в виде

$$Syst := \begin{pmatrix} \infty & -100 & \infty & 21.622 & 45\cdot deg \\ 0 & 10 & 15 & -393.333 & 22.883 \\ 1 & 1.5 & 1 & 1 & 2 \\ 45.766 & 32.433 & 20 & 1 & 0 \end{pmatrix}$$

Обратившись к программе Diaphragm с признаком v =1, получаем действительные коэффициенты виньетирования:

$$s := \mathrm{Diaphragm}(Syst) \qquad s = \begin{pmatrix} 200 \\ 385 \\ -1 \\ \{3,1\} \end{pmatrix}$$

Коэффициенты виньетирования

$$(s3)_0 = 0.452 \qquad (s3)_1 = 0.431$$

Действительные коэффициенты виньетирования отличаются от значения 50% в меньшую сторону.

9. Моделирование центрированной оптической системы с зеркальными элементами.

При описании системы с зеркальными элементами в общем случае необходимо использовать трёхмерное пространство.

Положение луча в трёхмерном пространстве описывается тремя пространственными координатами произвольной точки (x, y, z) на луче и направляющими косинусами (p, q, m).

$$p = \cos(\alpha), \quad q = \cos(\beta), \quad m = \cos(\gamma), \qquad (52)$$

где α, β и γ – углы между направлением луча и осями координат, соответственно *OX*, *OY* и *OZ*, *n* – показатель преломления среды.

Между направляющими косинусами справедливо соотношение:

$$p^2 + q^2 + m^2 = 1. \qquad (53)$$

С другой стороны инвариант Аббе, лежащий в основе габаритных вычислений, в угловом виде:

$$\gamma \cdot n' - \gamma \cdot n = h \cdot \varphi. \qquad (54)$$

имеет двумерное содержание. В этом выражении используется тангенс угла между лучом и осью *OZ* и высота падения луча на главную плоскость.

Угловое пространственное положение точки предмета определяется двумя углами: полевым углом β и азимутальным углом ψ (рис.50).

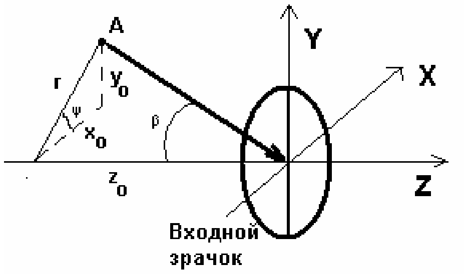


Рис.50. К определению полевого и азимутального углов

Полевой угол определяется высотой положения точки предмета *r* и расстоянием от входного зрачка до плоскости предмета z_0:

$$\beta = \operatorname{arctg} \frac{r}{z_0} = \operatorname{arctg} \frac{\sqrt{x_0^2 + y_0^2}}{z_0}. \qquad (55)$$

Азимутальный угол определяется положением точки предмета в плоскости предмета:

$$\psi = \operatorname{arctg} \frac{y_0}{x_0}. \qquad (56)$$

Безаберационные рефракционные оптические элементы осесимметричной системы не влияют на азимутальный угол. Он остаётся неизменным при любом числе преломлений. Это обстоятельство позволило вести расчёт осесимметричных систем в двух измерениях, на плоскости.

Рассмотрим действие плоского зеркала О*B* (рис.51) на пространство предмета. Пусть проекция A_X точки *A* на плоскость О*YZ* расположена в меридиональной плоскости зеркала, плоскости, содержащей оптическую ось (ось О*Z*) и нормаль к зеркалу *N* (плоскость рисунка). Плоскость зеркала наклонена к оптической оси на угол -α. Оптическая ось в пространстве изображения испытывает излом на угол 2α. Для спрямления оптической оси пространство изображения необходимо повернуть на угол 2α около оси О*X* против часовой стрелки. В результате ордината точки изображения меняет знак на противоположный. При наличии отражательных элементов, как видим, азимутальный угол изображения (рис. 51) изменится скачкообразно.

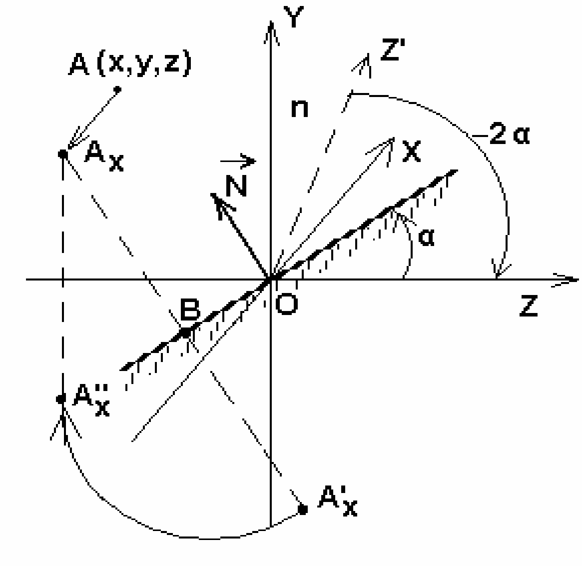


Рис.51. Развёртка плоского зеркала

Опишем данную ситуацию аналитически. Дано: координаты точки предмета (x_0, y_0, z_0) и направляющие косинусы зеркала (0, q, m). Определить: координаты точек A' и A'' и угол поворота.

Решение**.** Точка B находится посередине между точками A и A'.

$$B = \frac{A + A'}{2} \quad \rightarrow \quad A' = 2B - A. \qquad (57)$$

Направляющие косинусы нормали зеркала выразим через угол α.

$$q = n \cdot \cos(\alpha), \quad m = n \cdot \sin(\alpha). \qquad (58)$$

Из двух возможных направлений нормали к зеркалу выбираем такое, что q>0. Уравнение следа зеркала на плоскости O*YZ:*

$$y = -z \cdot \mathrm{tg}(\alpha) \quad \rightarrow \quad y \cdot \cos(\alpha) + z \cdot \sin(\alpha) = 0. \qquad (59)$$

Вектор AB, $A(y_0,\ z_0)$, $B(y_1,\ z_1)$, как перпендикуляр к плоскости коллинеарен нормали, поэтому:

$$\frac{\cos(\alpha)}{y_1 - y_0} = \frac{\sin(\alpha)}{z_1 - z_0} \quad \rightarrow \quad y_1 \sin(\alpha) - z_1 \cos(\alpha) = y_0 \sin(\alpha) - z_0 \cos(\alpha). \qquad (60)$$

Получили систему из двух уравнений (59), (60) относительно двух неизвестных y_1 и z_1, решение имеет вид:

$$\begin{matrix} y_1 \\ z_1 \end{matrix} = \begin{matrix} \sin(\alpha) \\ \cos(\alpha) \end{matrix} \left(-y_0 \sin(\alpha) + z_0 \cos(\alpha)\right). \qquad (61)$$

После несложных тригонометрических преобразований находим координаты изображения A':

$$\begin{matrix} y' = 2y_1 - y_0 = -y_0 \cos(2\alpha) - z_0 \sin(2\alpha) \\ z' = 2z_1 - z_0 = -y_0 \sin(2\alpha) + z_0 \cos(2\alpha) \end{matrix}. \qquad (62)$$

В матричной записи данное решение с учётом того, что $x'=x_0$ имеет вид:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = M_1 \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}, \quad где \quad M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\cos(2\alpha) & -\sin(2\alpha) \\ 0 & -\sin(2\alpha) & \cos(2\alpha) \end{pmatrix} \qquad (63)$$

Определитель матрицы равен -1. Таким образом, преобразование зеркала в данном случае есть преобразование поворота на угол 2α с инверсией.

Для выравнивая оптической оси требуется поворот вокруг оси *OX* на угол -2α.

Поворот считается положительным, если осуществляется в правой системе координат против часовой стрелки, если смотреть с конца оси вращения. На рисунке 52 показаны положительные повороты вокруг всех трёх осей правой системы координат.

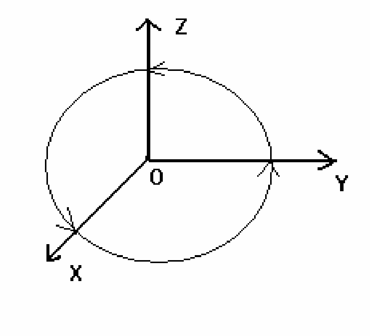


Рис.52. Положительные направления в правой системе координат

При преобразованиях поворота справедливы следующие мнемонические правила:

1) одноимённые координаты имеют перед собой множитель косинус, разноимённые – синус.

2) координаты оси, от которой отсчитывается угол, записываются с положительным углом, поэтому справедлива такая запись преобразования поворота: $y' = y \cdot \cos(\alpha) + z \cdot \sin(\alpha)$,

3) а координата оси, к которой направлен положительный поворот, записывается с отрицательным углом, и справедлива такая запись: $z' = z \cdot \cos(\alpha) - y \cdot \sin(\alpha)$.

Таким образом, поворот относительно оси *OX* на угол -2α осуществляется с помощью матрицы M_0:

$$M_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(2\alpha) & -\sin(2\alpha) \\ 0 & \sin(2\alpha) & \cos(2\alpha) \end{pmatrix}. \qquad (64)$$

Определитель такой матрицы равен 1. Матрица M_0 описывает операцию устранения излома оптической оси.

Совместное действие двух матриц даёт изображение *A’’*:

$$A''=M_0A'=M_0M_1A, \quad M=M_0M_1=\begin{pmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{pmatrix}. \qquad (65)$$

Таким образом, в данном случае, когда плоское зеркало параллельно оси *OX*, независимо от величины угла наклона зеркала к оптической оси происходит инверсное преобразование одной лишь ординаты, тогда как абсцисса и аппликата точки объекта не изменяются.

Рассмотрим общий случай. Пусть нормаль к зеркалу имеет направляющие косинусы (*p*, *q*, *m*), а точка предмета координаты (x_0, y_0, z_0). Зеркало пересекает оптическую ось в точке с аппликатой *Z*. Требуется определить координаты точки изображения после развёртки зеркала до выпрямления оптической оси.

План решения данной задачи следующий.

1) Повернём систему координат, так чтобы ось *OX* была параллельна зеркалу, в результате меридиональная плоскость зеркала окажется совмещённой с координатной плоскостью *OYZ*. Мы получим ситуацию, рассмотренную ранее.
2) Получим изображение точки предмета в частном случае положения системы координат, применив преобразование (65).
3) Осуществим обратный поворот к повороту в пункте 1.

Чтобы определить угол поворота системы до совмещения меридиональной плоскости зеркала с плоскостью *OYZ* рассмотрим положение нормали к зеркалу в пространстве: рис.53.

Из рисунка следует, что

$$\sin(\varphi)=\frac{p}{\sqrt{p^2+q^2}}, \quad \cos(\varphi)=\frac{q}{\sqrt{p^2+q^2}}. \qquad (66)$$

Осуществляется положительный поворот относительно оси *OZ* от оси *OX* (знак угла положителен) к оси *OY* (знак угла отрицателен). Матрица преобразования будет следующей:

$$S=\begin{pmatrix}\cos(\varphi) & \sin(\varphi) & 0\\ -\sin(\varphi) & \cos(\varphi) & 0\\ 0 & 0 & 1\end{pmatrix}=\frac{1}{\sqrt{p^2+q^2}}\begin{pmatrix}q & p & 0\\ -p & q & 0\\ 0 & 0 & \sqrt{p^2+q^2}\end{pmatrix}. \qquad (67)$$

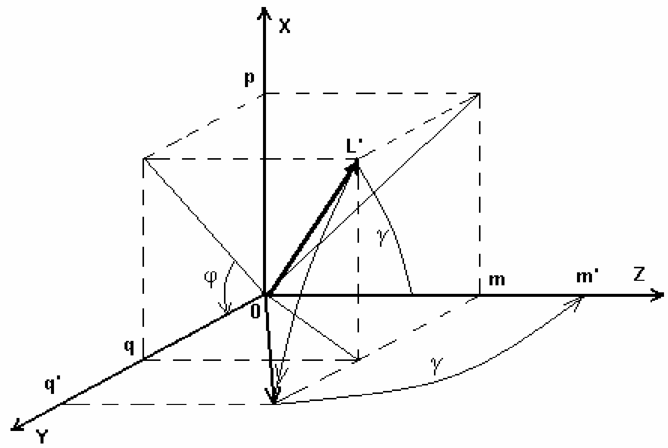


Рис.53. Поворот зеркала до положения,
когда нормаль к зеркалу будет перпендикулярна оси OX

Нам потребуется и обратная матрица. Для данного типа матриц обратной будет транспонированная матрица, поэтому:

$$S^{-1} = \frac{1}{\sqrt{p^2+q^2}} \begin{pmatrix} q & -p & 0 \\ p & q & 0 \\ 0 & 0 & \sqrt{p^2+q^2} \end{pmatrix}. \qquad (68)$$

Объединим все действия трёх матриц путём их перемножения и получим одну матрицу преобразования, осуществляющую развёртку зеркала в общем случае:

$$T = S^{-1}M \cdot S = \frac{1}{p^2+q^2} \begin{pmatrix} q^2-p^2 & 2pq & 0 \\ 2pq & p^2-q^2 & 0 \\ 0 & 0 & \sqrt{p^2+q^2} \end{pmatrix}. \qquad (69)$$

Как видим, поперечные координаты изображения изменяются с изменением наклона зеркала относительно оптической оси. Азимутальный угол будет меняться скачкообразно.

Аналогично преобразуются и угловые координаты луча, то есть направляющие косинусы луча также преобразуются скачкообразно.

При двумерном описании системы достаточно было следить за изменением апертурного угла, то есть угла между лучом и оптической осью. При введении в систему зеркал с общим положением необходимо прослеживать пространственное положение луча.

Чтобы упростить задачу трассировки косого луча через параксиальную оптическую систему воспользуемся следующим замечанием. Применим преобразование (69) к координатам осевой точки:

$$T \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix}.$$

Как видим, для осевой точки матрица T равносильна единичной матрице. Это значит, что при спрямлении оптической оси в системе с плоским зеркалом, зеркало не вносит никаких изменений в ход лучей осевой точки предмета. Оно может быть заменено плоскостью с одинаковыми показателями преломления слева и справа. С помощью программы Pupil в такой системе получаем информацию о положении задних отрезков всех поверхностей системы и далее мы можем использовать её при габаритном расчёте параксиальной оптической системы с косыми лучами. Этот приём использован ниже при составлении программы.

9.1. Модель с зеркальными отклоняющими элементами

Обладая информацией о задних отрезках системы, построим программу, с помощью которой кроме габаритного расчёта можно было бы промоделировать передаточную функцию погрешностей наклона и децентрировки элементов системы. Для более компактного представления и оперирования с информацией часть данных опишем в комплексной форме. Система MathCAD органично поддерживает комплексные вычисления и преобразования. Информационная матрица системы может иметь вид:

$$\mathrm{Info} := \begin{pmatrix} z'_0 & z'_1 & .. & z'_{N-1} & D_v \\ t_0 & t_1 & .. & t_{N-1} & z_v \\ p_0 & p_1 & .. & p_{N-1} & z_0 \\ D_0 & D_1 & .. & D_{N-1} & 1 \vee 0 \\ \delta\alpha_0 & \delta\alpha_1 & .. & \delta\alpha_{N-1} & \beta \\ \delta\rho_0 & \delta\rho_1 & .. & \delta\rho_{N-1} & \psi \end{pmatrix} \qquad (70)$$

Здесь первые *N* (*N* - число поверхностей) столбцов информационной матрицы отражают построение информации о каждой поверхности системы, начиная с первой по ходу луча.

Строка 1. Задний отрезок *k*-поверхности (*k*=0,1,…, *N*-1).

Строка 2. Толщина или воздушный промежуток до предыдущей поверхности (t_0=0).

Строка 3. Комплексное число *p + i q*, где *p*, q – соответственно *X*- и *Y*-компонента вектора направляющих косинусов нормали к поверхности.

Строка 4. Диаметр оправы *D* в виде *D+i*, где *i* – мнимая единица. Наличие мнимой единицы указывает на то, что данная оптическая поверхность есть плоскость.

Строка 5. Векторная угловая погрешность поверхности. Величина погрешности – это модуль числа. Аргумент числа – азимутальный угол.

Строка 6. Векторная угловая погрешность децентрировки. Величина погрешности – это модуль числа. Аргумент числа – азимутальный угол.

Последний столбец матрицы Info:

- диаметр входного зрачка,
- алгебраическое расстояние от осевой точки первой поверхности до входного зрачка,
- передний отрезок,
- признак диафрагм, если он 0, то вычисляются размеры диафрагм, осуществляющих виньетирование 0 либо 50%, если же он 1, то вычисляются коэффициенты виньетирования,
- полевой угол,
- азимутальный угол точки предмета.

Для проведения габаритного расчёта получим формулы вычисления координат луча на текущей оптической поверхности (главной плоскости). Обратимся к рис.54.

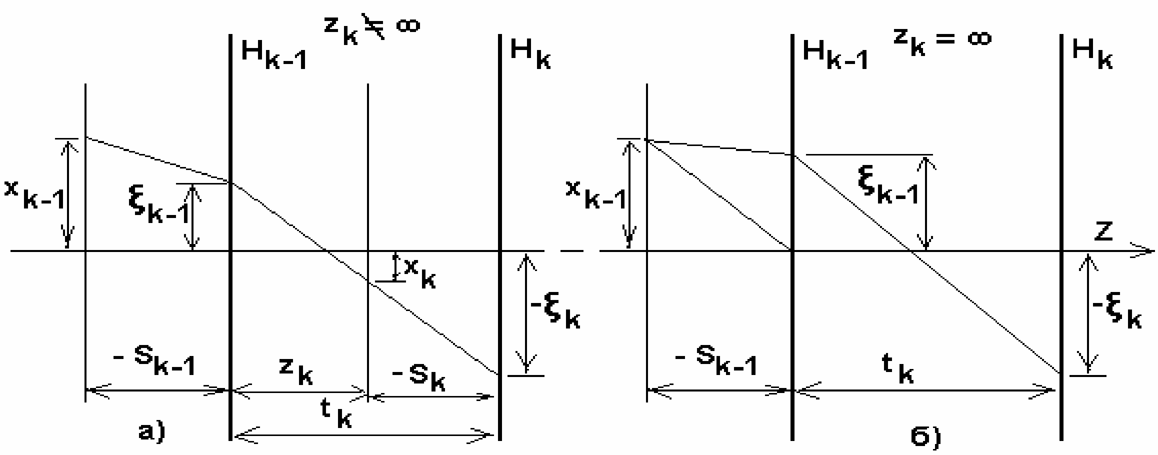


Рис.54. Преломление проекции косого луча на главной плоскости

Если передний отрезок $S_{k\text{-}1} \neq \infty$, то координаты точки ($x_{k\text{-}1}$, $y_{k\text{-}1}$) предмета (главная плоскость $H_{k\text{-}1}$) при условии, что и $z_k \neq \infty$, трансформируются с коэффициентом линейного увеличения:

$$V_{k-1} = \frac{z_k}{z_{k-1} - t_{k-1}}; \qquad (71)$$

для сферической оптической поверхности и с коэффициентом:

$$V_{k-1} = 1; \qquad (71а)$$

для плоской поверхности трансформируются в координаты изображения (x_k, y_k):

$$x_k = V_{k-1} x_{k-1}, \qquad y_k = V_{k-1} y_{k-1}. \qquad (72)$$

Положение луча на $H_{k\text{-}1}$ – главной плоскости известно - ($\xi_{k\text{-}1}$, $\eta_{k\text{-}1}$), тогда из геометрических свойств (рис.50а) получаем соотношения:

$$\left| \begin{aligned} \xi_k &= \xi_{k-1} + \frac{t_k}{z_k}(x_k - \xi_{k-1}), \\ \eta_k &= \eta_{k-1} + \frac{t_k}{z_k}(y_k - \eta_{k-1}) \end{aligned} \right. . \qquad (73)$$

Если же $z_k = \infty$ (рис. 50,б), то координаты луча на последующей поверхности находят из формул:

$$\left| \begin{aligned} \xi_k &= \xi_{k-1} + \frac{t_k}{z_{k-1} - t_{k-1}} x_{k-1}, \\ \eta_k &= \eta_{k-1} + \frac{t_k}{z_{k-1} - t_{k-1}} y_{k-1} \end{aligned} \right. . \qquad (74)$$

В этом случае для дальнейших преобразований вычисляем угол луча с оптической осью β и азимутальный угол ψ:

$$\mathrm{tg}(\beta_{k-1}) = \frac{\sqrt{x_{k-1}^2 + y_{k-1}^2}}{|z_{k-1} - t_{k-1}|}, \quad \mathrm{tg}(\psi) = \frac{y_{k-1}}{x_{k-1}}. \qquad (75)$$

Наконец, если $z_{k-1} = \infty$, (рис.55),

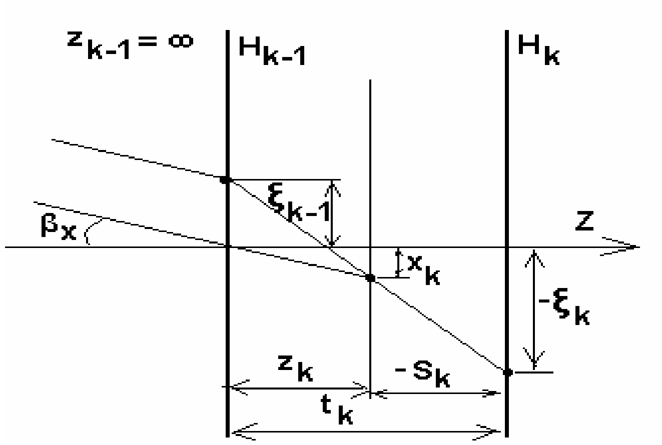


Рис.55. Преломление параллельного пучка

то координаты точки изображения вычисляем по формулам:

$$x_k = z_k \mathrm{tg}(\beta_{k-1})\cos(\psi_{k-1}), \quad y_k = z_k \mathrm{tg}(\beta_{k-1})\sin(\psi_{k-1}). \qquad (76)$$

Дальнейшие вычисления координат луча (ξ_k, η_k) на последующей главной плоскости производятся по формулам (73).

Координаты точки предмета вычисляются по формулам:

$$x = (z_V - z_0)\mathrm{tg}(\beta)\cos(\psi), \quad y = (z_V - z_0)\mathrm{tg}(\beta)\sin(\psi). \qquad (77)$$

При габаритном расчёте необходимо прослеживать прохождение трёх лучей, одного апертурного и двух габаритных. Возможность в среде MathCAD оперировать матричными вычислениями позволяет сократить объём программы. Положение трёх точек предмета или изображения будем описывать матрицей 2x3. Если плоскость предмета или изображения находится на конечном расстоянии, то матрица имеет вид

$$X = \begin{matrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \end{matrix} . \qquad (78)$$

Здесь (x_0,, y_0), (x_1,, y_1), (x_2,, y_2) координаты в плоскости предмета или изображения соответственно апертурного луча и двух габаритных, проходящих через диаметрально противоположные точки входного зрачка. Если же плоскость предмета или изображения расположена в бесконечности, то содержание матрицы меняется на угловое следующим образом

$$X = \begin{matrix} \mathrm{tg}(u_0) & \mathrm{tg}(u_1) & \mathrm{tg}(u2) \\ \psi_0 & \psi_1 & \psi_2 \end{matrix} . \qquad (78a)$$

Составим программу единичного акта преломления на оптической поверхности. Главную плоскостью поверхности задана передним S и задний отрезок S' и расстояниями до предыдущей и последующей поверхности. Для крайних поверхностей соответствующие величины заменяются нулями. Матрицу поверхности запишем в виде:

$$P = \begin{matrix} t_0 & t_1 \\ S & S' \end{matrix} . \qquad (79)$$

Блок схема программы представлена ниже. В вычисляемых блоках указаны номера формул, по которым производится вычисление.

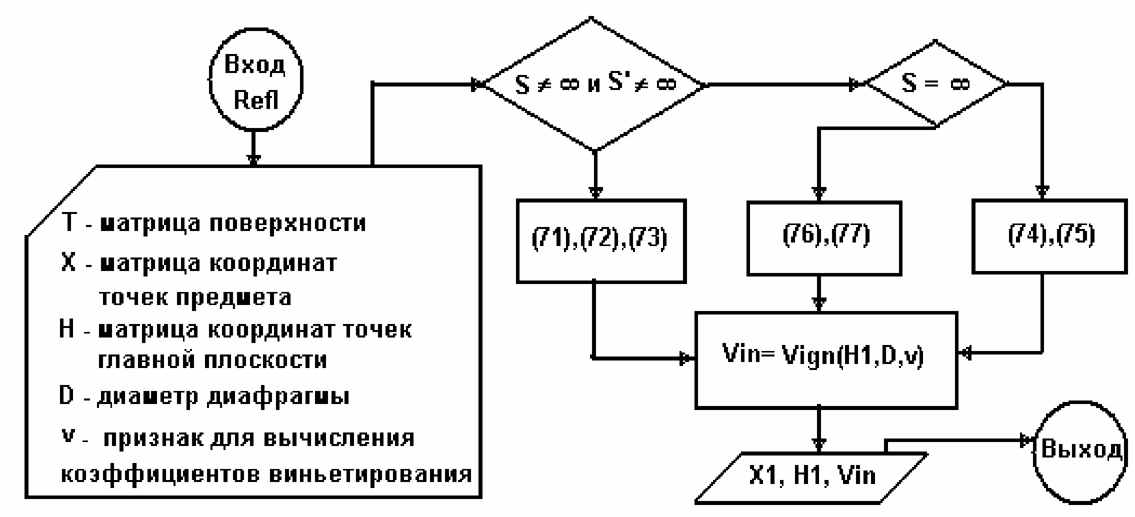


Рис.56. Блок схема программы преломления Refl

Текст программы Refl приведён ниже.

```
Refl(T,X,H,P,D,v) := | if T_{1,0} ≠ ∞ ∧ T_{1,1} ≠ ∞
                     |    | x1 ← if(Im(D) = 0, T_{1,1}/T_{1,0}, 1)·X
                     |    | ξ1 ← H + (T_{0,1}/T_{1,1})·(x1 − H)
                     | otherwise
                     |    | if T_{1,0} = ∞
                     |    |    | x1 ← T_{1,1}·stack( vectorize[ (X^T)^<0>·cos[ (X^T)^<1> ] ], vectorize[ (X^T)^<0>·sin[ (X^T)^<1> ] ] )
                     |    |    | ξ1 ← H + (T_{0,1}/T_{1,1})·(x1 − H)
                     |    | otherwise
                     |    |    | a ← augment( |X^<0>|, |X^<1>|, |X^<2>| )
                     |    |    | for k ∈ 0..2
                     |    |    |    ψ_k ← if(X_{1,k} > 0, π, −π)/2  on error  atan(X_{1,k}/X_{0,k}) + if(X_{0,k} > 0, 0, π)
                     |    |    | x1 ← stack[a, (ψ_0  ψ_1  ψ_2)]
                     |    |    | ξ1 ← H + (T_{0,1}·X)/(T_{1,0} − T_{0,0})
                     | Vin ← Vign(ξ1, v, P, Re(D))
                     | (x1  ξ1  Vin)^T
```

Рис.57. Текст программы преломления “косых” лучей на системе главной плоскости

Пояснения к программе Refl.

1) Вводимые величины *T*, *X* и *H* – матричные.

2) *P* – *X*- и *Y*- направляющие косинусы в виде комплексного числа *p* + *i q*.

3) При вычислении координат изображения *x*1 использована функция if, проверяющая наличие мнимой единицы в строке диаметров поверхностей. Наличие мнимой единицы указывает на то, что поверхность – плоскость.

4) Верхний индекс *k* в угольных скобках обозначает *k*-столбец матрицы.

5) Стрелка над операндами оператора умножения или деления вводит почленную операцию.

6) Оператор stack формирует матрицу путём присоединения матриц с одинаковым числом столбцов сверху вниз.

7) Оператор augment формирует матрицу из нескольких матриц с одинаковым количеством строк путём присоединения слева направо.

8) Функция on error вычисляет выражение слева от неё, если выражение справа невозможно вычислить.

9) Программа возвращает матрицу координат точек изображения $x1$, матрицу координат точек на главной плоскости (поверхности) и или матрицу диафрагм (v=0), или вектор коэффициентов виньетирования (v=1).

В данном случае составления программы использован модульный принцип. Следующим модулем служит программа Enter, которая формирует информацию на входе системы: определяет координаты точек пересечения лучей с первой поверхностью (главной плоскостью), высоты пересечения и входные направляющие косинусы и вычисляет размер диафрагм, обеспечивающих уровни 0 и 50% виньетирования, или коэффициент виньетирования. Кроме того, осуществляется учёт погрешностей наклона и децентрировки первой поверхности. Текст программы приведён ниже:

$$
\begin{aligned}
\mathrm{Enter}(Syst) := \;& N \leftarrow \mathrm{cols}(Syst) - 1\\
& \text{if } Syst_{2,N} \neq \infty\\
& \quad x \leftarrow \left(Syst_{1,N} - Syst_{2,N}\right)\cdot\tan\left(Syst_{4,N}\right)\cdot\cos\left(Syst_{5,N}\right)\\
& \quad y \leftarrow \left(Syst_{1,N} - Syst_{2,N}\right)\cdot\tan\left(Syst_{4,N}\right)\cdot\sin\left(Syst_{5,N}\right)\\
& \quad X \leftarrow \begin{pmatrix} 0 & x & x \\ 0 & y & y \end{pmatrix}\\
& X \leftarrow \begin{pmatrix} \tan\left(Syst_{4,N}\right) & \tan\left(Syst_{4,N}\right) & \tan\left(Syst_{4,N}\right) \\ Syst_{5,N} & Syst_{5,N} & Syst_{5,N} \end{pmatrix} \text{ otherwise}\\
& Y \leftarrow \begin{pmatrix} 0 & 0 & 0 \\ Syst_{0,N} & Syst_{0,N} & -Syst_{0,N} \end{pmatrix}\\
& f \leftarrow \mathrm{stack}\left(\overrightarrow{\left[\left(((X))^T\right)^{\langle 0\rangle}\cdot\cos\left[\left(((X))^T\right)^{\langle 1\rangle}\right]\right]}, \overrightarrow{\left[\left(((X))^T\right)^{\langle 0\rangle}\cdot\sin\left[\left(((X))^T\right)^{\langle 1\rangle}\right]\right]}\right)\\
& H \leftarrow Y + Syst_{1,N}\cdot \mathrm{if}\left[Syst_{2,N} \neq \infty, \frac{(X - Y)}{Syst_{1,N} - Syst_{2,N}}, f\right]\\
& Vin \leftarrow \mathrm{Vign}\left(H, Syst_{3,N}, Syst_{2,0}, \mathrm{Re}\left(Syst_{3,0}\right)\right)\\
& (X \;\; H \;\; Vin)^T
\end{aligned}
$$

Рис.58. Программа Enter формирования входных лучей

Пояснения к программе Enter.

1) Первый вычислительный блок задействован в случае конечного расстояния до плоскости предмета.

2) В первых двух строках блока вычисляются координаты точки изображения по заданным полевому и азимутальным углам, в третьей – формируется матрица точек предмета.

3) Второй вычислительный блок задействован в случае предмета на бесконечности. В нём формируется матрица направлений лучей.

4) Матрица *Y* содержит координаты крайних, диаметрально расположенных точек входного зрачка. В данном случае в плоскости *OYZ*.

5) Матрица направлений *f* формируется для компактной записи следующей строки, в которой по условию функции if, если передний отрезок равен бесконечности, вычисляется матрица точек пересечения лучей с передней главной плоскостью по формулам (76). Взаимодействие матричных операторов, используемых при формировании f описаны выше в пояснениях к программе Refl.

6) В предпоследней строке происходит обращение к программе виньетирования Vign.

И наконец, в программе ModelMir (рис.62) объединяются модули Refl и Enter в программу габаритного расчёта и учёта влияния погрешностей наклона и децентрировки поверхностей системы.

Пояснения к программе ModelMir

1) До цикла по *N* поверхностям системы (*k*=0..*N*-1) происходит обращение к подпрограмме Enter, с помощью которой формируется матрица координат или направлений в пространстве предмета, матрица координат точек лучей на входной поверхности и информация об диафрагмах и виньетировании.

2) В первой строке цикла вычисляется величина переднего отрезка.

3) Первый внутренний цикл производит корректировку координат точки предмета из-за погрешностей наклона и децентрировки поверхности в случае конечного расстояния до плоскости предмета. Для компактной записи используется комплексная форма представления входных данных.

4) Второй внутренний цикл задействован для корректировки направлений лучей в пространстве предмета в случае бесконечного расстояния до плоскости предмета.

5) В следующей строке происходит присвоение расстояния между поверхностями или последнего отрезка в случае последней поверхности переменой *t*1.

6) Формируется матрица параметров *m*.

7) Переменной *a* присваивается результат вычислений в программе Refl.

8) До цикла по *N* поверхностям системы (*k*=0..*N*-1) происходит обращение к подпрограмме Enter, с помощью которой формируется матрица координат или направлений в пространстве предмета, матрица координат точек лучей на входной поверхности и информация о диафрагмах и виньетировании.

9) В первой строке цикла вычисляется величина переднего отрезка.

10) Первый внутренний цикл производит корректировку координат точки предмета из-за погрешностей наклона и децентрировки поверхности в случае конечного расстояния до плоскости предмета. Для компактной записи используется комплексная форма представления входных данных.

11) Второй внутренний цикл задействован для корректировки направлений лучей в пространстве предмета в случае бесконечного расстояния до плоскости предмета.

```
ModelMir(Syst) := | A ← Enter(Syst)
                  | X ← A_0
                  | H ← A_1
                  | Vin_0 ← A_2
                  | N ← cols(Syst) − 1
                  | for k ∈ 0..N − 1
                  |   | z0_k ← if(k = 0, Syst_{2,N}, Syst_{0,k−1} − Syst_{1,k})
                  |   | q ← if(|Syst_{4,k}| = 0, 0, arg(Syst_{4,k}))
                  |   | for o ∈ 0..2                                              if z0 ≠ ∞
                  |   |   | r ← sqrt((|X^<o>|)^2 + (z0_k)^2)
                  |   |   | X_{0,o} ← X_{0,o} − r·tan(|Syst_{4,k}|)·cos(q) − Re(Syst_{5,k})
                  |   |   | X_{1,o} ← X_{1,o} − r·tan(|Syst_{4,k}|)·sin(q) − Im(Syst_{5,k})
                  |   | for o ∈ 1..2                                              otherwise
                  |   |   | X_{0,o} ← tan(atan(X_{0,o}) − |Syst_{4,k}|)
                  |   |   | X_{1,o} ← X_{1,o} − q
                  |   | t1 ← if(k ≠ N − 1, Syst_{1,k+1}, Syst_{0,k})
                  |   | m ← [ Syst_{1,k}   t1        ]
                  |   |     [ z0_k         Syst_{0,k} ]
                  |   | a ← Refl(m, X, H, Syst_{2,k+1}, Syst_{3,k+1}, Syst_{3,N})
                  |   | X ← a_0
                  |   | H ← a_1
                  |   | Vin_{k+1} ← a_2
                  | [ X^<1> ]
                  | [ Vin   ]
```

Рис.59. Текст программы ModelMir

12) До цикла по *N* поверхностям системы (k=0..N-1) происходит обращение к подпрограмме Enter, с помощью которой формируется матрица координат или направлений в пространстве предмета, матрица координат точек лучей на входной поверхности и информация о диафрагмах и виньетировании.

13) В первой строке цикла вычисляется величина переднего отрезка.

14) Первый внутренний цикл производит корректировку координат точки предмета из-за погрешностей наклона и децентрировки поверхности в случае конечного расстояния до плоскости предмета. Для компактной записи используется комплексная форма представления входных данных.

15) Второй внутренний цикл задействован для корректировки направлений лучей в пространстве предмета в случае бесконечного расстояния до плоскости предмета.

16) В следующей строке происходит присвоение расстояния между поверхностями или последнего отрезка в случае последней поверхности переменой *t*1.

17) Формируется матрица параметров *m*.

18) Переменной *a* присваивается результат вычислений в программе Refl.

19) Далее следует присвоение координат точек изображения и последующей поверхности (или плоскости изображения системы в случае *k*=*N*-1) соответственно переменным *X* и *H*, а также обращение к программе Vign, которое происходит, если поверхность не последняя.

20) Программа возвращает координаты точки изображения и информацию о виньетировании по поверхностям.

21) Текст программы Vign (рис.47) дополняется операторами матричной обработки входной матрицы *X* для формирования высот лучей, здесь применяется операция поворота системы координат до совмещения линии координат габаритных лучей с осью *OX* (рис.60).

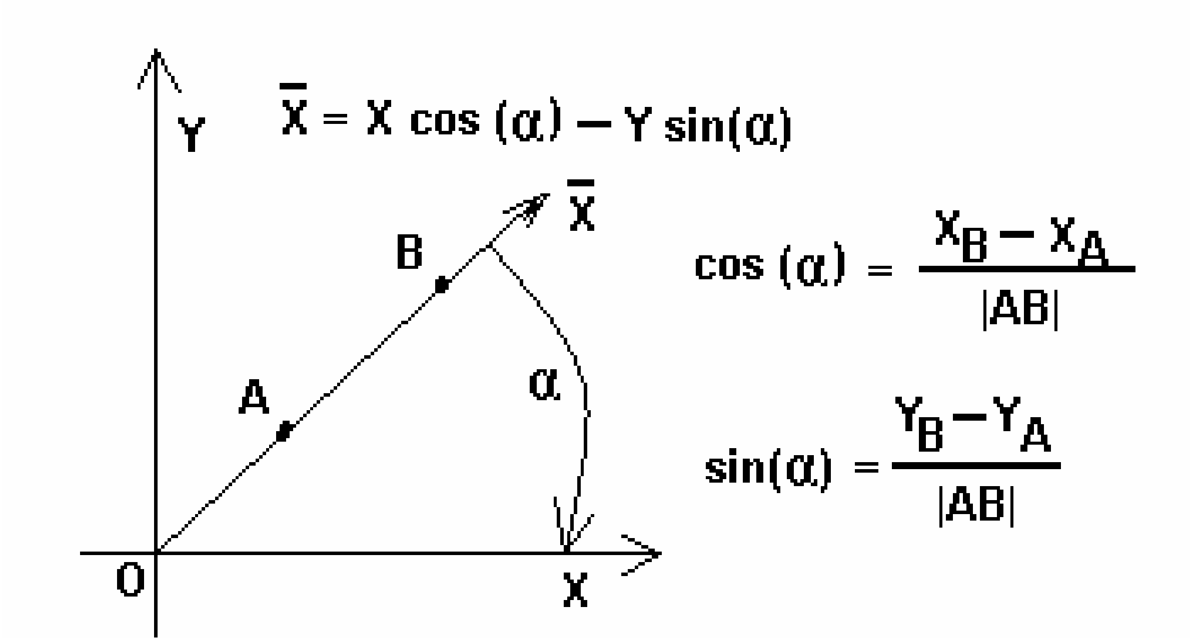


Рис.60. К формированию высот лучей

$$
Vign(X,V,P,D) := \left|
\begin{array}{l}
\text{return } 0 \text{ if } \left|X^{\langle 1\rangle} - X^{\langle 2\rangle}\right| < 10^{-9} \\
h_0 \leftarrow \left|X^{\langle 0\rangle}\right| \\
h_1 \leftarrow \dfrac{X^{\langle 1\rangle}\cdot X^{\langle 2\rangle} - \left(\left|X^{\langle 1\rangle}\right|\right)^2}{\left|X^{\langle 1\rangle} - X^{\langle 2\rangle}\right|} \\
h_2 \leftarrow \dfrac{\left(\left|X^{\langle 2\rangle}\right|\right)^2 - X^{\langle 1\rangle}\cdot X^{\langle 2\rangle}}{\left|X^{\langle 1\rangle} - X^{\langle 2\rangle}\right|} \\
\text{return } \begin{pmatrix} 2\cdot \max\left(\left|h_1\right|, \left|h_2\right|\right) \\ 2\cdot Dv\left(0.5, h_1, h_2\right) \\ 2\cdot \left|h_0\right| \end{pmatrix} \cdot \dfrac{1}{\sqrt{1-\left(\left|P\right|\right)^2}} \text{ if } V = 0
\end{array}
\right.
$$

Далее по тексту програмы (рис.50)

Рис.61. Дополнения к программе Vign

9.2. Прямоугольная призма (пример)

Для примера рассмотрим габаритный расчёт прямоугольной призмы и определим передаточный коэффициент суммарной первичной ошибки параллельности граней призмы для случая, когда угловой размер предмета, расположенного перед призмой на расстоянии 20 мм, составляет $\beta = \pm 5^o$ (рис.62). Световой диаметр входного пучка *d* = 10 мм. Задача решается подбором габарита призмы, такого чтобы выходной световой диаметр с виньетированием 0% был не больше размера выходной грани *D*.

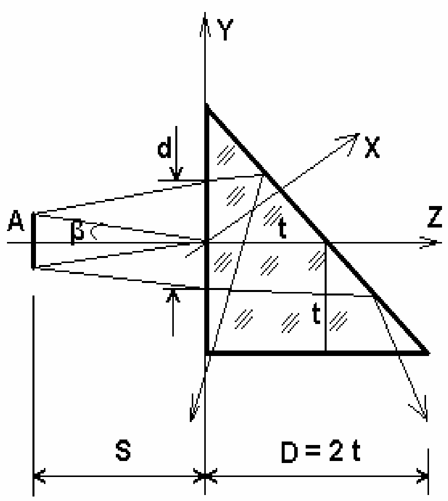


Рис.62. Габаритный расчёт прямоугольной призмы

Зададимся полуинтервалом угловой погрешности плоскостей призмы |*c*| <0.1 угл.мин. и равномерной функцией плотности вероятности погрешности. Проведём моделирование погрешностей и составим информационную матрицу. Фрагмент документа приведён ниже.

$$M := 20 \qquad j := 0..M-1 \qquad c := 0.1$$

$$\alpha1 := \mathrm{runif}\left(M, \frac{-c}{60}\cdot deg, \frac{c}{60}\cdot deg\right) \qquad \psi1 := \mathrm{runif}(M, 0, 2\pi)$$

$$\alpha2 := \mathrm{runif}\left(M, \frac{-c}{60}\cdot deg, \frac{c}{60}\cdot deg\right) \qquad \psi2 := \mathrm{runif}(M, 0, 2\pi)$$

$$\alpha3 := \mathrm{runif}\left(M, \frac{-c}{60}\cdot deg, \frac{c}{60}\cdot deg\right) \qquad \psi3 := \mathrm{runif}(M, 0, 2\pi)$$

$$SystR_j := \begin{array}{r} \mathbf{s'} \\ \mathbf{t} \\ \mathbf{p+i\ q} \\ \mathbf{D} \\ \Delta\alpha \\ \Delta\rho \end{array} \left(\begin{array}{ccccc} -30 & -40 & \frac{-100}{3} & 5 & \mathbf{d} \\ 0 & \boxed{10} & \boxed{10} & 0 & \mathbf{z_v} \\ 0 & \frac{-\sqrt{2}}{2}\cdot i & 0 & -20 & \mathbf{z_0} \\ 20+i & 20+i & 20+i & 0 & \mathbf{v} \\ \alpha1_j\cdot e^{i\cdot\psi1_j} & \alpha2_j\cdot e^{i\cdot\psi2_j} & \alpha3_j\cdot e^{i\cdot\psi2_j} & 5\cdot deg & \boldsymbol{\beta} \\ 0 & 0 & 0 & \frac{\pi}{2} & \boldsymbol{\psi} \end{array}\right)$$

В первой строке матриц $SystR_j$ стоят задние отрезки плоскостей призмы. Для их определения нет необходимости в программе Pupil. Воспользуемся формулой отрезков:

$$S' = \frac{n'}{n} S\,. \tag{80}$$

Матрица снабжена комментариями: слева строк, справа – последнего столбца. Пунктиром отмечены подбираемые габаритные параметры призмы.

Для габаритного расчёта воспользуемся одной из информационных матриц $SystR_0$. Результаты расчёта представлены ниже

$$D := \mathrm{ModelMir}(SystR_0)_1 \quad D = \begin{pmatrix} \{3,1\} \\ \{3,1\} \\ \{3,1\} \\ 0 \end{pmatrix} \quad D_0 = \begin{pmatrix} 10 \\ 5 \\ 10 \end{pmatrix} \quad D_1 = \begin{pmatrix} 14.5 \\ 6.667 \\ 13.334 \end{pmatrix} \quad D_2 = \begin{pmatrix} 18.999 \\ 8.333 \\ 16.667 \end{pmatrix}$$

Структурной переменной *D* присваиваются результаты расчёта по программе ModelMir, касающиеся расчёта диаметров. Выходные диаметры находятся в третьей строке переменной D_2. Параметр *t* подбираем так, чтобы первый элемент вектора D_2 определяющий искомый размер призмы, пропускающей без виньетирования заданный пучок света, удовлетворял условию: 18,999 < 2*t*=2*10.

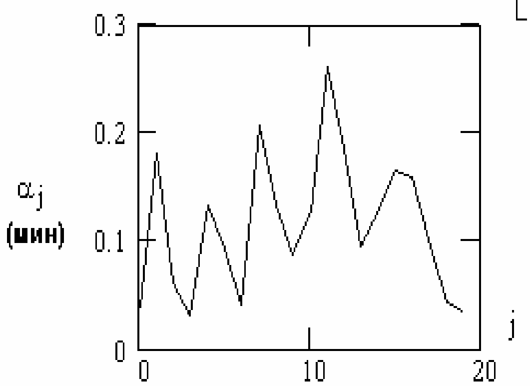
Далее, поставляя на вход программе MolelMir последовательность системных матриц, в первом элементе выводного вектора мы получаем координаты точки изображения x_j и y_j, j=0,1,…,19. С помощью функции stdev определяем СКО координат точек изображения. Зная величину последнего отрезка, рассчитываем угловую погрешность и строим график. Отношение среднего углового отклонения к величине полуинтервала первичной суммарной векторной погрешности (c) даёт среднее значение передаточного коэффициента *K:*

$$A_j := \mathrm{ModelMir}\left(\mathrm{SystR}_j\right)_0 \qquad x_j := \left(A_j\right)_0 \qquad \mathrm{stdev}(x) = 8.692 \times 10^{-4}$$

$$y_j := \left(A_j\right)_1 \qquad \mathrm{stdev}(y) = 9.282 \times 10^{-4}$$

$$\alpha_j := \mathrm{atan}\left[\frac{\sqrt{\left(x_j - \mathrm{mean}(x)\right)^2 + \left(y_j - \mathrm{mean}(y)\right)^2}}{\left|\left(\mathrm{SystR}_0\right)_{0,2}\right|}\right]\cdot\frac{60}{deg}$$

Последний отрезок



$$\mathbf{K} = \frac{\mathrm{mean}(\alpha)}{c} = 1.15$$

10. Моделирование оптических систем с призменными элементами

10.1. Плоскопараллельная пластина

А) Плоскопараллельная пластина перпендикулярна оптической оси.

Этот случай полностью включается в рассмотренную ранее модель осесимметричной оптической системы. Плоскопараллельная пластина, перпендикулярная оптической оси, может быть рассмотрена как линза с бесконечными радиусами кривизны.

Заметим, что, как и всякая линза, плоскопараллельная пластина имеет сферическую аберрацию. В параксиальном приближении аберрации отсутствуют. Действие любого оптического элемента в параксиальном приближении сводится к пространственному смещению изображения относительно предмета. Для плоскопараллельной пластины это смещение выражается наиболее просто.

Последовательно применим инвариант Аббе для плоскопараллельной пластины толщиной d, с показателем преломления n_2, помещённой в среду с показателем преломления n_1. Правая часть инвариантов равна нулю, поскольку оптическая сила плоскости равна нулю.

$$\frac{n_2}{S_1'} - \frac{n_1}{S_1} = 0, \qquad S_1' = \frac{n_2}{n_1} S_1,$$

$$S_2 = S_1' - d, \quad \rightarrow \quad S_2 = \frac{n_2}{n_1} S_1 - d,$$

$$\frac{n_1}{S_2'} - \frac{n_2}{S_2} = 0. \qquad S_2' = \frac{n_1}{n_2} S_2 = S_1 - \frac{n_1}{n_2} d \quad \rightarrow \Delta S = d + S_2' - S_1 = \frac{n_2 - n_1}{n_2} d$$

(81)

В воздухе (n_1=1) смещение изображения плоскопараллельной пластиной:

$$\Delta S = \frac{n-1}{n} d. \qquad (82)$$

Таким образом, наличие в системе плоскопараллельной пластины, перпендикулярной оптической оси, при габаритном расчёте может быть учтено с помощью введения поправки ΔS на продольное положение изображения (рис.63):

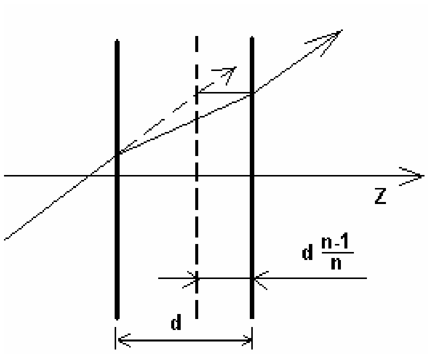


Рис.63. Редуцирование плоскопараллельной линзы к воздуху

Такая операция, исключающая трассировку лучей через плоскопараллельную пластину, называется редуцированием данного элемента к воздуху. Учёт элемента происходит на завершающем этапе – определении координат изображения. Однако в этом случае мы лишаемся возможности определения габаритных размеров элемента.

10.2. Моделирование призм

Воздействие призмы на изображение точки сводится к последовательности преломлений и отражений от плоскостей, расположенных, в общем случае, произвольно относительно оптической оси. Оптическая ось, отождествлённая с главным лучом осевого пучка, распространяющимся в идеальной призме, то есть имеющей номинальные значения углов и абсолютную однородность среды, претерпевает пространственные изломы в актах преломления и отражения.

Компьютерная модель призмы предназначена для исследования влияния технологических и юстировочных погрешностей призмы, определения передаточных коэффициентов влияния, оценки допусков на первичные погрешности, проведения габаритного расчёта.

Поскольку в конструкторской практике наряду с прямым расчётом призм для удобства расчётов используется приём “развёртка призм” (см., например, [5]) рассмотрим два варианта модели призм: без развёртки, в общем случае пространственное моделирование, и с развёрткой – двумерное, плоскостное моделирование.

В обоих случаях сформулируем исходные данные. Пусть ось *OZ* декартовой системы координат направлена по оптической оси в пространстве предмета и имеет начало *O*, расположенное в точке пересечения оптической осью входной грани призмы (рис.64).

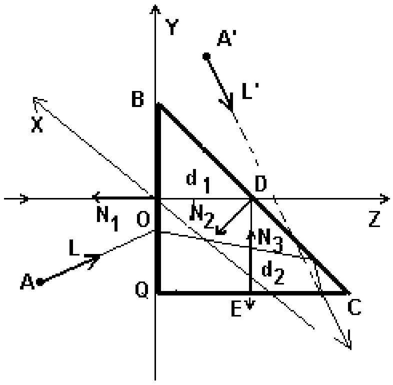


Рис.64. Положение призмы в системе координат

Пространственное положение граней призмы определяется отрезками d_1, d_2 и т.д. по ходу изломов оптической оси в номинальной призме и направляющими косинусами граней, имеющих реальное положение с учётом погрешностей углов призмы. В пространстве предмета задано положение точки предмета *A* и/или направление падающего на призму луча *L*.

Пространственное моделирование призм

С помощью модели требуется определить положение точки изображения и/или направление луча после последней грани призмы, с произвольно ориентированными вследствие угловых погрешностей гранями. Расстояния между центрами граней могут включать погрешности толщины призмы. Сначала рассмотрим вывод формул, описывающих построение изображения, а затем преломление на плоской оптической поверхности. Если задано только положение точки предмета, то изображение строится в параксиальном приближении, т.е. с помощью лучей, близких к нормали к поверхности: $L = -N$.

Построение изображения точки плоской оптической поверхностью

Рассмотрим произвольно ориентированную плоскость, разделяющую оптические среды с показателями преломления *n* и *n’*. Начало декартовой системы координат расположено в точке пересечения плоскости с оптической осью (рис.65).

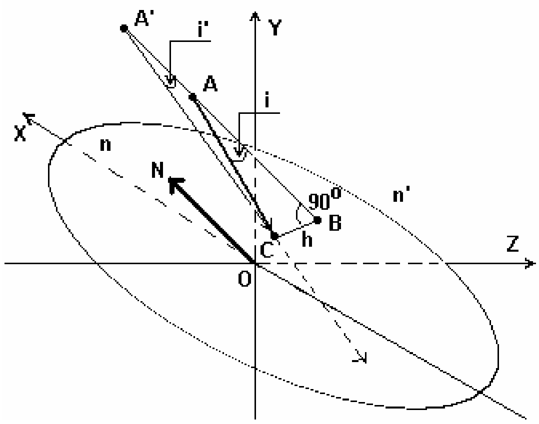


Рис.65. Построение изображения с помощью оптической плоскости.
N – нормаль к поверхности, i – угол падения, i' – угол преломления.
A – точка предмета, A’ – точка изображения для луча с высотой h

Предметный луч *AC* преломляется на поверхности *OBC* и следует в направлении *A’C*. На пересечении с нормалью к поверхности *AB* находится изображение точки *A*, построенное лучом, пересекающим поверхность на высоте *h*.

Дано: Координаты точки предмета $A(x_0, y_0, z_0)$ *и/или направление лучевого вектора* $n \cdot AC = L(p_0, q_0, m_0)$. *Направление орта нормали к поверхности* $\vec{N}(p, q, m)$.

Из двух возможных направлений нормали выбираем то, которое располагается в первой среде.

Определить: Координаты точки изображения $A'(x', y', z')$ *и/или направление преломлённого лучевого вектора* $n' \cdot A'C = L'(p', q', m')$.

Обозначим передний отрезок *S* = *BA*, задний отрезок *S’* = *BA’*. Из треугольников *ABC* и *A’BC*, имеющих общую высоту *h*, получаем соотношение, связывающее отрезки:

$$S' = \frac{S}{n \cdot \cos(i)} \sqrt{n'^2 - n^2 \sin^2(i)} \,. \tag{83}$$

Обобщим его на случай отражения:

$$S' = \frac{sign(n')S}{n \cdot \cos(i)} \sqrt{n'^2 - n^2 \sin^2(i)} \tag{84}$$

Положение точки изображения *A’* свяжем с положением точки предмета *A* и основанием нормали *B*, используя (84) и тот факт, что *A’* лежит на прямой, проходящей через точки *A* и *B*, уравнением:

$$A' = B + \frac{A - B}{S} S' = B + (A - B)\frac{sign(n')}{n \cdot \cos(i)}\sqrt{n'^2 - n^2 \sin^2(i)}\,. \qquad (85)$$

Используя методы аналитической геометрии в пространстве [6], получим координаты точки $B(x_1, y_1, z_1)$, основания перпендикуляра *AB*, и выразим угол падения через исходные данные:

$$B = I_0 \cdot A, \quad I_0 = -\begin{matrix} p^2 - 1 & p \cdot q & m \cdot p \\ p \cdot q & q^2 - 1 & m \cdot q \\ p \cdot m & q \cdot m & m^2 - 1 \end{matrix}\,, \qquad (86)$$

$$\cos(i) = -\frac{(N, L)}{n}\,. \qquad (87)$$

Объединяя (85), (86) и (87), получим:

$$A' = M \cdot A, \quad M = [I_0 - (I_0 - E)t], \quad t = \frac{sign(n')}{(N, L)}\sqrt{n'^2 - n^2 + n^2 \cdot (N \cdot L)^2}\,. \qquad (88)$$

В случае отражения t = -1 и матрица преобразования *M* имеет вид:

$$M = 2I_0 - E\,. \qquad (88a)$$

И так, соотношение (88) с учётом (86) даёт решение задачи построения изображения плоской оптической поверхностью.

Преломление луча на плоской оптической поверхности.

Воздействие оптической поверхности на падающий луч таково, что вектор нормали *N* поверхности в точке встречи луча с поверхностью коллинеарен вектору убыли лучевого вектора [7].

$$\begin{aligned} \mu \cdot p &= p_0 - p' = -\Delta p \\ \mu \cdot q &= q_0 - q' = -\Delta q \quad , \\ \mu \cdot m &= m_0 - m' = -\Delta m \end{aligned} \qquad (89)$$

где μ – коэффициент пропорциональности исключаем с помощью третьего уравнения в (89):

$$\Delta p = \frac{p}{m}\Delta m, \quad \Delta q = \frac{q}{m}\Delta m\,. \qquad (90)$$

Используя соотношения для лучевых компонент $p_0{}^2 + q_0{}^2 + m_0{}^2 = n^2$,

$p'^2 + q'^2 + m'^2 = n'^2$, исключаем Δp и Δq из (90) и получаем квадратное уравнение относительно $\frac{\Delta m}{m}$:

$$\frac{\Delta m^2}{m^2} + 2(p_0 p + q_0 q + m_0 m)\frac{\Delta m}{m} - (n'^2 - n^2) = 0 . \qquad (91)$$

Решение уравнения (91) после анализа знаков для случаев преломления и отражения запишется в виде:

$$\mu = -\frac{\Delta m}{m} = \tau - sign(n' \cdot \tau)\sqrt{\tau^2 + n'^2 - n^2}, \quad где\ \tau = (L \cdot N) = (p_0 p + q_0 q + m_0 m) \qquad (92)$$

С учётом (92) получаем искомую формулу для лучевых компонент преломлённого луча:

$$\begin{matrix} p' \\ q' \\ m' \end{matrix} = \begin{matrix} p_0 \\ q_0 \\ m_0 \end{matrix} - \begin{matrix} p \\ q \\ m \end{matrix} \left[(N \cdot L) - sign\left(n' \cdot (N \cdot L)\right)\right]\sqrt{(N \cdot L)^2 + n'^2 - n^2}) . \qquad (93)$$

Необходимые выражения для анализа погрешностей получены.

Наконец, данную модель можно расширить на случай выполнения габаритного расчёта. Для этого требуется получить выражение для координат точки встречи произвольного луча с поверхностью (точка *C* (рис.65)). Решением является обобщение соотношения (86). Координаты точки *C* получим аналогично координатам точки *B*:

$$C = I \cdot A, \quad I = \frac{1}{(N \cdot L)} \begin{matrix} -(mm_0 + qq_0) & p_0 \cdot q & m \cdot p_0 \\ p \cdot q_0 & -(pp_0 + mm_0) & m \cdot q_0 \\ p \cdot m_0 & q \cdot m_0 & -(pp_0 + qq_0) \end{matrix} . \qquad (94)$$

Моделирование развёртки призм

Развёртка призмы, преобразующей ход лучей, должна сводиться к плоскопараллельной пластине. Вследствие погрешностей углов и параллельности граней призма разворачивается в клин, отклонение лучей на котором задаётся суммарной погрешностью функционирования призмы.

Развёртка реальной призмы осуществляется последовательным спрямлением изломов осевого луча, происходящих в результате преломления и отражения от граней, имеющих ориентацию, учитывающую первичные погрешности.

Направляющие косинусы преломлённого (отражённого) осевого луча найдём с помощью (93), положив $p_0 = q_0 = 0$, $m_0 = n$, $L = n(0,0,1)$.

$$\begin{matrix} p'_0 \\ q'_0 \\ m'_0 \end{matrix} = \frac{1}{n'} \begin{matrix} 0 \\ 0 \\ n \end{matrix} - \begin{matrix} p \\ q \\ m \end{matrix} \left[nm - sign\left(n' \cdot nm\right)\right]\sqrt{(nm)^2 + n'^2 - n^2}) . \qquad (95)$$

В случае отражения (95) трансформируется к виду:

$$\begin{pmatrix} p_0' \\ q_0' \\ m_0' \end{pmatrix} = 2m \cdot \begin{pmatrix} p \\ q \\ m \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}. \qquad (95a)$$

Чтобы осуществить развёртку грани с направляющими косинусами (*p,q,m)* необходимо вектор (p_0', q_0', m_0') (95) совместить с осью *OZ*. Эту операцию удобно провести в два шага. Сначала повернём пространство изображения вокруг оси *OZ* на угол φ до совмещения проекции орта преломлённого осевого луча на плоскость *OXY* с осью *OY*, затем поворотом вокруг оси *OX* на угол γ завершим совмещение направлений.

Матрица преобразований с учётом того, что

$$\cos(\gamma) = m, \quad \sin(\gamma) = \sqrt{1-m^2}, \quad \sin(\varphi) = \frac{p}{\sqrt{1-m^2}}, \quad \cos(\varphi) = \frac{q}{\sqrt{1-m^2}},$$

имеет вид:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) \\ 0 & -\sin(\gamma) & \cos(\gamma) \end{pmatrix} \cdot \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\cos(\gamma)\sin(\varphi) & \cos(\gamma)\cos(\varphi) & \sin(\gamma) \\ \sin(\gamma)\sin(\varphi) & -\sin(\gamma)\cos(\varphi) & \cos(\gamma \end{pmatrix} = \begin{pmatrix} \frac{q}{\sqrt{1-m^2}} & \frac{p}{\sqrt{1-m^2}} & 0 \\ -\frac{m \cdot p}{\sqrt{1-m^2}} & \frac{m \cdot q}{\sqrt{1-m^2}} & \sqrt{1-m^2} \\ p & q & m \end{pmatrix} \qquad (96)$$

Если же *p*=*q*=0, то *R*=*E*, где *E* – единичная матрица.

Таким образом, получены все формулы для построения модели призмы в двух вариантах: с развёрткой и без неё. Моделирование проведём в среде на примере прямоугольной призмы. Ограничимся случаем моделирования угловых погрешностей.

Пространственная модель призмы

Рассмотрим пример прямоугольной призмы (*AP*-90^0), имеющей погрешности углов её граней (рис.66).

Каждая из трёх граней призмы имеет по две погрешности углов, указанных на рис.4 стрелками. Номинальные значения векторов нормали с учётом излома оптической оси:

$$N_1(0, 0, -1), \quad N_2\left(0, -\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right), \quad N_3(0, 0, -1).$$

Орт падающего на призму луча: $L(0, 0, 1)$. Пусть *n*=1, *n’*= 1,5.

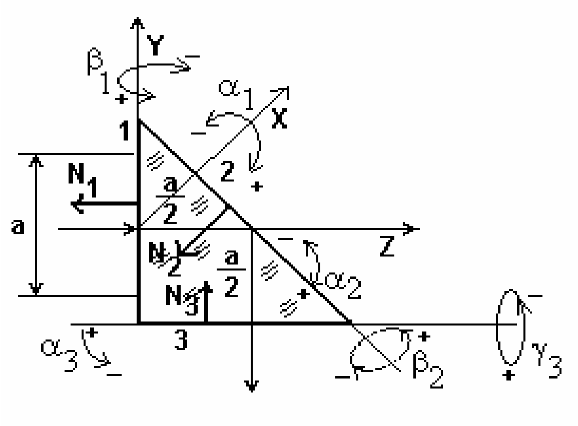


Рис.66. Прямоугольная призма

Среди допусков на углы призм рассматривают [8] допуски на сами углы и допуски на параллельность граней, пирамидальность. Очевидно, что эти допуски могут быть составлены из допусков на отклонение нормалей граней от их номинальных положений. При этом возможны две ситуации моделирования.

Во-первых, задавшись величиной одной из погрешностей, определяют частичную погрешность угловую или пространственную и, взяв их отношение, определяют величину передаточного коэффициента данной первичной погрешности. Расчёт допусков проводится по известным методикам [9] на основе передаточных коэффициентов.

Во-вторых, исходя из уровня технологичности, задаются интервалами первичных погрешностей и законами их распределения, быть может, с учётом передаточных коэффициентов, и определяют СКО суммарной погрешности. Налаживая обратную связь, можно подобрать удовлетворительный набор величин первичных погрешностей, обеспечивающий необходимую суммарную погрешность.

Например, смоделируем вторую ситуацию. Зададим в общем случае шесть одинаковых полуинтервалов погрешностей углов граней.

$$\Delta\alpha_1 = \Delta\beta_1 = \Delta\alpha_2 = \Delta\beta_2 = \Delta\alpha_3 = \Delta\gamma_3 = 2'.$$

Размерность случайных векторов погрешностей, например, N=20. Зададимся равномерным законом распределения всех первичных погрешностей, используя встроенную в MathCAD функцию:

$$\alpha_1 = runif\,(N, -\Delta\alpha_1, \Delta\alpha_1), \quad \alpha_2 = runif\,(N, -\Delta\alpha_2, \Delta\alpha_2), \ldots \quad и \quad т.\,д.$$

Определим реальные направления ортов нормалей граней призмы:

$$\vec{N}_1\left(\sin(-\beta_1), \quad \sin(\alpha_1), \quad -\sqrt{1-\sin^2(\beta_1)-\sin^2(\alpha_1)}\right),$$

$$\vec{N}_2\left(\sin(-\beta_2), \quad \sin(-\frac{\pi}{4}+\alpha_2), \quad -\sqrt{1-\sin^2(\beta_2)-\sin^2(\alpha_2-\frac{\pi}{4})}\right),$$

$$\vec{N}_3\left(\sin(-\gamma_3), \quad \sin(\alpha_3), \quad -\sqrt{1-\sin^2(\gamma_3)-\sin^2(\alpha_3)}\right).$$

Для вычисления направления выходного луча составим программу. Исходную информацию относительно направляющих косинусов нормалей граней, направляющих косинусов входного луча и значение показателя преломления призмы в воздухе удобно разместить в матрице, например:

$$Syst = \begin{matrix} p_0 & p_1 & p_2 & p_3 \\ q_0 & q_1 & q_2 & q_3 \\ n & m_1 & m_2 + i & m_3 \end{matrix} .$$

В первом столбце расположены направляющие косинусы входного луча и показатель преломления материала призмы, в следующих столбцах по ходу луча указаны направляющие косинусы текущих ортов нормалей поверхностей, причём признаком отражающей поверхности служит мнимое слагаемое в третьем столбце, соответствующем второй отражающей поверхности.

Текст программы приведён ниже на рис.67.

$$\mathrm{React}(L,N,n,n') := \left| \begin{array}{l} NL \leftarrow N \cdot L \\ \left[L - N \cdot \left(NL - \mathrm{sign}(n' \cdot NL) \cdot \sqrt{NL^2 + n'^2 - n^2} \right) \right] \end{array} \right.$$

$$\mathrm{PrizmP}(S) := \left| \begin{array}{l} k \leftarrow \mathrm{cols}(S) - 1 \\ L \leftarrow S^{\langle 0 \rangle} \\ n \leftarrow L_2 \\ L_2 \leftarrow \sqrt{1 - (L_0)^2 - (L_1)^2} \\ \text{for } i \in 1..k \\ \quad \left| \begin{array}{l} n_t \leftarrow \mathrm{if}(i = 1, 1, n) \\ n'_t \leftarrow \mathrm{if}(i = 1, n, \mathrm{if}(i = k, 1, -n)) \\ L \leftarrow \mathrm{React}\left(L, S^{\langle i \rangle}, n_t, n'_t\right) \end{array} \right. \\ L \end{array} \right.$$

Рис.67. Текст программ для моделирования углов отклонения призмой

Программа состоит из двух частей. Первая часть, программа React, – вычисление преломлённого орта на заданной оптической поверхности по формуле (95), *L* – орт падающего луча, *N* – орт нормали к поверхности, *n*, *n*` - показатели преломления сред. Вторая часть – собственно вычисление направления орта выходного луча. В начале программа обрабатывает информационную матрицу: определяет число граней (*k*), строит орт падающего луча (L), определяет величину показателя преломления призмы (n). Далее в цикле по граням призмы происходит определение текущих значений показателей преломления (n_t, n'_t) и следующее значение орта луча, после взаимодействия с текущей оптической поверхностью.

Результаты моделирования приведены на графиках (рис.68) зависимости проекций углов на координатные оси и модуля углов отклонения от номера эксперимента.

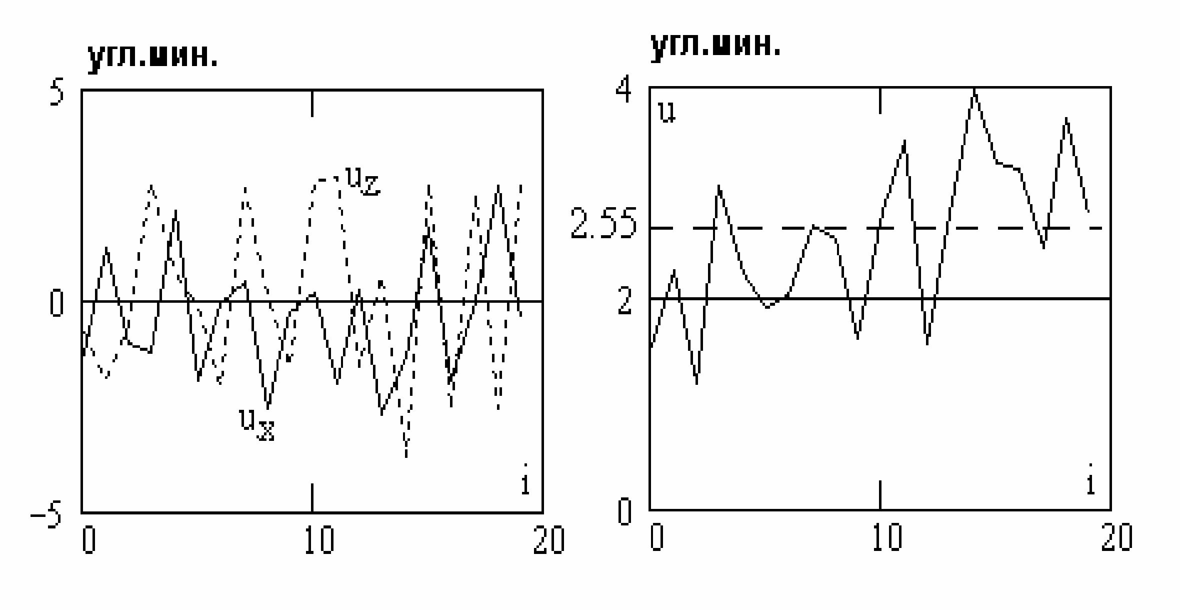


Рис.68. Зависимость проекций углов отклонения орта выходного луча (слева) и модуля угла (справа) от номера эксперимента. Среднее значение угла отклонения 2,55 угл. мин

Модель развёртки призмы

Применение развёртки призм в расчётах схем сохраняет оптическую ось без изломов. В результате погрешностей углов призмы выходной луч испытывает отклонения в поперечной плоскости *OXY*. Модель развёртки призмы дополняется трассировкой осевого луча и спрямляющими поворотами пространства изображения после каждой оптической поверхности (96). Текст программы представлен на рис.69.

В отличие от текста программы моделирования призмы в модели развёртки вводится орт осевого луча (P), координаты которого (p,q,m), полученные после взаимодействия с текущей оптической плоскостью (программа React) используются для построения матрицы спрямления оптической оси (R).

Результат моделирования отличается от предыдущего результата лишь направлением распространения.

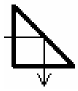
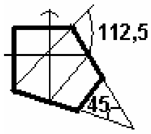
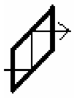
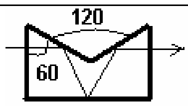
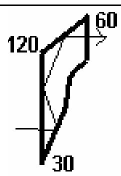
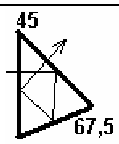
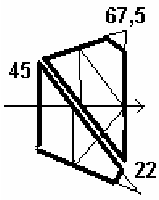
Аналогично на основе приведённых формул могут быть построены модели призм с исследованием структуры изображения и габаритного расчёта.

$$
\begin{aligned}
\mathrm{PrizmR}(S) := \;&\left|\;\begin{aligned}
&k \leftarrow \mathrm{cols}(S) - 1\\
&L \leftarrow S^{\langle 0\rangle}\\
&n \leftarrow (L)_2\\
&(L)_2 \leftarrow \sqrt{1 - \left[(L)_0\right]^2 - \left[(L)_1\right]^2}\\
&P \leftarrow (0 \quad 0 \quad 1)^T\\
&\text{for } i \in 1..k\\
&\quad\left|\;\begin{aligned}
&n_t \leftarrow \mathrm{if}(i = 1, 1, n)\\
&n'_t \leftarrow \mathrm{if}(i = 1, n, \mathrm{if}(i = k, 1, -n))\\
&P \leftarrow \mathrm{React}\left(P, S^{\langle i\rangle}, n_t, n'_t\right)\\
&L \leftarrow \mathrm{React}\left(L, S^{\langle i\rangle}, n_t, n'_t\right)\\
&p \leftarrow \frac{P_0}{n'_t}\\
&q \leftarrow \frac{P_1}{n'_t}\\
&m \leftarrow \frac{P_2}{n'_t}\\
&R \leftarrow \begin{pmatrix} \frac{q}{\sqrt{1-m^2}} & \frac{p}{\sqrt{1-m^2}} & 0 \\ \frac{-m\cdot p}{\sqrt{1-m^2}} & \frac{m\cdot q}{\sqrt{1-m^2}} & \sqrt{1-m^2} \\ p & -q & m \end{pmatrix} \quad \text{if } \; |m| < 1 - 10^{-9}\\
&R \leftarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{otherwise}\\
&L \leftarrow R\cdot L
\end{aligned}\right.\\
&L
\end{aligned}\right.
\end{aligned}
$$

Рис.69. Текст программы трассирования луча через развёртку реальной призмы

В таблице приведены значения конструктивных коэффициентов некоторых призм.

Таблица 5. Призмы и коэффициенты развёртки

№	Наименование	Вид	Коэффициент
1	Простая прямоугольная призма		1
2	Пентапризма		$2+\sqrt{2}$
3	Ромбическая призма		2
4	Призма Довэ		3,337
5	Призма Аббе		$3\sqrt{3}$
6	Призма Лемана		$2{,}5\sqrt{3}$
7	Призма Шмидта		$1+\dfrac{\sqrt{2}}{2}$
8	Призма Пехана		$\dfrac{5+3\sqrt{2}}{2}$

ПРИЛОЖЕНИЕ

Лабораторная работа №1

Моделирование передаточной функции первичных погрешностей параллелограммного механизма

Схема механизма приведена на рис.1

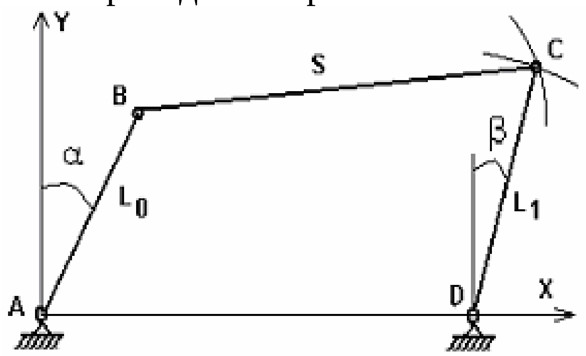


Рис.1. Схема четырёхзвенного механизма

Передаточная функция параллелограммного механизма: β = α не содержит параметров конструкции: координат положения шарниров *A* и *D*, длин шатуна *AB*, коромысла *DC* и штанги *BC*. В этом случае можно использовать метод преобразования исходной схемы устройства [1] в четырёхзвенный механизм. Суть его заключается в изменении исходных линейных размеров контура четырёхзвенника, получении аналитического выражения, с помощью которого путём дифференцирования находят соответствующую передаточную функцию.

Метод моделирования является более удобным и наглядным способом анализа. Кроме того, сложность аналитических преобразований, если такие возникают, не является для метода моделирования препятствием. Метод моделирования позволяет получить все возможные передаточные функции.

Пусть заданы следующие параметры четырёхзвенного механизма:

$$A(0,0), \quad |AB| = L_0, \quad |BC| = S, \quad D(x_1, y_1), \quad |CD| = L_1.$$

и угол поворота шатуна α.

Выведем необходимые выражения для моделирования устройства. Координаты точки *B*:

$$x_B = -L_0 \sin(\alpha), \quad y_B = L_0 \cos(\alpha), \qquad (1)$$

где (x_0,y_0) – координаты шарнира *A*, L_0 – длина кривошипа.

Координаты точки *C* найдем на пересечении дуг с центрами в точках *B* и *D* и радиусами, соответственно, *S* и L_1 (рис.1)

$$\begin{aligned}(x_C - x_B)^2 + (y_C - y_B)^2 = S^2\\ (x_C - x_1)^2 + (y_C - y_1)^2 = L_1^2\end{aligned}. \qquad (2)$$

Вычитая уравнения одно из другого, находим связь между координатами:

$$y_C = a - bx_C, \quad a = \frac{S^2 - L_0^2 - L_1^2 + x_1^2 + y_1^2}{2(y_1 - y_B)}, \quad b = \frac{x_1 - x_B}{y_1 - y_B}. \qquad (3)$$

Подставляя (3) в первое уравнение (2) получаем квадратное уравнение:

$$\begin{aligned}&Ax_C^2 - 2Bx_C + C = 0,\\ &A = 1 + b^2, \quad B = x_B + b(a - y_B), \quad C = x_B^2 + (a - y_B)^2 - S^2.\end{aligned} \qquad (4)$$

Анализ показывает, что из двух решений задаче удовлетворяет одно:

$$x_C = \frac{B + \sqrt{B^2 - AC}}{A}. \qquad (5)$$

Координаты шарнира *C* известны, и, наконец, определяем искомый угол поворота коромысла *DC* с помощью обратной тригонометрической функции:

$$\beta = \begin{cases} 0, & |x_1 - x_C| < \varepsilon, \\ \operatorname{arctg}\left(\dfrac{y_1 - y_C}{x_1 - x_C}\right)^o + 90, & \beta < 90^o \end{cases}, \qquad (6)$$

где ε – малая величина, например, 10^{-9}.

На этом заканчивается первый этап моделирования, в результате которого получены необходимые математические выражения.

Второй этап – составление блок схемы программы, моделирующей работу механизма.

Необходимо очертить круг вводимых величин и проследить их преобразование в теле программы до формирования выходных величин (см. раздел 2.6).

Третий этап написание программы и её отладка. Тестом для отладки может служить, очевидно, параллелограммный механизм в его идеальном исполнении.

На четвёртом этапе, имея в распоряжении отлаженную программу для вычисления передаточной функции *F* четырёхзвенного механизма, проводим его исследование.

Для вычисления коэффициента влияния первичной погрешности β, например, длины кривошипа L_0, достаточно воспользоваться следующей формулой, полагая ΔL малой величиной:

$$\beta_{L0}(\alpha) = \frac{F(\alpha, x_1, y_1, L_0 + \Delta L,, L_1, S) - \alpha}{\Delta L}. \qquad (7)$$

Строим график погрешности в зависимости от входного угла поворота кривошипа.

Например, для случая $x_1 = S = 10$, $y_1 = 0$, $L_1 = L_2$, $\Delta L = 10^{-3}$ график коэффициента влияния (7) имеет вид:

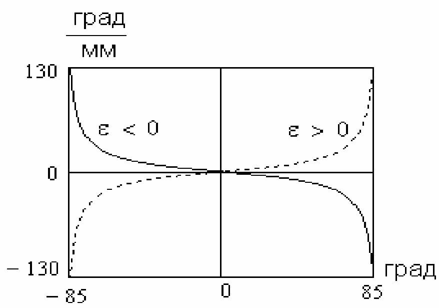


Рис.2. Зависимость коэффициента влияния погрешности поворота коромысла параллелограммного механизма от угла поворота

Как видим из графика, погрешность минимальна в окрестности угла 0^o. Передаточная функция в окрестности 0^o линейна, крутизна графика составляет $k = 0{,}2$.

$$
F(\alpha, x1, y1, L_0, L_1, S) := \left|
\begin{array}{l}
\alpha \leftarrow \alpha \cdot deg \\
x_B \leftarrow -L_0 \cdot \sin(\alpha) \\
y_B \leftarrow L_0 \cdot \cos(\alpha) \\
a \leftarrow \dfrac{S^2 - L_1{}^2 - L_0{}^2 + x1^2 + y1^2}{2 \cdot (y1 - y_B)} \\
b \leftarrow \dfrac{x1 - x_B}{y1 - y_B} \\
A \leftarrow 1 + b^2 \\
B \leftarrow x_B + b \cdot (a - y_B) \\
C \leftarrow x_B{}^2 + (a - y_B)^2 - S^2 \\
D \leftarrow B^2 - A \cdot C \\
x_c \leftarrow \left(\dfrac{B + \sqrt{D}}{A} \right) \\
y_c \leftarrow a - b \cdot x_c \\
t \leftarrow 0 \quad \text{if} \quad |x1 - x_c| < 10^{-9} \\
t \leftarrow \operatorname{atan}\left(\dfrac{y1 - y_c}{x1 - x_c} \right) \cdot \dfrac{1}{deg} + 90 \quad \text{otherwise} \\
t \leftarrow t - 180 \quad \text{if} \quad t > 90 \\
t
\end{array}
\right.
$$

Рис.3. Вариант текста программы моделирования четырёхзвенного механизма

Задание

1) Составить блок-схему программы.
2) Написать в среде MathCAD программу и провести её отладку.
3) Определить передаточную функцию первичной погрешности, указанной в таблице, построить график и провести анализ, определить крутизну графика.
4) Исследовать влияние отношения сторон параллелограмма на передаточную функцию
5) Исследовать влияние величины погрешности на крутизну графика передаточной функции параллелограммного механизма.

№ варианта	1	2	3	4	5	6	7	8	9
Первичная погрешность	Δx_0	Δy_0	$\Delta \mathrm{r}_0$	Δx_1	Δy_1	Δr_1	ΔL_0	ΔL_1	ΔS

Лабораторная работа №2

Моделирование передаточной функции погрешности фокусного расстояния линзовой системы

Задание.

1) Прочитать раздел 6 настоящего пособия.
2) Составить комментарии к программе Kardinal.
3) Составить системную матрицу объектива.
4) Провести вычисления фокусного расстояния в среде MathCAD.
5) Определить передаточные функции первичных погрешностей объектива Русинова [3] (см. таблицу 1) двух указанных параметров. Промоделировать соответствующие частичные погрешности и суммарную погрешность.
6) Рассчитать коэффициент корреляции.
7) Построить гистограмму суммарной погрешности.
8) Обсудить результаты и сделать выводы.

Таблица 1. Объектив Русинова

Объектив	r	n_C	t
f''=200,0 D=30 мм	144,66		
		1,6213	3,0
	54,44		
		1,5139	9,0
	-189,13		

Таблица 2. Варианты задания к лабораторной работе 2

№ варианта	Факторы	№ варианта	Факторы	№ варианта	Факторы
1	r_1, r_2	8	r_2, n_1	15	r_3, t_2
2	r_1, r_3	9	r_2, n_2	16	n_1, n_2
3	r_1, n_1	10	r_2, t_1	17	n_1, t_1
4	r_1, n_2	11	r_2, t_2	18	n_1, t_2
5	r_1, t_1	12	r_3, n_1	19	n_2, t_1
6	r_1, t_2	13	r_3, n_2	20	n_2, t_2
7	r_2, r_3	14	r_3, t_1	21	t_1, t_2

Лабораторная работа №3

Окуляр Эрфле

Задание

1) Прочитать раздел 7 настоящего пособия.
2) Составить комментарий к программе Pupil.
3) Отладить и протестировать программу.
4) Определить передаточные функции A_k, *(k=1,2,3)* торцевого биения линз окуляра.
5) Задавшись угловой погрешностью наблюдения Δy и полагая, что данные первичные погрешности случайны и не содержат систематической погрешности, определить допуски на первичные погрешности δx_k. В этом

случае: $\delta x_k = \dfrac{\Delta y}{A\sqrt{n}}$,

где n – число первичных погрешностей.

6) Промоделировать первичные погрешности в определённых в пункте 4 границах (см. лаб.2) и определить суммарную погрешность.
7) Сравнить промоделированную суммарную погрешность с заданной величиной Δy. Сделать выводы.

Лабораторная работа №4

Определение положения входного зрачка и коэффициентов виньетирования

Задание

Прочитать раздел 8 и рассчитать положение входного зрачка и коэффициенты виньетирования для объективов [4], конструктивные параметры которых представлены в таблицах 1-4.

1) Минитар-1Л,
2) Опалар
3) 3С
4) Т-54

Таблица 1. Конструктивные параметры объектива “Минитар-1Л”
(f'=32,59, s=-765 – -2966 мм, λ=646,1 нм, $2\omega=65^o$)

№	*R* (мм)	*t* (мм)	*n*	Марка стекла	*D* (мм)
1	12,078				14,2
		1,95 ± 0,02	1,746046	СТК9	
	18,707				13,08
2		0,06 ± 0,02	1		
	13,459				12,52
		1,95 ± 0,1	1,746046	СТК9	
	81,66				11,84
		0	1		
	81,66				11,84
		0,78 ± 0,1	1,746231	ТФ4	
	19,999				10,26
3		1,76 ± 0,01	1		
	-48,98				9,16
		0,78 ± 0,02	1,746231	ТФ4	
	11,092				8,56
4		1,17 ± 0,01	1		
	24,55				8,66
		1,76 ± 0,02	1,746231	ТФ4	
	-24,55				8,60
5		0,9	1		
	∞				8.29
			1		

Таблица 2. Конструктивные параметры объектива “Опалар” (f'= 34,807 мм, s = ∞ , λ = 0,65628 мкм, 2ω =56°)

№	*R*(мм)	*t* (мм)	*n*	Марка стекла	D
1	14,093				15,82
		1,078	1,739667	СТК19	
	15,065				12,98
		0,04	1		
2	12,745				15,52
		1,212	1,739667	СТК19	
	14,970				12,98
		0,03	1		
3	12,815				14,15
		1,950	1,739667	СТК19	
	34,081				10.88
		0,8	1		
4	-118,557				13,88
		1,557	1,666602	ТФ2	
	9,855				10,74
		1,94	1		
5	28,977				13,54
		3,510	1,739667	СТК19	
	-12,827				10,21
		0	1		
6	-12,827				10.21
		0,593	1,512184	К14	
	-270,594				9,47
7		0,9	1		
	∞				12,43

Таблица 3. Конструктивные параметры объектива “3С”
(f'=34,80 мм, $s=\infty$, λ = 0,65628 мкм, $2\omega = 63^o$)

№	R (мм)	t (мм)	N	Марка стекла	D
1	12,474				14,91
		1,69	1,739667	СТК19	
	16,493				14,08
		$0{,}29\pm0{,}02$	1		
2	12,706				13,30
		2,18	1,739667	СТК19	
	28,71				12,20
		$0{,}8\pm0{,}01$	1		
3	-130,32				12,20
		1,07	1,666602	ТФ2	
	9,954				10,05
		$1{,}98\pm0{,}02$	1		
4	32,28				10,00
		3,02	1,739667	СТК19	
	-13,243				9,60
		0	1		
5	-13,243				9,53
		0,62	1,512184	К14	
	-146,22				9,40
6		0,9	1		
	∞				12,429

Таблица 4. Конструктивные параметры объектива “Т-57”
(f'=34,94 мм, $s = \infty$, λ = 0,65628 мкм, $2\omega = 63^o$)

№	R (мм)	t (мм)	N	Марка стекла	D
1	9,772				11,5
		$3{,}81\pm0{,}01$	1,656004	СТК3	
	18,621				9,42
		$0{,}98\pm0{,}01$	1		
2	-29,850				9,42
		$0{,}94\pm0{,}01$	1,658782	БФ28	
	10,471				8,3
		$0{,}81\pm0{,}01$	1		
3	20,51				8,45
		$2{,}55\pm0{,}02$	1,738053	СТК9	
	-20,51				8,5
		0,9	1		
	∞				9,98

Лабораторная работа №5

Передаточные коэффициенты децентрировки

Задание

Прочитать раздел 9 данного пособия и определить передаточные коэффициенты децентрировки поверхностей линз объективов [4]:

1) Минитар-1Л,
2) Опалар
3) ЗС
4) Т-54

Конструктивные параметры объективов приведены в лабораторной работе №4.

Литература

1. С.М.Латыев. Конструирование точных (оптических) приборов, часть 2, С-Петербург, 1999, с.55.
2. А.Г. Сергеев, В.В.Крохин. Метрология. М., Логос, 2000.
3. М.М. Русинов, Габаритные расчёты оптических систем, М., 1963, с.271-391.
4. С.М.Латыев, Г.В. Егоров, И.Н. Тимощук, Проектирование оптико-электронных приборов и систем. Методические указания к выполнению курсового проекта. С-Петербург, 2001, с. 46-47.
5. В.Р. Чуриловский. Теория оптических приборов. М. Машиностроение, 1966.
6. И.Н. Бронштейн, К.А. Семендяев. Справочник по математике. М., Наука, 1981.
7. М. Борн, Э. Вольф. Основы оптики. М., Наука, 1970.
8. М.Д. Мальцев. Расчёт допусков на оптические детали. М., Машиностроение, 1974.
9. С.М. Латыев. Компенсация погрешностей в оптических приборах. Л., Машиностроение, 1985.